Variants of Maker-Breaker games on complete and random graphs

Dissertation (monograph) approved by the
Doctoral Degree Committee of
Hamburg University of Technology

in pursuit of the academic degree of

Doktor der Naturwissenschaften (Dr. rer. nat.)

written by
Yannick Mogge

from
Winsen (Luhe)

2025

1. Reviewer: Prof. Dr. Anusch Taraz
2. Reviewer: Prof. Dr. Mirjana Mikalački

Date of oral examination: February 18th, 2025

Summary

An $(a : b)$ positional game is a perfect information game for two players, which is played on a hypergraph $(X, \mathcal{F})$, where $\mathcal{F}$ denotes the family of winning sets. During each round, both players claim a and b unclaimed elements of the board, respectively, according to some predefined rules. The first player to achieve their goal (for example being the first player to claim all elements of a winning set) is declared the winner of the game.

Maker-Breaker games are a specific variant of positional games. Here, both players (called Maker and Breaker) alternately claim a and b unclaimed elements of the board, and Maker wins if she is able to fully claim a winning set by the end of the game, otherwise Breaker wins. A common question when considering Maker-Breaker games is which player has a winning strategy. Since many standard games are easy wins for Maker, fast strategies take this concept one step further and not just ask the question by whom, but also how fast such games can be won.

One variant of Maker-Breaker games are Waiter-Client games. These are played similarly to Maker-Breaker games, but the distribution of the elements of the board is done as follows: For some bias b, during each round of the game Waiter offers $b + 1$ elements of X to Client from which he claims one for himself and the rest go to Waiter. Proceeding like this Waiter wins the game if she can force Client to claim all the elements of any winning set from $\mathcal{F}$. In Chapter 3 we study fast strategies for several Waiter-Client games played on the edge set of the complete graph on n vertices, i.e. $X = E(K_n)$, in which the winning sets are perfect matchings, Hamilton cycles, pancyclic graphs, fixed spanning trees, and factors of a given graph, in particular of fixed trees and triangles.

Another variant of Maker-Breaker games are Connector-Breaker games. These games are played similarly to regular Maker-Breaker games with the restriction for Connector (who takes the role of Maker) that she can only claim free elements of the board in such a way that her graph stays connected throughout the game. We will study such games in Chapter 4, and in particular in Section 4.1 we will study the $(2 : 2)$ Connector-Breaker connectivity game on a random graph $G \sim G_{n,p}$ and show that for $p = n^{-2/3-o(1)}$ Breaker asymptotically almost surely has a strategy to isolate a vertex in Connector's graph, therefore winning the game.

The last variant of Maker-Breaker games which plays a huge role in this thesis are Walker-Breaker games. These games are an extension of Connector-Breaker games in the sense that Walker (who takes the role of Maker) is only allowed to claim elements of the board according to a walk. We will study these games in Chapter 4 as well, and in Section 4.2 we will consider Walker's side of the $(2 : 2)$ Walker-Breaker Hamiltonicity game on a random graph $G \sim G_{n,p}$ and show that for $p = n^{-2/3+o(1)}$ Walker asymptotically almost surely has a strategy to claim a Hamilton cycle.

Combining both main results from Chapter 4 we thus show that the threshold probability for both the $(2 : 2)$ Connector-Breaker connectivity game on $G \sim G_{n,p}$ and the $(2 : 2)$ Walker-Breaker Hamiltonicity game on $G \sim G_{n,p}$ are of order $n^{-2/3+o(1)}$.

Contents

Chapter 1

Introduction

1.1 Overview of this thesis

Games! Already at a young age I became interested in the concept of games, and in some way or another, games have been a constant companion throughout my studies and personal life ever since. Over the years I got introduced to many combinatorial games of different complexities such as NIM or chess, and also to probably the most famous of them all: Tic-Tac-Toe. While it was easy to figure out that nobody could win Tic-Tac-Toe when both players played optimally, finding winning strategies for NIM was quite a bit harder, and to the current day finding a winning strategy for chess is still an unsolved problem.

Out of the three mentioned games, the game Tic-Tac-Toe belongs to the category of positional games, a specific type of combinatorial games, in which two players alternate in claiming yet unclaimed elements of some board according to some predefined rules. Before the game starts, a family of winning sets is defined, as well as goals which the players have to achieve to win the game. Pretty soon after starting my PhD, I learned more about positional games, and got introduced to different types of positional games such as Maker-Breaker games. Not much later I also learned about more recent variants of Maker-Breaker games, namely Waiter-Client games, Connector-Breaker games, and Walker-Breaker games. Those games and their study stayed with me throughout my whole PhD, and some of the results of my study can be found in this thesis.

While the study of combinatorial games is interesting in its own right, one might ask, why even bother to find strategies for any player to win the game. Only a finite amount of possible configurations can occur during the course of the game, so when there is no randomness involved we could just check every possible state with a computer to find out which player wins the game. The problem lies in the computing power of our available computers, which is some order of magnitude smaller than what is necessary to solve games like chess. Therefore finding simpler strategies which succeed against any strategy of the opponent becomes pretty useful. While we unfortunately will not solve chess in this thesis, we will nonetheless prove interesting results for some combinatorial games, with slightly simpler rulesets.

1.2 Historical Background

This section provides background information for the results of this thesis, starting with an introduction to positional games in Subsection 1.2.1. We then introduce Maker-Breaker games in Subsection 1.2.2, followed by biased games in Subsection 1.2.3 and fast strategies in Subsection 1.2.4. Next we will introduce Waiter-Client games in Subsection 1.2.5. We continue with a definition of the threshold probability and probabilistic intuition in Subsection 1.2.6, and finish this section with Connector-Breaker games in Subsection 1.2.7 and finally Walker-Breaker games in Subsection 1.2.8.

1.2.1 Positional games

Since we just gave a rough definition of positional games before, let us now define them more precisely. A *positional game* is a perfect information game that is played by two players on some *board* X equipped with a family $\mathcal{F} \subset 2^X$ of subsets, which represent the *winning sets*. In this context perfect information means that all of the information about the board, past moves, and winning sets, is open to both players at all time, and there exist no secrets (like hidden cards in hand, for example). During each round both players claim one (or more) previously unclaimed elements of the board according to predefined rules, which depend on the specific game variant. We also add some *winning criteria* that the players have to achieve to win the game, which can also differ for both players. To give some examples of such winning criteria, these can range from being the first player to fully claim a winning set, to claiming a specific fraction of each winning set, or even completely avoiding specific sets or preventing the other player from achieving their goal.

The most famous example of a positional game is probably the aforementioned Tic-Tac-Toe, where both players alternately claim one element in a 3×3 grid, and the first player to claim all elements in a row, column, or diagonal, wins the game.

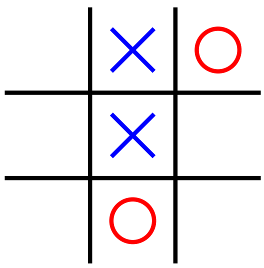

Figure 1.2.1: The game Tic-Tac-Toe

Another famous example of a positional game is Hex, which was invented by Piet Hein in 1942. Here, two players try to build a connection between two opposite sides of a board by claiming elements of a rhombic grid of hexagons. This game was even commercially sold.

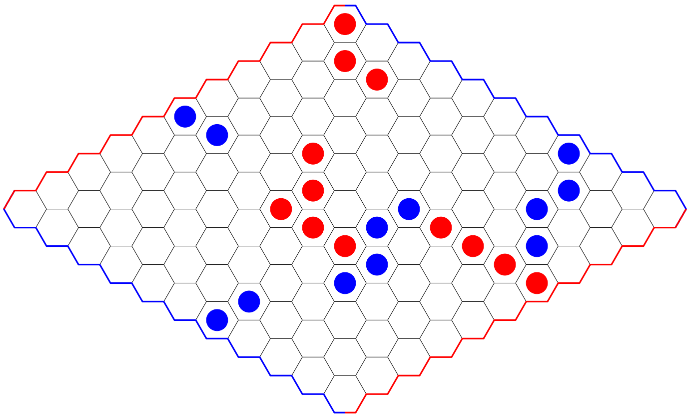

Figure 1.2.2: The game Hex

The intense study of positional games was started with a seminal paper by Hales and Jewett [38], who considered a more generalised version of Tic-Tac-Toe, and was continued by Erdős and Selfridge [28], who shortly afterwards introduced Maker-Breaker games. We will explain this type of games in more detail in Subsection 1.2.2. Many beautiful results have been proven ever since that also lead to the introduction and study of several other versions of positional games such as Waiter-Client games. These games were introduced under the name Picker-Chooser game by Beck (see e.g. [6]) and we will give a more detailed description of Waiter-Client games in Subsection 1.2.5. Interestingly, some of the results even provide intriguing connections between positional games and other branches of combinatorics such as random graph theory (see e.g. [6], [9], [10], [33], [49], [52]), extremal combinatorics, and Ramsey theory (see e.g. [6], [38], [58]). These results include (but are not limited to) finding strategies for the players (when playing on the edges of a complete graph) to either claim or prevent large cliques, claiming the edges of a Hamilton cycle, a pancyclic graph, or a specifically chosen graph H, and generating random graphs or graphs with large minimum degree.

For an introductory overview of positional games we also recommend the book **Positional Games** by Hefetz, Krivelevich, Stojaković, and Szabó [42], as well as the survey **Positional Games** by Krivelevich [50]. For further reading we also recommend Beck's famous book **Combinatorial Games: Tic-Tac-Toe Theory** [6], which also includes the aforementioned games Tic-Tac-Toe and Hex as well as many other interesting games.

1.2.2 Maker-Breaker games

Positional games, in which both players try to achieve the same goal (while at the same time trying to prevent the other player from achieving their goal), belong to the category of so-called *strong games*. When we think about who can win a strong game, by using a strategy stealing argument it is quite easy to see that the second player cannot have a winning strategy (see e.g. [42]). Let us assume that a winning strategy for the second player exists. The first player can just play an arbitrary move and afterwards pretend to be the second player and steal the second player's winning strategy, which would result in a win for the first player, thus such a strategy cannot exist. This argument can be applied since claiming more elements of the board never hurts the respective player.

The game Tic-Tac-Toe belongs to the category of strong games. As mentioned before, if both players play optimally, Tic-Tac-Toe ends in a draw. On the other hand, a draw in Hex is impossible, since at the end of the game, if one of the players cannot connect their opposite sides, the other player has to have a connection between their sides. Using the strategy stealing argument, this means that there has to exist a strategy for the first player to win, but so far no winning strategy is known.

To combat the problem that the second player can never win a strong game, so-called *Maker-Breaker* games were first introduced by Erdős and Selfridge in [28] and have since become the most studied type of positional games. Maker-Breaker games bestow different goals on both players (who are called *Maker* and *Breaker*, respectively): If Maker succeeds in fully claiming all elements of at least one winning set, she is declared the winner. If she is not able to do so, Breaker wins the game, which essentially means that he claimed at least one element of every winning set. This also means that a draw is impossible.

To get some intuition about Maker-Breaker games let us briefly consider Maker-Breaker Tic-Tac-Toe, where Maker's goal is the same as in regular Tic-Tac-Toe, while Breaker just tries to prevent Maker from fully claiming a row, column, or diagonal. It is a simple exercise to show that in this game Maker actually has a winning strategy.

Maker-Breaker games are often played on the edge set of some given graph, which is quite often the complete graph on n vertices, which we will denote by K_n. Then both players alternately claim edges of this graph, with Maker's goal for example being to create a subgraph which fulfils certain properties, like containing a Hamilton cycle, a spanning tree, or even just a triangle.

Our first example of a Maker-Breaker game on a graph is the Maker-Breaker *connectivity game*. This game variant is played on the edges of some graph G where the winning sets are all spanning trees of G. For this game, Lehman [54] proved a powerful theorem (by using an elegant inductive argument) which classifies all graphs on which Maker can win the Maker-Breaker connectivity game as a second player. This theorem can be found in the following form in [42].

Theorem 1.2.1 (Theorem 7.1.2 in [42])**.** *Let G be a graph. Maker (as the second player) has a winning strategy in the Maker-Breaker connectivity game on $E(G)$ within $n-1$ turns, if and only if G contains at least two edge disjoint spanning trees.*

Let us consider the Maker-Breaker connectivity game played on the edges of K_n. Since for $n > 3$ this graph contains two (or even more) edge disjoint spanning trees, using Theorem 1.2.1 we can immediately conclude that this game is won by Maker.

1.2.3 Biased games

Theorem 1.2.1 suggests, that when playing on $E(K_n)$ Breaker is quite weak compared to Maker. In fact, many natural games on $E(K_n)$ are easy wins for Maker, for example the *perfect matching game*, where the winning sets are all perfect matchings of K_n, and the *Hamiltonicity game*, where the winning sets are all Hamilton cycles of K_n (see e.g. [16], [42]). Thus we can think about ways to give Breaker more power, for example by allowing him to claim more than one edge per turn. Of course, we can also allow Maker to claim more than one edge per turn. This leads us to the introduction of a *bias* for Maker-Breaker games. While this is the most common method to increase Breaker's power, there are many other options available, like playing on a sparser board than the complete graph, or restricting Maker's movements. We will come back to these concepts in Subsections 1.2.6–1.2.8.

Let m and b be integers. In the $(m : b)$ Maker-Breaker game, Maker and Breaker are allowed to claim (up to) m or (up to) b elements in each round, respectively. From now on, if $m = b = 1$ we call the game *unbiased*, otherwise we call the game *biased*, with m and b being the respective biases of Maker and Breaker. If $m = 1$ and $b > 1$ we also call the game *b-biased*.

We would like to emphasize that Maker-Breaker games are *bias monotone* in the sense that claiming more elements of the board never hurts the corresponding player. Given $(X, \mathcal{F})$ and having Maker's bias m fixed, we can thus find an integer b_0, called the *threshold bias*, such that Breaker wins the (m, b) Maker-Breaker game if and only if $b \geqslant b_0$ holds (except for trivial games, where Maker can win before Breaker's first move).

It is a very natural first question to ask what happens if we increase Breaker's bias b, while at the same time leaving Maker's bias m at 1. As mentioned before, the unbiased version of the perfect matching game on $E(K_n)$ is an easy win for Maker, and it is rather simple to find a winning strategy for Maker, provided that n is sufficiently large (see e.g. [42]). Thus let us see what happens when b becomes larger. This ultimately leads to the question of finding the threshold bias $b_{\mathcal{PM}}$ where a Maker's win suddenly turns into a Breaker's win (see e.g. [37], [49]). In the case of the perfect matching game on $E(K_n)$ this happens when $b_{\mathcal{PM}} = (1 + o(1))\frac{n}{\ln(n)}$.

A very similar behaviour can be shown for the b-biased connectivity game on $E(K_n)$, which was first studied by Chvátal and Erdős [16]. They proved that its threshold bias $b_{\mathcal{C}}$ is bounded from above by $(1 + o(1))\frac{n}{\ln(n)}$, since for a larger bias Breaker has a strategy to isolate a vertex in Maker's graph. A matching lower bound was later given by Gebauer and Szabó [37].

Lastly, let us also mention the Hamiltonicity game on $E(K_n)$ again. For the b-biased version, the upper bound for the threshold bias $b_{\mathcal{H}}$ trivially carries over from the connectivity game, since Hamilton cycles are spanning structures and Breaker's strategy in the connectivity game isolates a vertex and thus prevents any spanning structure. A matching lower bound was later proven by Krivelevich [49], thus showing that the threshold bias for the $(1 : b)$ Hamiltonicity game is also of the order $(1 + o(1))\frac{n}{\ln(n)}$.

1.2.4 Fast strategies

In some way, the threshold bias gives us some measure of how powerful Maker is in comparison to Breaker. For all of the games we mentioned so far, Breaker must be allowed to claim a lot more edges per turn than Maker to be able to win. But we can also think of other ways to measure Maker's power. For example, let us consider a game where we know that Maker has a winning strategy, but then look at how fast Maker can actually achieve her goal against any strategy of Breaker. Therefore we would like to introduce the concept of *fast strategies* next, where we are not just interested in a strategy for Maker to win, but a strategy for Maker to win as fast as possible.

While the question of how fast Maker can win a specific game is interesting in its own right (since as mentioned before an answer to it can be seen as a measure of how powerful Maker is), finding fast winning strategies is even more relevant, as sometimes winning strategies for more involved games can be found by splitting the game into several stages, in which Maker's goal is to first create a simpler or a nice behaving structure as fast as possible, so that afterwards she still has enough options to extend this structure into a full winning set, since so far not too many edges were claimed by Breaker (for an example, see [49]).

Let $\tau_{MB}(\mathcal{F}, b)$ denote the smallest integer t such that Maker has a strategy to win the b-biased Maker-Breaker game with winning family $\mathcal{F}$ within t turns. Observe first that this value is bounded trivially from below by $m_{\mathcal{F}} := \min\{|F| : F \in \mathcal{F}\}$. If Maker can win in exactly this number of rounds, we will say that the game is won *perfectly fast*, while we say that the game is won *asymptotically fast* when the smallest necessary number of rounds to win is of size $(1 + o(1))m_{\mathcal{F}}$.

As a first example, let us look at the unbiased Maker-Breaker connectivity game on K_n again. Let $\mathcal{C}_n$ denote the family of all spanning trees of K_n, and let n be large enough. In our example this means that we therefore consider $\tau_{MB}(\mathcal{C}_n, 1)$. When Maker plays with the strategy due to Lehman [54], she creates a spanning tree by always connecting a new vertex (which is chosen in a specific way) to her current graph in each turn, thus she never closes a cycle during the game. As a result she never creates any wasted edges and therefore she wins the game perfectly fast. Since every spanning tree of K_n consists of $n - 1$ edges, we get $\tau_{MB}(\mathcal{C}_n, 1) = n - 1$.

As a second example, let us look at the Hamiltonicity game. This time, let $\mathcal{H}_n$ denote the family of all Hamilton cycles of K_n, and let n be large enough. Already in their early paper, Chvátal and Erdős [16] proved that Maker can win this game in at most $2n$ rounds. This was later improved to $\tau_{MB}(\mathcal{H}_n, 1) \leqslant n + 2$ by Hefetz, Krivelevich, Stojaković, and Szabó [40] by building a Hamilton path from a perfect matching and then applying the Pósa rotation technique [60]. Finally, Hefetz and Stich [45] were fighting for the exact result and proved that $\tau_{MB}(\mathcal{H}_n, 1) = n + 1$ by providing a rather technical proof involving multiple case distinctions.

Let us briefly note that Maker cannot win in n turns, since after her $(n - 1)$st turn there is at most one edge which could turn Maker's graph into a Hamilton cycle, thus Breaker can claim this edge and prevent Maker from winning in n turns. Therefore Maker cannot win this game perfectly fast. The minimal number of turns in which Maker theoretically could win this game is thus $n+1$, and she can indeed win in that many turns.

Not so much is known about fast strategies for other positional games other than Maker-Breaker games, but there exist some results for fast strategies in unbiased Avoider-Enforcer games [41], which is a type of game where Avoider tries to avoid claiming a winning set, while Enforcer tries to force Avoider to claim a winning set. This thesis provides some further results to partially fill this gap: Chapter 3 is devoted to the study of fast strategies for another one of those variants, namely Waiter-Client games.

1.2.5 Waiter-Client games

Waiter-Client games and *Client-Waiter games* were first introduced by Beck (see e.g. [6]) under the names *Picker-Chooser games* and *Chooser-Picker games*, and have received increasing attention recently (see e.g. [1], [2], [20], [26], [27]). Similar to Maker-Breaker games, these games are played on some hypergraph $(X, \mathcal{F})$ with biases m and b (if $m = b = 1$ we also call the game unbiased), both players claim elements from X, and exactly one player aims for a winning set while the other player tries to prevent that. This time though, instead of claiming the elements of X alternately, the rule of how Waiter and Client pick the elements of the board is as follows: In every round, Waiter picks $m+b$ previously unclaimed elements of the board and Client chooses exactly m of those elements to be claimed by him, while all of the other elements go to Waiter. If in the last round there are less than $m+b$ unclaimed elements of the board left, Client chooses (up to) b elements which go to Waiter and afterwards claims the leftover elements himself.

Contrary to Maker-Breaker games, in the Waiter-Client game, Waiter is said to be the winner if by the end of the game Client claimed all the elements of any winning set from $\mathcal{F}$. In this case we say that Waiter *forces* Client to occupy a winning set. In the Client-Waiter game, Waiter however is said to be the winner if she can prevent Client from claiming all the elements of any winning set until the end of the game. Otherwise, Client is declared the winner.[1]

One may wonder whether there is some connection between the above games and Maker-Breaker games in general. Beck [6] observed that for a few natural families of winning sets $\mathcal{F}$ Waiter wins the Waiter-Client (or Client-Waiter) game more easily than Maker (or Breaker) wins the corresponding Maker-Breaker game. Later on, this was conjectured to be true in general by Csernenszky, Mándity, and Pluhár [25].

Conjecture 1.2.2 (Conjecture 1 in [25])**.** *Waiter wins a Waiter-Client (Client-Waiter) game on* $(X, \mathcal{F})$ *if Maker (Breaker) as second player wins the corresponding Maker-Breaker game.*

While the above conjecture has recently been disproved by Knox [47] with a very specific counterexample, there is still a chance that Beck's intuition holds for many typical winning

[1] Sometimes in the literature the distribution of the edges is switched: in those cases Client picks m elements that go to Waiter while he claims b for himself, and Waiter wins if she owns a winning set at the end of the game.

families $\mathcal{F}$. A few examples of the games supporting the conjecture are already given by Bednarska-Bzdęga in [8], and it is still interesting to find out which games won by Maker are also won by Waiter. In Chapter 3 we will provide a variety of examples in which Waiter in a Waiter-Client game not just wins the game, but also wins at least as fast as Maker does in the corresponding Maker-Breaker game. To make this more precise, analogously to the definition of $\tau_{MB}(\mathcal{F}, b)$ before, let $\tau_{WC}(\mathcal{F}, b)$ denote the smallest integer t such that Waiter has a strategy to win the b-biased Waiter-Client game with winning family $\mathcal{F}$ within t rounds. We thus want to find examples of games such that $\tau_{WC}(\mathcal{F}, b) \leqslant \tau_{MB}(\mathcal{F}, b)$ holds.

1.2.6 Probabilistic Intuition

Going back to the idea of how to give Breaker more power in a Maker-Breaker game, we will next consider the method of playing on a sparser board than the complete graph K_n. A typical approach is to play a (not necessarily biased) Maker-Breaker game on the edges of a sparse random graph G sampled according to the *binomial random graph model* $G_{n,p}$ (for short we will write $G \sim G_{n,p}$), where we fix n vertices and each edge appears with probability p independently of all other choices. Since smaller values for p favour a Breaker's win, when the value of p increases the most natural question to ask is around which value of p a likely Breaker's win turns into a likely Maker's win (see e.g. [58], [61]).

It is well known that for monotone increasing graph properties $\mathcal{F}$ the binomial random graph model always comes with a *threshold probability* p^* (see e.g. [12]) such that

$$\mathbb{P}\left(G \sim G_{n,p} \text{ satisfies } \mathcal{F}\right) \to \begin{cases} 0 & \text{if } p = o(p^*) \\ 1 & \text{if } p = \omega(p^*)\,. \end{cases}$$

For some properties $\mathcal{F}$ there is even a *sharp threshold* in the sense that

$$\mathbb{P}\left(G \sim G_{n,p} \text{ satisfies } \mathcal{F}\right) \to \begin{cases} 0 & \text{if } p \leqslant (1+o(1))p^* \\ 1 & \text{if } p \geqslant (1+o(1))p^* \end{cases}$$

holds. For example, the properties of a graph being connected or containing a Hamilton cycle are examples of such properties with a sharp threshold (see e.g. [11], [46]).

There exists an interesting connection between positional games and the aforementioned threshold probability. Let us assume that in the $(m:b)$ Maker-Breaker game on K_n Maker and Breaker do not play according to a deterministic strategy but instead they play purely at random. Once every edge of K_n has been claimed by either player, the final graph consisting of Maker's edges will then behave similarly to a random graph $G \sim G_{n,p}$ with $p = \frac{m}{m+b}$. It is well known that the (sharp) threshold probability p^* for $G \sim G_{n,p}$ being connected or containing a Hamilton cycle satisfies $p^* = (1+o(1))\frac{\ln(n)}{n}$ (see e.g. [11], [46]). Surprisingly, when $m = 1$, this corresponds to $b = (1+o(1))\frac{n}{\ln(n)}$ and thus perfectly matches the threshold bias for the Maker-Breaker connectivity and Hamiltonicity game. In other words, for most values of b, a randomly played $(1:b)$ Maker-Breaker connectivity or Hamiltonicity game on K_n is very likely to end up with the same winner as the corresponding deterministically played game. This phenomenon is usually referred to as the *probabilistic intuition*.

While the Maker-Breaker connectivity and Hamiltonicity game on K_n might be the most famous examples of games that fulfil the probabilistic intuition, there exists a wide range of further games fulfilling this property as well, for example the perfect matching game and the doubly biased $(m : b)$ connectivity game when Maker's bias satisfies $m = o(\ln(n))$ [44]. On the other hand, there also exist a variety of games, where this intuition fails, such as the diameter game [5], the K_t-factor game [55], and the H-game [10]. Therefore, for any positional game it is a very natural question to ask to which of those categories it belongs.

Stojaković and Szabó [61] were the first who initiated the study of Maker-Breaker games on a random graph $G \sim G_{n,p}$. We say that a graph $G \sim G_{n,p}$ has a property $\mathcal{F}$ *asymptotically almost surely* (for short we write a.a.s.), if $\mathbb{P}\left(G \text{ has property } \mathcal{F}\right) \to 1$ for $n \to \infty$. Stojaković and Szabó considered a variety of games with their main goal to find the threshold probability p^* at which (when p becomes larger) an almost sure Breaker's win turns into an almost sure Maker's wins. The existence of such a (not necessarily sharp) threshold is guaranteed by the fact that the property of Maker having a winning strategy is monotone increasing (see e.g. [42]).

For the connectivity game it is obvious that the threshold probability needs to satisfy $p^* \geqslant (1 + o(1))\frac{\ln(n)}{n}$ since for smaller values of p a random graph $G \sim G_{n,p}$ almost surely contains isolated vertices (see e.g. [11], [46]). Stojaković and Szabó could show that indeed $p \geqslant (1 + o(1))\frac{\ln(n)}{n}$ is enough for Maker to win the connectivity game on $G \sim G_{n,p}$ almost surely. Interestingly, this threshold probability asymptotically equals the reciprocal of the threshold bias for the corresponding Maker-Breaker game on K_n. This is another phenomenon which has also been observed for many other natural games (see e.g. [30], [39], [58], [61]).

In Chapter 4 we will focus on playing on a random graph and finding the threshold probability for specific games. Again, similar to when we consider fast strategies, we will change the Maker-Breaker game variant which we study. Therefore let us introduce *Connector-Breaker games* and *Walker-Breaker games* next.

1.2.7 Connector-Breaker games

Recently, under the name *PrimMaker-Breaker games*, London and Pluhár [56] introduced a connected version of Maker-Breaker games. These games, which we will call *Connector-Breaker games* in the following, are played in the same way as Maker-Breaker games, the only difference being that *Connector* (in the role of Maker) needs to choose her edges in such a way that her graph stays connected throughout the game. In the same paper London and Pluhár [56] studied the Connector-Breaker connectivity game on K_n, and even more recently Corsten, Mond, Pokrovskiy, Spiegel, and Szabó [24] discussed the variant in which Connector's goal is to claim an odd cycle of K_n.

For the unbiased Connector-Breaker connectivity game, London and Pluhár [56] proved the following:

Theorem 1.2.3 (Theorem 1 in [56]). *Playing a* $(1:1)$ *Connector-Breaker connectivity game on a graph* G *with* n *vertices, Connector wins as the first player if and only if* G *contains a copy of* H_n*, where* H_n *is the graph* $K_{n-2,2}$ *with an additional edge inside its two-element colour class.*

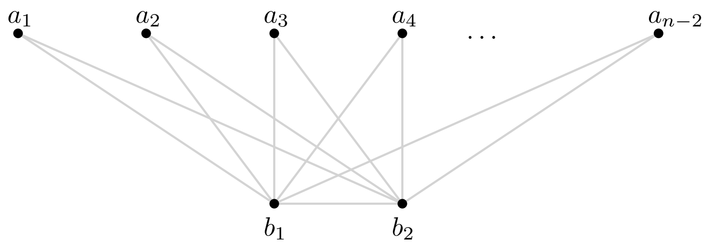

Figure 1.2.3: The graph H_n

Let us also consider the b-biased Connector-Breaker connectivity game. One can easily see that for $b \geqslant 2$ this game is won by Breaker on every graph G, because Breaker can isolate a vertex doing the following. After Connector's first turn, Breaker picks an arbitrary vertex v not touched by Connector (that is a vertex with no incident edge claimed by Connector), and then after Connector's turn he claims every available edge between v and every vertex touched by Connector (which there are at most two of). Since from this point onwards Connector needs to claim at least two edges in one turn to reach v, Breaker can easily prevent Connector from ever including v in her graph (see [56]). Therefore the threshold bias for the $(1:b)$ Connector-Breaker connectivity game on any graph G equals 2. Also, if the game is played on $G \sim G_{n,p}$, then p needs to be almost 1 (because of Theorem 1.2.3) for Connector to almost surely have a winning strategy, since otherwise G will most likely not contain a copy of H_n. Note that both of these observations are in huge contrast to the results for the analogous Maker-Breaker games. However, by increasing Maker's bias by just one, London and Pluhár [56] showed that the situation changes drastically.

Theorem 1.2.4 (Theorem 2 in [56]). *Playing a* $(2:b)$ *Connector-Breaker connectivity game on* K_n*, Connector wins if* $b < \frac{n}{8\ln(n)}$*, and Breaker wins if* $b > \frac{n}{\ln(n)}$*.*

Hence, for Connector-Breaker games a small change in Connector's bias can lead to a big difference of the threshold bias. Moreover, Theorem 1.2.4 shows that for the connectivity game the threshold bias in the $(2:b)$ Connector-Breaker variant is of the same order as in the Maker-Breaker variant.

1.2.8 Walker-Breaker games

Walker-Breaker games are defined in the same way as Connector-Breaker games, but further restrict the possible choices for *Walker* (in the role of Maker) to claim elements of the board: She is only allowed to claim edges of the graph according to a walk. That is, when the game starts she picks a starting position, and from that point onwards, in each round she must either claim a free edge incident to her current position, or walk along an edge that she claimed earlier in the game.

Afterwards the other endpoint of that edge becomes the new position of Walker. These games were introduced by Espig, Frieze, Krivelevich, and Pegden [29], and further studied by Clemens and Tran [23] as well as by Forcan and Mikalački (see e.g. [35], [36]), amongst others. For the $(2:b)$ Walker-Breaker connectivity and Hamiltonicity game on K_n, Forcan and Mikalački [36] were able to show that the threshold bias is again of order $\frac{n}{\ln(n)}$, which behaves similarly to the $(2:b)$ Connector-Breaker variant discussed above.

Let us briefly mention that Connector-Breaker games and Walker-Breaker games are quite similar in the sense that any strategy for Breaker in a Connector-Breaker game can be applied in the same way in the corresponding Walker-Breaker game, and similarly any strategy for Walker in a Walker-Breaker game can be applied in the same way for Connector in a Connector-Breaker game (where Connector just skips any move in which Walker does not claim a new edge).

1.3 Main results

This section serves as an overview of the main results of this thesis, as well as other important results for reference. Proofs of our results will be provided in the subsequent chapters. We will start with our results for fast strategies in Waiter-Client games on K_n and afterwards we will state our results for Connector-Breaker and Walker-Breaker games on $G \sim G_{n,p}$.

1.3.1 Fast strategies in Waiter-Client games

Perfect matching game. As stated earlier, the perfect matching game is an easy win for Maker in the unbiased Maker-Breaker game on $E(K_n)$. Let $\mathcal{PM}_n$ denote the family of all perfect matchings of K_n. Hefetz, Krivelevich, Stojaković, and Szabó [40] proved that $\tau_{MB}(\mathcal{PM}_n, 1) = \frac{n}{2} + 1$ for large enough even n, thus showing that the unbiased perfect matching game is won asymptotically fast by Maker. In fact, at the moment when Maker wins the game, her graph consists of a perfect matching and at most one *wasted* edge. We will show that the same is true for the unbiased Waiter-Client game on K_n.

Theorem 1.3.1. *For every large enough even integer n the following holds:*

$$\tau_{WC}(\mathcal{PM}_n, 1) = \frac{n}{2} + 1\,.$$

We will also study the biased version of this game. Mikalački and Stojaković [57] proved that even for $b = O(\frac{n}{\ln(n)})$ the biased Maker-Breaker perfect matching game is won asymptotically fast by Maker. Note that the bound on b is best possible (in its order of magnitude), as for $b > (1+o(1))\frac{n}{\ln(n)}$ Breaker has a strategy to isolate a vertex in Maker's graph [16].

Theorem 1.3.2 (Theorem 1.1 and Theorem 1.5(i) in [57])**.** *There exist constants $\delta, c, C > 0$ such that for every large enough n and for every $b \leqslant \delta\frac{n}{\ln(n)}$ the following holds:*

$$\frac{n}{2} + cb \leqslant \tau_{MB}(\mathcal{PM}_n, b) \leqslant \frac{n}{2} + Cb\ln(b)\,.$$

The question of how fast Maker can win if Breaker has an even larger bias was recently studied by Brustle, Clusiau, Narayan, Ndiaye, Reed, and Seamone [14]. They were able to prove the following result.

Theorem 1.3.3 (Theorem 1.1 in [14])**.** *For every f which is $\omega(1)$ and for n large enough, if $b < \frac{n}{\ln(n)} - \frac{f(n)n}{(\ln(n))^{5/4}}$ the following holds:*

$$\tau_{MB}(\mathcal{PM}_n, b) \leqslant \frac{n}{2} + o(1)n\,.$$

Our main result for the Waiter-Client version of that game shows that in this setting for a fast strategy we can even allow the bias to be linear in n, while at the same time being able to win even faster than in the Maker-Breaker setting.

Theorem 1.3.4. *There exist constants $\delta, C > 0$ such that for every large enough even n and for every $b \leqslant \delta n$ the following holds:*

$$\tau_{WC}(\mathcal{PM}_n, b) \leqslant \frac{n}{2} + Cb\,.$$

Hamiltonicity game. As we already mentioned earlier, Hefetz and Stich [45] proved that $\tau_{MB}(\mathcal{H}_n, 1) = n + 1$ holds. We will show that Waiter can win the Waiter-Client version of the unbiased Hamiltonicity game in the same number of rounds.

Theorem 1.3.5. *For every large enough integer n the following holds:*

$$\tau_{WC}(\mathcal{H}_n, 1) = n + 1\,.$$

The biased version of the Maker-Breaker Hamiltonicity game was first considered by Mikalački and Stojaković [57]. For Breaker's side they were able to show that Breaker can postpone Maker's win by some time depending on Breaker's bias.

Theorem 1.3.6 (Theorem 1.5(ii) in [57])**.** *For every $b \geqslant 1$ and for every large enough n the following holds:*

$$\tau_{MB}(\mathcal{H}_n, b) \geqslant n + \frac{b}{2}\,.$$

For Maker's side, they were able to prove the following results, depending on Breaker's bias.

Theorem 1.3.7 (Theorem 1.2 in [57])**.** *There exists a constant $C > 0$ such that for every large enough n and for $b \geqslant 2$ and $b = o\left(\frac{\ln(n)}{\ln(\ln(n))}\right)$ the following holds:*

$$\tau_{MB}(\mathcal{H}_n, b) \leqslant n + Cb^2 \ln(b)\,.$$

Theorem 1.3.8 (Theorem 1.3 in [57])**.** *There exist constants $\delta, C > 0$ such that for every large enough n and for every $b \leqslant \delta\sqrt{\frac{n}{\ln^5(n)}}$ the following holds:*

$$\tau_{MB}(\mathcal{H}_n, b) \leqslant n + Cb^2 \ln^5(b)\,.$$

For an even larger bias, a result by Krivelevich [49] is applicable, which can be formulated as follows:

Theorem 1.3.9 (Theorem 1 in [49])**.** *For every large enough n and for every $b \leqslant \left(1 - \frac{30}{\ln^{1/4}(n)}\right)\frac{n}{\ln(n)}$ the following holds:*

$$\tau_{MB}(\mathcal{H}_n, b) \leqslant 14n\,.$$

More recently, Brustle, Clusiau, Narayan, Ndiaye, Reed, and Seamone [15] improved this result even further. They were able to allow even larger biases for Breaker, and at the same time they were able to show the following upper bound on the number of turns in which Maker is able to win the biased Hamiltonicity game.

Theorem 1.3.10 (Theorem 1.1 in [15])**.** *There exists a constant C such that for every large enough n and for every $b < \frac{n}{\ln(n)} - \frac{Cn}{(\ln(n))^{3/2}}$ the following holds:*

$$\tau_{MB}(\mathcal{H}_n, b) \leqslant n + \frac{Cn}{(\ln(n))^{1/2}}\,.$$

For the biased Waiter-Client Hamiltonicity game, we could even improve these bounds. We were able to show a better upper bound on the number of rounds, while again (as in the perfect matching game) allowing the bias to be of linear size.

Theorem 1.3.11. *There exist constants $\delta, C > 0$ such that for every large enough n and for every $b \leqslant \delta n$ the following holds:*

$$\tau_{WC}(\mathcal{H}_n, b) \leqslant n + Cb\,.$$

Pancyclicity game. Quite recently (in [9] and [33]) it was suggested to generalise the Hamiltonicity game even further by choosing the winning sets to be all *pancyclic* subgraphs of K_n, meaning all subgraphs containing cycles of all possible lengths between 3 and n. Denote with $\mathcal{PC}_n$ the family of all such subgraphs. Ferber, Krivelevich, and Naves [33] proved that for $b = o(\sqrt{n})$ the b-biased Maker-Breaker pancyclicity game on K_n is won by Maker, while it was already known before that for $b \geqslant 2\sqrt{n}$ Breaker wins, since he can block all triangles [16]. In contrast to this result, it was shown by Bednarska-Bzdęga, Hefetz, Krivelevich, and Łuczak [9], that the threshold bias in the corresponding Waiter-Client game is linear in n. Apart from that, not much is known for games with $\mathcal{PC}_n$ being the family of winning sets. In particular, no tight result on the number of rounds has been proven yet. In this thesis we prove the following result.

Theorem 1.3.12. *In the unbiased Waiter-Client pancyclicity game the following holds:*

$$\tau_{WC}(\mathcal{PC}_n, 1) = n + (1 + o(1))\log_2(n)\,.$$

Note that this means that Waiter wins almost perfectly fast, as every spanning pancyclic subgraph of K_n contains at least $n + (1 - o(1))\log_2(n)$ edges [13]. Moreover, the second order term in the above theorem will be made even more precise later on (see also the remark at the end of Section 3.3).

Connectivity and fixed spanning trees. The aforementioned connectivity game on K_n is also easily won by Maker in the unbiased setting. Indeed, we already discussed that for large enough n, Maker even has a strategy to create a Hamilton cycle asymptotically fast. Moreover, as was also mentioned earlier, since there is no reason for Maker to close cycles in the connectivity game, following the earlier mentioned result by Lehman [54], Maker can win the unbiased connectivity game perfectly fast within $n-1$ rounds.

Due to the simplicity of the aforementioned game, Ferber, Hefetz, and Krivelevich [32] introduced a variant of the connectivity game in which Maker aims to occupy a copy of some given spanning tree T. Obviously, in order to have a winning strategy for Maker, we cannot choose T completely arbitrarily, since Breaker can block large stars. Thus, it is natural to put some degree constraints on the desired tree T. Let $\mathcal{F}_T$ denote the family of all copies of T in K_n. Ferber, Hefetz, and Krivelevich [32] first proved the following result.

Theorem 1.3.13 (Theorem 1.1 in [32]). *Let $\varepsilon > 0$. Then for every large enough integer n the following holds. Let T be any spanning tree on n vertices, with maximum degree $\Delta(T) \leqslant n^{0.05-\varepsilon}$, and let $b \leqslant n^{0.005-\varepsilon}$ be any positive integer. Then*

$$\tau_{MB}(\mathcal{F}_T, b) = n + o(n)\,.$$

Thus, even when the maximum degree and Breaker's bias are increasing with n, Maker has a strategy to win the fixed spanning tree game asymptotically fast. Naturally, one may wonder whether the error term in the above theorem can be improved when we put stronger constraints on $\Delta(T)$ and b. This question was answered by Clemens, Ferber, Glebov, Hefetz, and Liebenau [17] as follows.

Theorem 1.3.14 (Theorem 1.1 and Theorem 1.4 in [17]). *Let Δ be a positive integer, then for every large enough integer n the following holds. For every spanning tree T on n vertices with maximum degree $\Delta(T) \leqslant \Delta$, we have*

$$n-1 \leqslant \tau_{MB}(\mathcal{F}_T, 1) \leqslant n+1\,.$$

Moreover, if T is a tree chosen uniformly at random among all labelled trees on n vertices (not necessarily having a constant bound on the maximum degree), then a.a.s.

$$\tau_{MB}(\mathcal{F}_T, 1) = n-1\,.$$

That is, for most choices of T, Maker wins the tree embedding game perfectly fast.

In this thesis we will show that in the unbiased Waiter-Client game, Waiter has a fast winning strategy in the game $(E(K_n), \mathcal{F}_T)$ with at most one wasted edge. Moreover, in contrast to the above theorems we may also allow the maximum degree to grow much faster with n.

Theorem 1.3.15. *There exists a constant $\varepsilon > 0$ such that the following holds for every large enough integer n. Let T be a spanning tree of K_n with $\Delta(T) \leqslant \varepsilon\sqrt{n}$, then*

$$n-1 \leqslant \tau_{WC}(\mathcal{F}_T, 1) \leqslant n\,.$$

Moreover, the lower and the upper bounds are tight, surprisingly also for $\Delta(T) \leqslant 3$.

Let us also mention that there exist some very recent results about spanning trees with an even larger maximum degree as well. In [2] we could show, that Waiter in the unbiased Waiter-Client game on K_n can even force any spanning tree of maximum degree roughly $\frac{1}{3}n$, and she can even do so fast (albeit not as fast as for trees with smaller maximum degree). To make this more precise, let us state the following theorem.

Theorem 1.3.16 (Theorem 1.2 in [2])**.** *For every $\varepsilon \in \left(0, \frac{1}{3}\right)$ there exists a constant c such that the following holds for every large enough integer n. Let T be a spanning tree of K_n with $\Delta(T) \leqslant \left(\frac{1}{3} - \varepsilon\right)n$, then*

$$\tau_{WC}(\mathcal{F}_T, 1) \leqslant n + c\sqrt{n}\,.$$

***H*-factor game.** Using our methods from the fixed spanning tree game, we are also able to describe fast winning strategies for games in which Waiter aims to create a factor of a fixed constant size tree. Note that the same kind of question was studied in the Maker-Breaker setting by Clemens and Mikalački, but only in the case when the fixed tree is either a path or a star [22].

Let us define a factor of a fixed graph more precisely: for a fixed graph H and an integer n satisfying $v(H)|n$, an *H-factor* of K_n is defined to be the vertex disjoint union of copies of H covering all vertices of K_n. Let $\mathcal{F}_{n,H-fac}$ be the family of all such subgraphs. We prove the following result:

Theorem 1.3.17. *Let $k \geqslant 2$ be a positive integer and let T be any fixed tree on k vertices. Provided that n is a large enough integer with $k|n$, the following holds:*

$$\frac{k-1}{k}n \leqslant \tau_{WC}(\mathcal{F}_{n,T-fac}, 1) \leqslant \frac{k-1}{k}n + 1\,.$$

Moreover, the lower and the upper bound are tight.

Observe that in every game considered so far, Waiter can always win at least asymptotically as fast as possible. We finish Chapter 3 by giving an example where this is not the case. We consider the *triangle factor game*, whose Maker-Breaker version has been discussed in [3] and [32]. Krivelevich and Szabó observed, that Maker cannot win the unbiased triangle factor game on K_n within less than $\frac{7}{6}n$ rounds (a proof is contained in [32]).

Theorem 1.3.18 (Theorem 4.3 in [32])**.** *For every large enough integer n such that $3|n$ the following holds:*

$$\frac{7}{6}n \leqslant \tau_{MB}(\mathcal{F}_{n,K_3-fac}, 1)\,.$$

We show that the triangle factor game cannot be won asymptotically as fast as possible, but that this game can still be won by Waiter at least (asymptotically) as fast as Maker can win in the corresponding Maker-Breaker version.

Theorem 1.3.19. *For every large enough integer n such that $3|n$ the following holds:*

$$\frac{13}{12}n \leqslant \tau_{WC}(\mathcal{F}_{n,K_3-fac}, 1) \leqslant \frac{7}{6}n + o(n)\,.$$

1.3.2 Connector-Breaker and Walker-Breaker

The topics of the final chapter of this thesis are Connector-Breaker and Walker-Breaker games. In particular, we will consider the connectivity game and the Hamiltonicity game of both variants when played on the edges of a random graph $G \sim G_{n,p}$. We already mentioned in Subsection 1.2.7 that when playing on the edges of some graph G, in the unbiased setting the winner of the game depends on whether a specific graph (which we called H_n) appears as a subgraph in G. Since we would like the bias to be balanced between both players but $G_{n,p}$ requires p to be almost 1 for H_n to appear as a subgraph, we will consider the aforementioned games in the $(2:2)$ setting instead. Our goal is to find the respective threshold probabilities for all versions of these games. Before we continue, let us define the following four threshold probabilities to simplify notation.

- $p^*_{CB}(\mathcal{C}_n)$ for the $(2:2)$ Connector-Breaker connectivity game on $G \sim G_{n,p}$
- $p^*_{CB}(\mathcal{H}_n)$ for the $(2:2)$ Connector-Breaker Hamiltonicity game on $G \sim G_{n,p}$
- $p^*_{WB}(\mathcal{C}_n)$ for the $(2:2)$ Walker-Breaker connectivity game on $G \sim G_{n,p}$
- $p^*_{WB}(\mathcal{H}_n)$ for the $(2:2)$ Walker-Breaker Hamiltonicity game on $G \sim G_{n,p}$

Surprisingly, all of these threshold probabilities not just differ from the Maker-Breaker variants of the respective games, but they are in fact close to each other:

Theorem 1.3.20. *For large enough n the following holds:*

$$p^*_{CB}(\mathcal{C}_n),\ p^*_{CB}(\mathcal{H}_n),\ p^*_{WB}(\mathcal{C}_n),\ p^*_{WB}(\mathcal{H}_n) = n^{-2/3+o(1)}.$$

This result also provides an answer to the question by London and Pluhár whether the $(2:2)$ Connector-Breaker connectivity game on a random graph $G \sim G_{n,p}$ behaves similarly to the $(1:1)$ Maker-Breaker version of this game. Even if Connector's bias is increased, a much denser random graph is necessary than in the Maker-Breaker version of this game for Connector to have a chance to win the connectivity game almost surely.

In this thesis we will only focus on Breaker's side in the Connector-Breaker connectivity game as well as Walker's side in the Walker-Breaker Hamiltonicity game. Let us state the following two theorems next (proofs of which can be found in Chapter 4) and afterwards give a short explanation why Theorem 1.3.20 follows from these.

Theorem 1.3.21. *Let $\varepsilon \in (0,1)$. Then, for $p \leqslant n^{-2/3-\varepsilon}$, playing a $(2:2)$ Connector-Breaker game on the edges of a random graph $G \sim G_{n,p}$, Breaker a.a.s. has a strategy to keep a vertex isolated in Connector's graph.*

Theorem 1.3.22. *Let $\varepsilon \in (0,1)$. Then, for $p \geqslant n^{-2/3+\varepsilon}$, playing a $(2:2)$ Walker-Breaker game on the edges of a random graph $G \sim G_{n,p}$, Walker a.a.s. has a strategy to claim a Hamilton cycle in G.*

These results are optimal up to the constant ε in the exponent. Since for $p \leqslant n^{-2/3-o(1)}$ Breaker can block any spanning structure in the $(2:2)$ Connector-Breaker game on $G \sim G_{n,p}$, the lower bound on the threshold probability for all mentioned games immediately follows. On the other hand, since for $p \geqslant n^{-2/3+o(1)}$ Walker has a strategy to claim a Hamilton cycle in the $(2:2)$ Walker-Breaker game on $G \sim G_{n,p}$ (which also contains a spanning tree of G), the upper bound on the threshold probability for all mentioned games immediately follows. Let us also note, that an independent proof of Connector's strategy in the $(2:2)$ Connector-Breaker connectivity game exists as well and can be found in [21].

Lastly, let us note that in fact we can prove an even slightly stronger result than Theorem 1.3.20. One of the substrategies in our main strategy in the proof of Theorem 1.3.22 uses an approach of Ferber, Krivelevich, and Naves [33], which allows Walker to create a graph that behaves almost like a typical random graph. This will result in the proof of an even stronger statement in Subsection 4.2.3, which requires the concept of local resilience. This concept will be introduced in Section 2.4. More details can be found in the aforementioned (sub-)sections.

1.4 Organisation

In Chapter 2 we will provide an overview of the notation that is used throughout this thesis. Additionally, we will introduce the most important tools that we use repeatedly in the proofs of our main results, such as Beck's winning criterion, a variety of box games, useful inequalities by Chernoff and Markov, and a result about local resilience for random graphs.

In Chapter 3 we will look at various Waiter-Client games and fast strategies for Waiter to win in as few rounds as possible. The games we consider are the unbiased and biased perfect matching and Hamiltonicity game, as well as the unbiased pancyclicity game, the unbiased fixed spanning tree game, the unbiased H-factor game for a fixed tree T of constant size, and lastly the unbiased triangle factor game. The results of this chapter are joint work with Dennis Clemens, Pranshu Gupta, Fabian Hamann, Alexander Haupt, and Mirjana Mikalački [18].

Chapter 4 is split into two main sections. In Section 4.1 we consider Breaker's side in the $(2:2)$ Connector-Breaker connectivity game on a random graph $G \sim G_{n,p}$. We will define so-called bad vertices, which we will then merge into so-called bad structures, and state an algorithm how to find such sets of vertices in a random graph. We will then give a strategy for Breaker that uses one of these bad structures to isolate a vertex in Connector's graph. The results are joint work with Dennis Clemens and Laurin Kirsch [21].

Afterwards, in Section 4.2, we consider Walker's side in the $(2:2)$ Walker-Breaker Hamiltonicity game on a random graph $G \sim G_{n,p}$. Here we will define so-called good structures which Walker can use to reach any vertex of a random graph in a constant number of turns. With this ability at hand we will then give a strategy for Walker that she can use to claim the edges of a graph containing a Hamilton cycle. The results are joint work with Dennis Clemens and Pranshu Gupta [19].

Chapter 2

Notation and Techniques

This chapter serves as an overview of the notation and helpful techniques used throughout this thesis. In Section 2.1 we will state relevant notation, and will give an overview of useful auxiliary tools for positional games in Section 2.2. Then we will state many helpful probabilistic tools in Section 2.3, and we will finish this chapter with results about local resilience in random graphs in Section 2.4.

2.1 Notation

The game-theoretic and graph-theoretic notation in this thesis is rather standard and most of the time follows the notation of [42] and [62].

For a positive integer n, we set $[n] := \{k \in \mathbb{N} : 1 \leqslant k \leqslant n\}$, and for an interval of integers we define $[a, b] := \{k \in \mathbb{N} : a \leqslant k \leqslant b\}$.

For a graph $G = (V, E)$ we write $V(G)$ and $E(G)$ for the vertex set and the edge set of G, respectively, and set $v(G) = |V(G)|$ and $e(G) = |E(G)|$. If $\{v, w\} \in E(G)$ is an edge, we denote it with vw for short, and we call w a *neighbour* of v in G. We set $N_G(v) = \{w \in V(G) : vw \in E(G)\}$ to be the *neighbourhood* of v in G, and call $d_G(v) = |N_G(v)|$ the *degree* of v in G. Moreover, $\Delta(G) = \max_{v \in V(G)} d_G(v)$ denotes the *maximum degree* of G and $\delta(G) = \min_{v \in V(G)} d_G(v)$ denotes the *minimum degree* of G. We write $G - v$ for the graph obtained from G by deleting the vertex v and all its incident edges. Whenever it is clear which graph is being looked at, we may omit the subscript G in the above notation.

Given any two subsets $A, B \subset V(G)$ and any vertex $v \in V(G)$, let $N_G(v, A) = N_G(v) \cap A$ denote the neighbourhood of v in A, and we set $d_G(v, A) = |N_G(v, A)|$ to be the degree of v into A. Moreover, we define $N_G(A) := \bigcup_{v \in A} N_G(v)$, $E_G(A, B) := \{vw \in E(G) : v \in A, w \in B\}$, $e_G(A, B) := |E_G(A, B)|$, $E_G(A) := E_G(A, A) = \{vw \in E(G) : v, w \in A\}$, and $e_G(A) := |E_G(A)|$.

Let two graphs H and G be given. If both $V(H) \subset V(G)$ and $E(H) \subset E(G)$ hold, we call H a *subgraph* of G, and we write $H \subset G$ for short. We also set $G \backslash H = (V(G), E(G) \backslash E(H))$ in this case. Given any $A \subset V(G)$, we call $G[A] = (A, E_G(A))$ the subgraph of G induced by A.

If there is a bijection $f : V(H) \to V(G)$ such that $vw \in E(H)$ holds if and only if $f(v)f(w) \in E(G)$ holds, the two graphs H and G are called *isomorphic* (denoted with $H \cong G$), and we also say that H is a *copy* of G in this case.

We represent a *path* P by a sequence $(v_1, v_2, \ldots, v_k)$, which means that $V(P) = \{v_1, v_2, \ldots, v_k\}$ and $E(P) = \{v_i v_{i+1} : 1 \leqslant i \leqslant k-1\}$ hold. Similarly, a *cycle* C is represented by a sequence $(v_1, v_2, \ldots, v_k)$, which means that $V(C) = \{v_1, v_2, \ldots, v_k\}$ and $E(C) = \{v_i v_{i+1} : 1 \leqslant i \leqslant k-1\} \cup \{v_k v_1\}$ hold. The length of both a path and a cycle is always the number of its edges. Let us also mention one special path: A *rooted cherry* is a path of length 2 where the root is its middle vertex.

Assume that some positional game, played on the edge set of some graph G, is in progress. Depending on the game, we denote the graphs consisting of either player's edges as follows:

- *Waiter-Client game:* Let W denote the graph consisting of Waiter's edges, and let C denote the graph consisting of Client's edges. For short, we also set $V_W = V(W)$, $E_W = E(W)$, and $E_C = E(C)$.

- *Connector-Breaker game:* Let C denote the graph consisting of Connector's edges, and let B denote the graph consisting of Breaker's edges. For short, we also set $V_C = V(C)$, $E_C = E(C)$, and $E_B = E(B)$.

- *Walker-Breaker game:* Let W denote the graph consisting of Walker's edges, and let B denote the graph consisting of Breaker's edges. For short, we also set $V_W = V(W)$, $E_W = E(W)$, and $E_B = E(B)$.

Any edge belonging to either player's graph is said to be *claimed*, while all the other edges of G are called *free*; the set of all free edges is denoted by F. Additionally, in the context of Walker-Breaker games, we say that an edge is available if it belongs to $F \cup W$.

Given a distribution $\mathcal{D}$ and a random variable X, we write $X \sim \mathcal{D}$ for X being sampled according to the distribution $\mathcal{D}$. With $\text{Bin}(n, p)$ we denote the binomial distribution with parameters n and p. Moreover, with $G_{n,p}$ we denote the Erdős-Renyi random graph model on n vertices and with edge probability p. If X is a random variable, we let $\mathbb{E}(X)$ denote its expectation. If $\mathcal{E}$ is an event, we let $\mathbb{P}(\mathcal{E})$ denote its probability. A sequence of events $\mathcal{E}_n$ is said to hold *asymptotically almost surely* (for short we write a.a.s.) if $\mathbb{P}(\mathcal{E}_n) \to 1$ for $n \to \infty$.

We use basic Landau notation, which means that for functions $f, g : \mathbb{N} \to \mathbb{R}$ we write $f(n) = o(g(n))$ if $\lim_{n\to\infty} \left|\frac{f(n)}{g(n)}\right| = 0$, $f(n) = O(g(n))$ if $\limsup_{n\to\infty} \left|\frac{f(n)}{g(n)}\right| < \infty$, $f(n) = \omega(g(n))$ if $\lim_{n\to\infty} \left|\frac{g(n)}{f(n)}\right| = 0$, and $f(n) = \Theta(g(n))$ if $f = O(g(n))$ and $g = O(f(n))$.

All of the main results in this thesis are asymptotic. Whenever necessary, we will assume n to be large enough. We will not optimize constants, and we will omit rounding signs whenever these are not crucial. To simplify some calculations, we use the following notation: Let $f : \mathbb{N} \to \mathbb{R}$ and $g : \mathbb{R} \times \mathbb{N} \to \mathbb{R}$ be functions such that g is monotone in the first variable. Then for every $n \in \mathbb{N}$ and $\alpha \in \mathbb{R}$ we write $f(n) = g(\pm\alpha, n)$ to say that $g(-\alpha, n) \leqslant f(n) \leqslant g(+\alpha, n)$. For instance, we would write $f(n) = \ln^{\pm\alpha}(n)$ instead of $\ln^{-\alpha}(n) \leqslant f(n) \leqslant \ln^{+\alpha}(n)$.

2.2 Tools for positional games

In this section we will introduce auxiliary tools which will often be used in proofs in later chapters. First we will state Beck's famous winning criterion, followed by a collection of different box games.

2.2.1 Beck's winning criterion

One ingredient for many strategies in biased Maker-Breaker games is Beck's winning criterion, which reads as follows:

Theorem 2.2.1 (Theorem 1 in [7]). *Let $a, b \in \mathbb{N}$, and let $(\mathcal{X}, \mathcal{F})$ be a hypergraph such that*

$$\sum_{F \in \mathcal{F}} (1+b)^{-|F|/a} < \frac{1}{b+1},$$

then Breaker has a winning strategy for the $(a:b)$ Maker-Breaker game $(\mathcal{X}, \mathcal{F})$.

This theorem is applied in Walker's strategy in Subsection 4.2.3, albeit in a slightly untypical way.

2.2.2 Maker-Breaker Box game

Another simple yet very useful tool is the following positional game, introduced by Chvátal and Erdős [16], which is called the *Box game* and is usually helpful to describe strategies that aim to bound the degrees in the opponent's graph. The game $\text{Box}(p, 1; a_1, \ldots, a_n)$ is played on a hypergraph $(X, \mathcal{H})$, with $\mathcal{H} = \{F_1, \ldots, F_n\}$ consisting of n pairwise disjoint hyperedges (called *boxes*), satisfying $|F_i| = a_i$ for every $i \in [n]$. In every round, *BoxMaker* claims at most p elements from X that have not been claimed before, while *BoxBreaker* solely claims one such element. If throughout the game BoxMaker succeeds in claiming all the elements of a box F_i, she is declared the winner of the game. Otherwise, i.e. when BoxBreaker succeeds in claiming at least one element in each box, BoxBreaker wins. The following lemma is a well-known criterion for BoxBreaker to have a winning strategy in the Box game (see e.g. [16], [42]).

Lemma 2.2.2. *Let $a_i = m$ for every $i \in [n]$, $p \geqslant 1$, and assume that $m > p(\ln(n) + 1)$, then BoxBreaker wins the game $\text{Box}(p, 1; a_1, \ldots, a_n)$.*

A winning strategy $\mathcal{S}$ for Breaker is the following one: in every round, BoxBreaker claims an element which belongs to a box that he does not have an element from and which, among all such boxes, contains the largest number of Maker's elements. In fact, the above lemma is Theorem 3.4.1 in [42], while the mentioned strategy is contained in its proof. As an immediate corollary of the above lemma we obtain the following:

Corollary 2.2.3. *Let BoxMaker and BoxBreaker play the game $\text{Box}(p, 1; a_1, \ldots, a_n)$ with boxes F_i of size $|F_i| = a_i \geqslant m$. Then following the strategy $\mathcal{S}$ from Lemma 2.2.2, BoxBreaker can guarantee that the following holds for every $i \in [n]$ throughout the game: as long as he does not claim an element in F_i, the number of BoxMaker's elements in F_i is bounded by $p(\ln(n) + 1)$.*

2.2.3 Continuous Box game

Another variant of the Box game we use is the *Continuous Box game* defined as follows. The game $\text{CBox}(b, 1; a_1, \ldots, a_n)$ is played with n pairwise disjoint boxes F_i, each with positive real weight a_i. The game is played between *CMaker* and *CBreaker*. In every round, CMaker claims weights from the boxes such that the total sum of the claimed weights is at most b, while CBreaker solely removes one box from the game (we may also say that CBreaker destroys that box). If throughout the game CMaker succeeds in claiming all the weight of a box, she is declared the winner of the game. Otherwise, i.e. when CBreaker succeeds in destroying all boxes, he wins. Similar to the Box game, let $\mathcal{S}^*$ be a strategy for CBreaker where he always destroys a box in which CMaker has claimed the largest weight.

The following lemma is an easy consequence of the results from [43].

Lemma 2.2.4. *Let CMaker and CBreaker play the game* $\text{CBox}(b, 1; a_1, \ldots, a_n)$ *with boxes* F_i *of size* $|F_i| = a_i$. *Then following the strategy* $\mathcal{S}^*$, *CBreaker can ensure that the following holds throughout the game: If* F_i *is a box which is still not destroyed by CBreaker, then the weight claimed by CMaker in box* F_i *is at most* $b(\ln(n) + 1)$.

2.2.4 MinBox game

The final variant of the Box game we want to mention is the *MinBox game*, which was introduced by Ferber, Krivelevich, and Naves in [33] and motivated by Gebauer and Szabó in [37]. Let positive integers n, D, b, and a real $\alpha \in (0, 1)$ be given. The game $\text{MinBox}(n, D, \alpha, b)$ is a $(1 : b)$ Maker-Breaker game played on a family of n pairwise disjoint boxes $F_1, \ldots, F_n$, each of size at least D, where Maker wins if and only if she occupies at least an α-fraction of elements in each of the n boxes. To analyse the game, let us use the following definitions throughout the game: For each box F, let $w_M(F)$ and $w_B(F)$ denote the number of elements that Maker and Breaker have claimed in F, respectively, and define the *danger* of a box as $\text{dang}(F) := w_B(F) - b \cdot w_M(F)$. Call F free if not every element of F is claimed yet, and call it active if $w_M(F) < \alpha|F|$. In [33] the following was proven.

Theorem 2.2.5 (Theorem 2.3 in [33])**.** *Let* $n, b, D \in \mathbb{N}$ *and* $\alpha \in (0, 1)$. *Assume that in the game* $\text{MinBox}(n, D, \alpha, b)$ *Maker plays as follows: In each turn, first Maker chooses an arbitrary free active box* F *with largest danger, and then she claims a free element of* F. *Then, throughout the game* $\text{dang}(F) \leqslant b(\ln(n) + 1)$ *holds for every active box* F.

2.3 Probabilistic tools and basic properties of $G_{n,p}$

In this section we present a few bounds on large deviations of random variables that will be used to identify typical edge distributions in a random graph $G \sim G_{n,p}$. Most of the time, we will use the following inequalities due to Chernoff (see e.g. [4], [46]).

Lemma 2.3.1. *If* $X \sim \mathrm{Bin}(n,p)$*, then*

- $\mathbb{P}\left(X < (1-\delta)np\right) < \exp\left(-\frac{\delta^2 np}{2}\right)$ *for every* $\delta > 0$*, and*
- $\mathbb{P}\left(X > (1+\delta)np\right) < \exp\left(-\frac{np}{3}\right)$ *for every* $\delta \geqslant 1$*.*

Lemma 2.3.2. *Let* $X \sim \mathrm{Bin}(n,p)$ *with expectation* $\mu = \mathbb{E}(X)$*, and let* $k \geqslant 7\mu$*, then*

$$\mathbb{P}\left(X \geqslant k\right) \leqslant e^{-k}.$$

These inequalities are useful to verify that a given binomial random variable $X \sim \mathrm{Bin}(n,p)$, where each of n independent rounds has probability p of being successful, is typically concentrated around its expectation $\mathbb{E}(X) = np$. To give an example and as a first application, let us prove the following claim, which gives an upper and a lower bound on the degree of every vertex in $G_{n,p}$. We will use this result later on in Chapter 4 when we analyse Walker's strategy.

Claim 2.3.3. *Let* $\varepsilon \in (0,1)$*,* $p \geqslant n^{-2/3}$*, and* $G \sim G_{n,p}$*. Then a.a.s.* $d_G(v) = (1 \pm \varepsilon)np$ *for every vertex* $v \in V(G)$*.*

Proof. For every $v \in V(G)$ we have $d_G(v) \sim \mathrm{Bin}(n-1,p)$ with $\mathbb{E}(d_G(v)) = (n-1)p \sim np$. Let us consider the upper bound first. Applying Lemma 2.3.1 we find that for each vertex the probability that its degree is too large is bounded by

$$\mathbb{P}\left(d_G(v) > (1+\varepsilon)np\right) \leqslant \exp\left(-\frac{1}{3}n^{1/3}\right).$$

To prove that this event is not likely to fail for any of the vertices of G, we take a union bound over all possible vertices $v \in V(G)$:

$$\begin{aligned}\mathbb{P}\left(\exists v \in V(G):\ d_G(v) > (1+\varepsilon)np\right) &< n \cdot \exp\left(-\frac{1}{3}n^{1/3}\right) = \exp\left(-\frac{1}{3}n^{1/3} + \ln(n)\right)\\ &< \exp\left(-n^{1/4}\right) = o(1).\end{aligned}$$

The proof of the lower bound is similar. Applying Lemma 2.3.1 we get that

$$\mathbb{P}\left(d_G(v) < (1-\varepsilon)np\right) \leqslant \exp\left(-\frac{1}{2}\varepsilon^2 n^{1/3}\right).$$

Again, we take a union bound over all possible vertices $v \in V(G)$, and hence we get that

$$\begin{aligned}\mathbb{P}\left(\exists v \in V(G):\ d_G(v) < (1-\varepsilon)np\right) &< n \cdot \exp\left(-\frac{1}{2}\varepsilon^2 n^{1/3}\right) = \exp\left(-\frac{1}{2}\varepsilon^2 n^{1/3} + \ln(n)\right)\\ &< \exp\left(-n^{1/4}\right) = o(1).\end{aligned}$$

Thus the claim follows. □

Let us also prove the following simple statement, which will be used in Breaker's strategy in Chapter 4. The proof is similar to the proof of Claim 2.3.3 and helps us to upper bound the degree in a random graph.

Claim 2.3.4. *Let $\varepsilon \in (0,1)$, $p = n^{-2/3-\varepsilon}$ and let $G \sim G_{n,p}$. Then with probability at least $1 - \exp(-n^{1/3-2\varepsilon})$ every vertex $v \in V(G)$ satisfies*

$$d_G(v) \leqslant 2n^{\frac{1}{3}-\varepsilon}. \tag{2.3.1}$$

Proof. For every $v \in V(G)$ we have $d_G(v) \sim \text{Bin}(n-1,p)$ with $\mathbb{E}(d_G(v)) = (n-1)p \sim np$. Applying Lemma 2.3.1 we deduce that

$$\mathbb{P}\left(d_G(v) > 2np\right) \leqslant \exp\left(-\frac{1}{3}n^{1/3-\varepsilon}\right).$$

The claim follows from taking a union bound over all possible vertices $v \in V(G)$:

$$\mathbb{P}\left(\exists v \in V(G): \ d_G(v) > 2np\right) < n \cdot \exp\left(-\frac{1}{3}n^{1/3-\varepsilon}\right) = \exp\left(-\frac{1}{3}n^{1/3-\varepsilon} + \ln(n)\right)$$
$$< \exp\left(-n^{1/3-2\varepsilon}\right).$$

□

Now let us also mention the well-known Markov inequality (see e.g. [46]), another useful inequality, which will also be applied in Chapter 4 for Breaker's strategy.

Lemma 2.3.5. *Let $X \geqslant 0$ be a random variable. For every $t > 0$ it holds that*

$$\mathbb{P}\left(X \geqslant t\right) \leqslant \frac{\mathbb{E}(X)}{t}.$$

2.4 Local resilience for random graphs

In Subsection 4.2.3, Walker's strategy uses a randomized substrategy similar to the one used by Ferber, Krivelevich, and Naves in [33], because we want to be able to relate Walker's graph to local resilience properties of random graphs. For this, we use the following definition. More details will follow with the proof of Theorem 4.2.7 in Subsection 4.2.3.

Definition 2.4.1. Let $\mathcal{P} = \mathcal{P}(n)$ be a monotone increasing graph property and $\varepsilon, p \in (0,1)$. Then we say that $\mathcal{P}$ is (p,ε)*-resilient* if a random graph $G \sim G_{n,p}$ a.a.s. satisfies the following: For every subgraph $G' \subseteq G$ such that $d_{G'}(v) \leqslant \varepsilon d_G(v)$ holds for every $v \in V(G)$, it is true that $G \backslash G' \in \mathcal{P}$.

This basically means that if a random graph has a certain property, even if we delete a few edges at every vertex, the resulting graph will still have that property. The Main Theorem for Walker (Theorem 1.3.22) will then follow from the more general Theorem 4.2.7 together with the following theorem of Lee and Sudakov [53], which says that the containment of a Hamilton cycle is a resilient graph property.

Theorem 2.4.2 (Theorem 1.1 in [53])**.** *For $n \in \mathbb{N}$, let $\mathcal{H} = \mathcal{H}_n$ denote the property of containing a Hamilton cycle (on n vertices). Then for every $\varepsilon \in (0,1)$ there exists $C = C(\varepsilon)$ such that the following is true: if $p \geqslant \frac{C\ln(n)}{n}$, then $\mathcal{H}$ is $\left(p, \frac{1}{2} - \varepsilon\right)$-resilient.*

Chapter 3

Fast Strategies in Waiter-Client Games

In this chapter we will provide proofs for every theorem related to fast strategies in Waiter-Client games. The results of this chapter are joint work with Dennis Clemens, Pranshu Gupta, Fabian Hamann, Alexander Haupt, and Mirjana Mikalački [18]. Whenever we use pictures to visualize certain game states, edges that Waiter offers are represented by dotted lines, Waiter's edges are coloured blue, and Client's edges are coloured red.

3.1 Unbiased perfect matching game

In this section we will prove Theorem 1.3.1 by showing a slightly stronger statement that will also be applied later in the discussion of the tree embedding game (see Section 3.4).

Theorem 3.1.1. *For large enough n, the following holds: Let $H \subset K_{n,n}$ be any subgraph with $e(H) \leqslant \frac{n}{2}$. Then, in the unbiased Waiter-Client game on $K_{n,n} \backslash H$, Waiter has a strategy to force a perfect matching of $K_{n,n}$ within $n+1$ rounds.*

Proof. Let $V = A \cup B$ be the bipartition of $K_{n,n}$. Throughout the game, we denote with R the set of isolated vertices in Client's graph. For $n-4$ rounds (Stage I), Waiter's strategy is to force a large matching in Client's graph greedily, while making sure that $e_{W \cup H}(R)$ decreases with every round as long as this value is still positive. Within five further rounds (Stage II), Waiter then completes this matching to a perfect matching.

If at any point during the game, Waiter is unable to follow her strategy, she forfeits the game (we will later see that this does not happen). The set R is dynamically updated after every turn. Waiter's strategy consists of the following two stages:

Stage I: This stage lasts $n-4$ rounds, in which Waiter forces a matching of size $n-4$ in Client's graph between A and B. Each round is played as follows: Let $u \in A \cap R$ be an arbitrary vertex maximizing $d_{W \cup H}(u, B \cap R)$. Then Waiter offers two free edges ub_1, ub_2 with $b_1, b_2 \in B \cap R$. By symmetry, assume that Client claims ub_1. Then, vertices u and b_1 are removed from R.

Stage II: When Waiter enters Stage II, Client's graph is a matching M' of size $n-4$. Let $S = A\backslash V(M')$ and $T = B\backslash V(M')$ at this point. Within five rounds, Waiter forces a matching of size 4 between S and T. The details are given later in the strategy discussion.

It is evident that, if Waiter can follow this strategy without forfeiting the game, she creates a perfect matching of $K_{n,n}\backslash H$ in Client's graph within $n+1$ rounds. It thus remains to show that she does not forfeit the game.

Strategy discussion:

Stage I: By induction on the number of rounds it follows that Waiter can follow the strategy of Stage I while she additionally maintains that

$$e_{W\cup H}(R) \leqslant \max\left\{0, \frac{|R|-n}{2}\right\} . \tag{3.1.1}$$

Indeed, the above inequality holds at the beginning of the game, since at that point $|R| = 2n$ and $e(H \cup W) \leqslant \frac{n}{2}$. Now consider any round r in Stage I, and assume that so far Waiter could follow her strategy and maintain Inequality 3.1.1. According to the strategy, she then picks a vertex $u \in A \cap R$ such that $d_{W\cup H}(u, B \cap R)$ is maximal. By induction we know that the number of vertices $b \in B \cap R$ with ub being free is at least

$$|B \cap R| - d_{W\cup H}(u, B \cap R) \geqslant \frac{|R|}{2} - \max\left\{0, \frac{|R|-n}{2}\right\} = \min\left\{\frac{|R|}{2}, \frac{n}{2}\right\} \geqslant 2\,.$$

Hence, there exist at least two vertices $b_1, b_2 \in B \cap R$ as required and Waiter can follow her strategy. Moreover, if $e_{W\cup H}(R) = 0$ was true at the beginning of round r, then the same still holds after the update of round r, since u gets removed from R. Otherwise, u was chosen with $d_{W\cup H}(u, B \cap R) \geqslant 1$ and $|R|$ decreases by 2 after the update, while $e_{W\cup H}(R)$ decreases by at least $d_{W\cup H}(u, B \cap R) \geqslant 1$. In any case, Inequality 3.1.1 holds again.

Stage II: When Waiter enters Stage II, Client's graph is a matching M' of size $n-4$. Moreover, using Inequality 3.1.1 from the end of Stage I, we deduce that $e_{W\cup H}(S,T) = 0$ needs to hold at that point. Waiter now forces a perfect matching between $S = \{s_1, s_2, s_3, s_4\}$ and $T = \{t_1, t_2, t_3, t_4\}$ as follows: In the first two rounds, she offers all of the edges s_1t_i with $i \in [4]$. Without loss of generality we may assume that Client chooses s_1t_1 and s_1t_2. Then, in the third round, Waiter offers s_2t_3 and s_2t_4, from which Client w.l.o.g. chooses s_2t_3. Afterwards, Waiter offers the edges s_3t_4 and s_4t_4, from which Client w.l.o.g. chooses s_3t_4. Finally, Waiter offers s_4t_1 and s_4t_2, and no matter which edge Client takes, a perfect matching is now finished. □

Proof of Theorem 1.3.1. If Waiter would want to win the unbiased perfect matching game on K_n within $\frac{n}{2}$ rounds, she would need a matching M of size $\frac{n}{2}-1$ after $\frac{n}{2}-1$ rounds. However, since there is only one possible edge to extend M to a perfect matching, but Waiter needs to offer two edges, Client can easily prevent the perfect matching in round $\frac{n}{2}$. Hence, $\tau_{WC}(\mathcal{PM}_n, 1) \geqslant \frac{n}{2}+1$ follows. For equality we just apply Theorem 3.1.1 with $H = \varnothing$ and let Waiter solely play on a subgraph of K_n isomorphic to $K_{\frac{n}{2},\frac{n}{2}}$. □

3.2 Unbiased Hamiltonicity game

In this section we will prove Theorem 1.3.5. We will actually prove a slightly stronger statement, which will give us some nice properties, which can be applied later in the discussions of the pancyclicity game (see Section 3.3) and the tree embedding game (see Section 3.4). If Waiter would just care about creating a Hamilton cycle, the proof could be simplified, but since we are not sacrificing speed by having a slightly more involved strategy, we will stick with the more complicated version. The statement we will prove reads as follows:

Theorem 3.2.1. *For large enough n, in the unbiased Waiter-Client game on K_n, Waiter has a strategy to force a Hamilton cycle H within $n+1$ rounds such that the following properties hold immediately after H is created:*

(1) $\forall v \in V(K_n) : d_W(v) < 10$.

(2) Let e_1^C be the first edge Client claims in the game, then $e_1^C \in E(H)$.

(3) There exists a path $P \subset C$ of length $\frac{1}{5}n$ s.t. $E_W(V(P)) = \varnothing$.

Proof. At the beginning of the game Waiter chooses an arbitrary subset $A_1 \subset V$ of size 4. Now her strategy is as follows: At first she forces four vertex disjoint paths P_i (with $i \in [4]$) in Client's graph, each having an endpoint in A_1, such that these paths cover the whole vertex set $V = V(K_n)$. Afterwards, she forces Client to connect the aforementioned paths to a Hamilton cycle such that the desired properties are satisfied.

Let $A_1 = \{a_i : i \in [4]\}$ and initially, for every $i \in [4]$, let P_i be the path consisting only of the vertex a_i. Waiter forces Client to extend the path P_i for every $i \in [4]$, such that a_i remains one of its endpoints, until the union of these four paths covers V. At any point of the game, we let $\mathcal{P}$ denote the collection of these four paths. We set $V(\mathcal{P}) = \bigcup_{i\in[4]} V(P_i)$ and $R = V \backslash V(\mathcal{P})$. Moreover, always let a_i' denote the endpoint of P_i which is not a_i (except when $v(P_i) = 1$ where we set $a_i' = a_i$), and let $A_2 = \{a_i' : i \in [4]\}$. For most of the game, Waiter's strategy is to consider the paths in pairs. She takes two turns to extend either P_1 and P_2 or P_3 and P_4 and does so alternately. In order to keep our notation short, we define π to be the permutation on $[4]$ with cycles $(1\ 2)$ and $(3\ 4)$, and we sometimes denote P_4 with P_0 when we consider indices modulo 4.

In the following, we present a strategy for Waiter and then prove that this strategy allows her to force a Hamilton cycle within $n+1$ rounds such that all of the desired properties are ensured. If at any point during the game, she is unable to follow her strategy, she forfeits the game (we will later see that this does not happen). The sets A_1, A_2, C, W, R, and $\mathcal{P}$ are updated at the end of every turn. Waiter's strategy consists of the following three stages:

Stage I: This stage lasts exactly $n-4$ rounds. During this stage Waiter extends the four paths P_i until $R = \varnothing$. She does this by alternating between two types of moves:

Type A: Let this be the i^{th} round, and $x \in R$ be a vertex maximizing $d_W(x, V(\mathcal{P}))$ (breaking ties arbitrarily). Then Waiter offers the edges xa_i' and xa_{i+1}' (with indices taken modulo 4). After Client has chosen one of these edges and thus extended one of the paths P_i or P_{i+1} (indices taken modulo 4), the sets A_2, C, W, R, and $\mathcal{P}$ are updated in the obvious way.

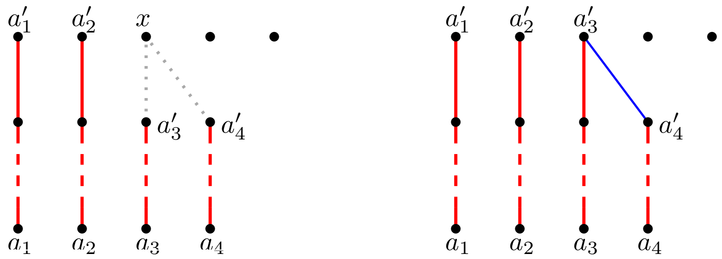

Figure 3.2.1: Waiter plays one move of Type A

Type B: Let this be the i^{th} round, and let $x, y \in R$ be picked arbitrarily. Moreover, let P_t be the path which was extended in the previous round. Then Waiter offers the edges $xa'_{\pi(t)}$ and $ya'_{\pi(t)}$. After Client has chosen one of these edges and thus extended the path $P_{\pi(t)}$, the sets A_2, C, W, R, and $\mathcal{P}$ are updated in the obvious way.

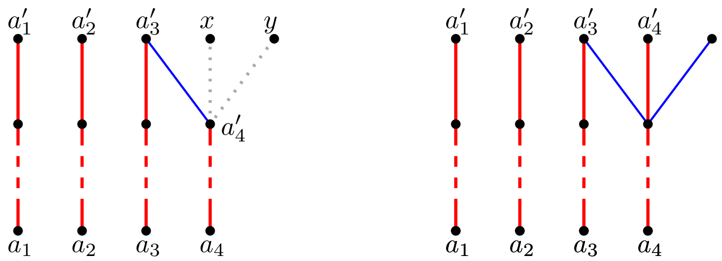

Figure 3.2.2: Waiter plays one move of Type B

As long as $|R| \geqslant 2$ holds, Waiter alternates between these two types of moves, with Type A being considered in odd rounds and Type B being considered in even rounds. Once $|R| = 1$ holds, Waiter plays one more round according to Type A. Afterwards, she proceeds with Stage II.

Stage II: This stage lasts exactly two rounds, in which Waiter forces Client to connect the paths from $\mathcal{P}$. As long as $|\mathcal{P}| > 2$, Waiter connects two paths in Client's graph as follows: She fixes a vertex $v \in A_2$ such that $d_W(v, A_2)$ is maximal and offers vx, vy where $x, y \in A_1$ are picked such that they do not belong to the same path as v. Without loss of generality assume that Client claims vx and thus connects two paths $P_{j_1}, P_{j_2} \in \mathcal{P}$. Then update A_1 and A_2 by removing v and x respectively, and update $\mathcal{P}$ by removing P_{j_1} and P_{j_2}, while adding the path induced by $E(P_{j_1}) \cup E(P_{j_2}) \cup \{vx\}$.

Once $|\mathcal{P}| = 2$ holds, Waiter proceeds with Stage III.

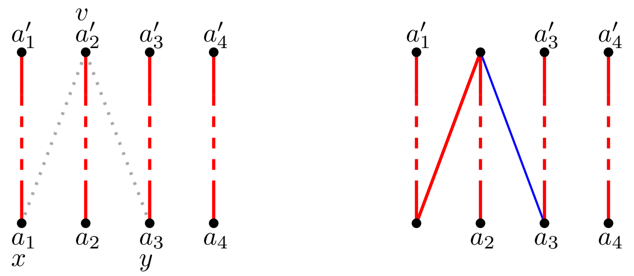

Figure 3.2.3: Waiter plays one round in Stage II

Stage III: Within three rounds Waiter forces a Hamilton cycle as desired. The details of how she can do this can be found in the strategy discussion.

It is evident that, if Waiter can follow the strategy without forfeiting the game, she creates a Hamilton cycle H within $n+1$ rounds. It thus remains to check that she does not forfeit the game and that H fulfils the properties (1)–(3) from Theorem 3.2.1.

Strategy discussion:

Stage I: At any point of the game we call a vertex $v \in R$ *bad* if $d_W(v, V(\mathcal{P})) \geqslant 1$ holds. We observe first that there will never be more than one such vertex which at the same time helps Waiter to follow the strategy of Stage I.

Observation 3.2.2. *For every $i \leqslant n-4$, Waiter can follow the i^{th} move of the proposed strategy. Moreover, the following holds for every $i \leqslant n-5$:*

(a) if i is odd, then after this round $e_W(A_2) = 1$ and no bad vertex exists. Moreover, the unique edge in $E_W(A_2)$ connects endpoints of P_i and P_{i+1} (indices taken modulo 4).

(b) if i is even, then after this round $e_W(A_2) = 0$ and there is exactly one bad vertex z. Moreover it holds that $d_W(z, V(\mathcal{P})) = d_W(z, V(P_{i-1}) \cup V(P_i)) = 1$ (indices taken modulo 4).

Moreover, (c) $e_W(A_2) \leqslant 2$ holds at the end of round $i = n-4$.

Proof. The statement follows by induction on i. Waiter can obviously follow the strategy for round 1, where she offers two edges according to Type A. The edge claimed by Client extends P_1 or P_2. After the update, the other edge belongs to $E_W(A_2)$ and connects the endpoints of P_1 and P_2, making sure that statement (a) holds for $i = 1$. Let $i > 1$ then.

Consider first the case when $i \leqslant n-5$ is even, and observe that $|R| \geqslant 2$ before round i. In round $i-1$ Waiter played according to Type A and extended a path P_t with $t \equiv i-1$ or i (mod 4). By induction, there was no bad vertex at the end of round $i-1$, but there was exactly one edge e^W in $E_W(A_2)$, and e^W connected endpoints of P_{i-1} and P_i (indices taken modulo 4). Now, in round i, Waiter wants to play according to Type B and needs to offer two edges $xa'_{\pi(t)}$ and $ya'_{\pi(t)}$ with $x, y \in R$. She can do this since $|R| \geqslant 2$ and x, y cannot be bad. The edge claimed by Client extends $P_{\pi(t)}$. By this, e^W is removed from $E_W(A_2)$ after the update, leading to $e_W(A_2) = 0$. The other edge goes to Waiter's graph and creates exactly one bad vertex $z \in \{x, y\}$. Since i is even and thus $\pi(t) \equiv i$ or $i-1 \pmod 4$, it follows that $d_W(z, V(\mathcal{P})) = d_W(z, V(P_{\pi(t)})) = d_W(z, V(P_{i-1}) \cup V(P_i)) = 1$ (indices taken modulo 4).

Consider next the case when $i \leqslant n-5$ is odd. By induction we know that $e_W(A_2) = 0$ and there was exactly one bad vertex z at the end of round $i-1$, but $d_W(z, V(\mathcal{P})) = d_W(z, V(P_{i-2}) \cup V(P_{i-1})) = 1$ (indices taken modulo 4). Now, in round i, Waiter wants to play according to Type A. She picks a vertex $x \in R$ maximizing $d_W(x, V(\mathcal{P}))$, hence setting $x = z$ by the uniqueness of the bad vertex. She needs to offer the edges xa'_i and xa'_{i+1}, which she can do since $d_W(x, V(P_i) \cup V(P_{i+1})) = 0$ (indices taken modulo 4). The edge claimed by Client extends a path in $\mathcal{P}$ by the vertex x, so that x is removed from R and there does not exist a bad vertex any more. After the update of $\mathcal{P}$ in round i, the edge which goes to Waiter's graph connects the endpoints of P_i and P_{i+1} (indices taken modulo 4) belonging to A_2, such that $e_W(A_2) = 1$ as claimed.

Finally, consider the case when $i = n-4$. Then, Waiter can follow the strategy for round i analogously to the case when $i \leqslant n-5$ is odd. By induction, using (a) or (b), it holds that $e_W(A_2 \cup R) = 1$ at the end of round $i-1$. Since, during round i, the last vertex of R is moved to A_2 and since Waiter receives only one new edge, it is immediately clear that $e_W(A_2) \leqslant 2$ afterwards. □

Stage II: When Waiter enters Stage II, Client's graph is the disjoint union of four vertex disjoint paths covering V, with each path being of length roughly $\frac{n}{4}$, since the pairs (P_1, P_2) and (P_3, P_4) were extended alternately during Stage I. Before we show that Waiter can follow Stage II of the proposed strategy, let us first observe how Waiter's edges are distributed at the end of Stage I.

Observation 3.2.3. *Right at the moment when Waiter enters Stage II, the following holds:*

(a) $E_W(A_1) \cup E_W(A_1, A_2) = \varnothing$ *and* $e_W(A_2) \leqslant 2$,

(b) $d_W(v) \leqslant 4$ *for every* $v \in V$,

(c) $E_W(V(P_i)) = \varnothing$ *for every* $i \in [4]$.

Proof. For (a) notice that only in the first four rounds of Stage I Waiter offers edges incident to A_1, and none of these edges is contained in A_1. All other endpoints of these edges are part of $V(\mathcal{P})$ at the end of the fifth round, since by property (a) of Observation 3.2.2, there do not exist bad vertices at that moment. But now, since all paths are extended further in Stage I by attaching edges to the vertices in A_2 and making appropriate updates, none of the mentioned endpoints belongs to A_2 later on. It thus follows that $E_W(A_1, A_1 \cup A_2) = \varnothing$ at the end of Stage I. The inequality $e_W(A_2) \leqslant 2$ is already given by property (c) in Observation 3.2.2.

For (b) observe that in Stage I, immediately after a vertex v is added to some path $P_i \in \mathcal{P}$, it holds that $d_W(v, V(\mathcal{P})) \leqslant 1$ if v was not bad before, or $d_W(v, V(\mathcal{P})) \leqslant 2$ if v was bad before. This degree may increase further by at most 2, when the pair of paths $(P_i, P_{\pi(i)})$ is considered for a further extension by a sequence of turns of Type A and Type B. But then, according to the strategy, both paths are extended, which ensures that from now on v is not an endpoint any more and Waiter does not offer any further edges at v throughout Stage I.

For (c), let e^W be any edge claimed by Waiter in Stage I. If this edge was offered by Type A, then after the corresponding round i, the edge e^W belongs to $E_W(V(P_i), V(P_{i+1}))$ (indices taken modulo 4). Otherwise, if e^W was offered by Type B, then after the corresponding round i, e^W connects the unique bad vertex z with the endpoint of one of the paths P_{i-1} or P_i. In the next round, playing according to Type A, Waiter makes sure that z is added to one of the paths P_{i+1} or P_{i+2}, leading to $e^W \in E_W(V(P_r), V(P_s))$ with $r \neq s$. □

Now, having Observation 3.2.3 in hand, one can easily see that Waiter can follow Stage II of her strategy without forfeiting the game. Indeed, by property (a) from the observation, we know that all edges between A_1 and $A_1 \cup A_2$ are free. Hence, she can offer edges vx and vy as required by her strategy. Moreover, we have $e_W(A_2) \leqslant 2$ at the beginning of Stage II. Since in Stage II Waiter always picks $v \in A_2$ such that $d_W(v, A_2)$ is maximized and since v is removed from A_2 after the update, it follows that $e_W(A_1 \cup A_2) = 0$ must hold at the end of Stage II.

Stage III: When Waiter enters Stage III, $\mathcal{P}$ consists of two paths, say P_1 and P_2, such that all four edges between their endpoints are still free. Moreover, it holds that $d_W(v) \leqslant 6$ for every $v \in V$, since these degrees were bounded by 4 at the end of Stage I and since Stage II took only two rounds.

Now, in Stage III, the first step for Waiter is to force a Hamilton path in Client's graph. To do so, she arbitrarily chooses an endpoint v of P_1 and offers the edges vx, vy with x, y being the endpoints of P_2. Let $P = (v_1, v_2, \dots, v_n)$ be the Hamilton path that is created in Client's graph by this first move. Then by the conditions from the beginning of Stage III we know that v_1v_n is still unclaimed.

Now, by using Pósa rotations [60], Waiter forces a Hamilton cycle in Client's graph. For her second move in Stage III Waiter picks two vertices v_i and v_j with $i, j \notin \{1, n-1, n\}$ such that they are not neighbours of each other, and such that $e_1^C \notin \{v_iv_{i+1}, v_jv_{j+1}\}$, where e_1^C denotes the edge claimed by Client in round 1, and such that the edges $v_1v_{i+1}, v_1v_{j+1}, v_iv_n, v_jv_n$ are still free. Such vertices must exist since Waiter's degree of all vertices is bounded by 7 at this moment. She offers v_iv_n and v_jv_n to Client, who w.l.o.g. claims v_iv_n. In the last round Waiter offers v_1v_{i+1} and v_1v_n, and no matter which edge Client chooses, that edge closes a Hamilton cycle H.

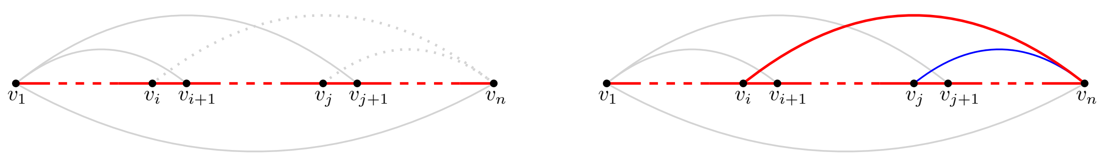

Figure 3.2.4: Waiter's second move in Stage III

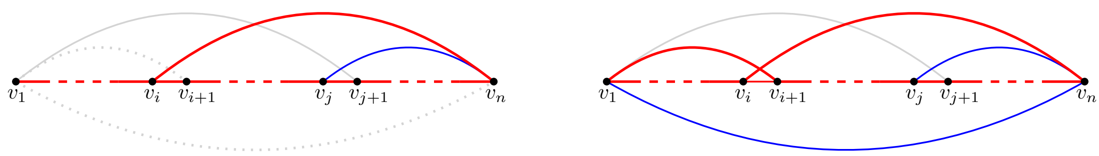

Figure 3.2.5: Waiter's third move in Stage III

Hence, in order to finish our argument, it remains to prove that the properties of Theorem 3.2.1 hold.

Property (1) holds since for every $v \in V$ we had $d_W(v) \leqslant 6$ at the beginning of Stage III, while Stage III lasted exactly three rounds. Property (2) holds because Client's only edge, which is not part of H, is either $v_i v_{i+1}$ or $v_i v_n$, depending on which edge Client claimed in the last round, and v_i was chosen in such a way that both edges differ from e_1^C.

For Property (3) recall that during Stage I and according to Observation 3.2.3 no interior edges of the paths $P_1, \ldots, P_4$ were claimed by Waiter, and each of these paths reached a length longer than $\frac{1}{5}n$. Also, when Waiter connects these paths during Stage II and Stage III, no such interior edges are created, since Waiter only offers edges between endpoints of paths, except for the very last two turns, where Waiter can claim at most two interior edges of the paths from Stage I. Moreover, when we remove the unique Client's edge which does not belong to H, at most one further path from Stage I can be destroyed, and hence there must remain at least one path supporting Property (3). □

Proof of Theorem 1.3.5. If Waiter wants to win the unbiased Hamiltonicity game on K_n within n rounds, she needs a Hamilton path of length $n-1$ after $n-1$ rounds. However, since there is only one possible edge to extend this to a Hamilton cycle and Waiter needs to offer two edges, Client can easily prevent the Hamilton cycle in round n. Hence, $\tau_{WC}(\mathcal{H}_n, 1) \geqslant n+1$ follows. For equality we apply Theorem 3.2.1. □

3.3 Unbiased pancyclicity game

Proof of Theorem 1.3.12. As preparation for the proof, set

$$g(n) = \lceil \log_2^{(k)}(n) \rceil = \lceil \underbrace{\log_2(\log_2(\ldots(\log_2(n))))}_{k \text{ times}} \rceil \quad \text{and} \quad f(n) = g(n) + 100$$

for any positive integer k. In the following we will describe a strategy for Waiter in the unbiased Waiter-Client game on K_n, and afterwards we will show that it is a strategy with which Waiter forces a pancyclic spanning subgraph of K_n within at most $n + \log_2(n) + O(\max\{f(n), k\})$ rounds. Whenever Waiter is not able to follow the proposed strategy, she forfeits the game (we will later see that this does not happen). The strategy is split into the following five stages:

Stage I: Within at most $n+1$ rounds, Waiter forces a Hamilton cycle $H = (v_1, v_2, \ldots, v_n)$ such that the following holds right at the moment when the Hamilton cycle is completed:

$(H1)$ $d_W(v) < 10$ for every $v \in V(K_n)$,

$(H2)$ for every $1 \leqslant j_1, j_2 \leqslant \frac{n}{10}$ with $|j_1 - j_2| \geqslant 2$ it holds that $v_{j_1}v_{j_2} \notin W \cup C$.

Further details can be found in the strategy discussion. Afterwards, Waiter proceeds with Stage II.

Stage II: This stage lasts two rounds. In the first round, Waiter offers the edges $v_1v_{f(n)+1}$ and $v_2v_{f(n)+2}$, among which Client needs to claim one; denote it with $w_1w_{f(n)+1}$, and afterwards let

$$w_i = \begin{cases} v_i & \text{if } w_1 = v_1 \\ v_{i+1} & \text{if } w_1 = v_2 \end{cases}$$

for every $i \in [n]$ (with $v_{n+1} := v_1$). In the second round, Waiter offers two free edges between $w_{f(n)+1}$ and $\{w_{n-60}, \ldots, w_{n-50}\}$, among which Client needs to choose one.

Afterwards, Waiter proceeds with Stage III.

Stage III: This stage lasts $f(n) - 2$ rounds. In the i^{th} round of Stage III, Waiter offers the edges w_1w_{i+2} and $w_{f(n)-i}w_{f(n)+1}$, among which Client always needs to claim one.

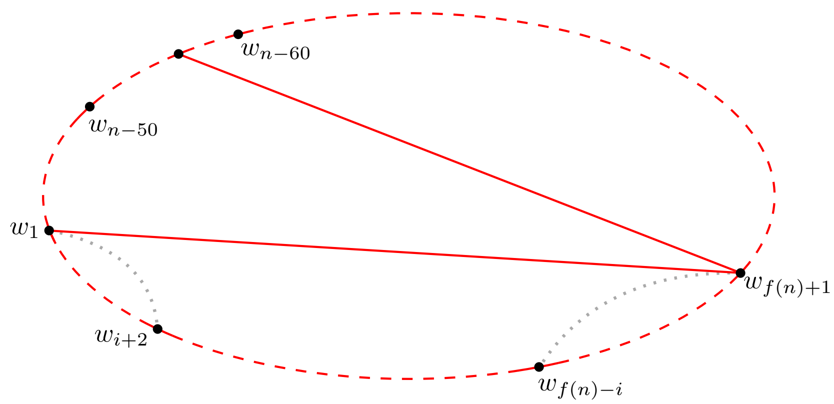

Figure 3.3.1: Waiter's i^{th} move in Stage III

Once all of the $f(n) - 2$ rounds of Stage III are played, Waiter proceeds with Stage IV.

Stage IV: This stage lasts at most $\lceil \log_2(n) \rceil$ rounds. Waiter makes sure that at the end of the i^{th} round of Stage IV, there exist vertices $w_{t_0}, w_{t_1}, \ldots, w_{t_i}$ such that the following holds:

$(W1)$ $f(n) + 1 = t_0 < t_1 < \ldots < t_i \leqslant n$,

$(W2)$ $w_{t_{j-1}}w_{t_j} \in C$ for every $j \in [i]$,

$(W3)$ $\min\{2t_{i-1} - 2i, n\} - 20 \leqslant t_i \leqslant \min\{2t_{i-1} - 2i, n\}$.

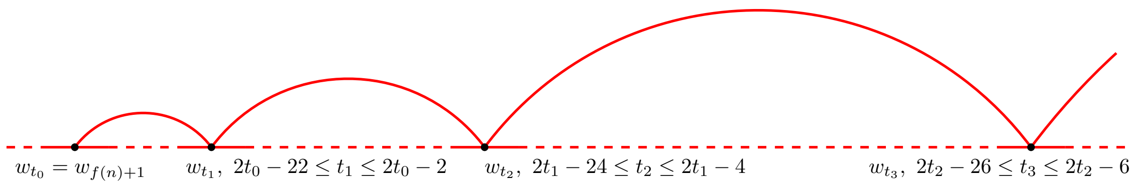

Figure 3.3.2: Edges which Waiter claims during Stage IV

In order to do so, first set $t_0 = f(n) + 1$, and then in the i^{th} round Waiter offers two free edges of the form $w_{t_{i-1}} w_j$ with $\min\{2t_{i-1} - 2i, n\} - 20 \leqslant j \leqslant \min\{2t_{i-1} - 2i, n\}$. For the edge $w_{t_{i-1}} w_j$ chosen by Client, Waiter then sets $t_i := j$.

Once there is a round s such that $n - 20 \leqslant t_s \leqslant n$ holds, Waiter stops with Stage IV and proceeds with Stage V.

Stage V: This stage lasts at most $k - 1$ rounds. For her i^{th} move, Waiter aims to make Client claim an edge $w_1 w_\ell$ with $2 \log_2^{(i)}(n) \leqslant t_j \leqslant \ell \leqslant t_j + 20 \leqslant 10 \log_2^{(i)}(n)$ for some $j \leqslant s$. In case Client does not already possess such an edge, Waiter just offers two free edges which fulfil the aforementioned property. Otherwise, Waiter just skips that move and proceeds to her next move.

In the following discussion, we need to check two properties for the given strategy: (1) Waiter can always follow the proposed strategy without forfeiting the game, and (2) when Stage V is over, Client's graph is pancyclic. Just note that when both properties are fulfilled a pancyclic graph can be forced within at most

$$(n+1) + 2 + (f(n) - 2) + \lceil \log_2(n) \rceil + (k-1) = n + (1 + o(1)) \log_2(n)$$

rounds, according to the bounds on the number of rounds given in the descriptions of the stages.

Strategy discussion:

(1) – Following the strategy: Waiter can follow Stage I because of Theorem 3.2.1. According to that theorem, Waiter can force a Hamilton cycle H within $n + 1$ rounds such that Property $(H1)$ holds immediately after H is created. Moreover, she can make sure that right at this moment there is a path $P \subset H$ of length $\frac{n}{5}$ such that $E_W(V(P)) = \varnothing$ holds. We can then split P into two subpaths Q_1 and Q_2 of length $\frac{n}{10}$ each. Since $e(C \backslash H) = 1$ holds at the end of round $n + 1$, we know that there must be some $i \in [2]$ with $E_C(V(Q_i)) \backslash E(Q_i) = \varnothing$. Labelling the vertices of H in such a way that $V(Q_i) = \{v_1, \ldots, v_{\frac{n}{10}}\}$ holds, we obtain Property $(H2)$.

Afterwards, in Stage II and in Stage III, Waiter needs to offer several edges contained in $E(\{v_i : i \leqslant f(n) + 2\}) \backslash E(H)$, which are still free by Property $(H2)$ and since $f(n) + 2 < \frac{n}{5}$. She also needs to offer two edges between $w_{f(n)+1}$ and $\{w_{n-60}, \ldots, w_{n-50}\}$ which is possible since $d_W(w_{f(n)+1}) < 10$ at the end of Stage I.

Next consider Stage IV and observe the following: if Waiter can follow this part of her strategy and as long as $t_i < n - 20$ holds, we have $t_i \geqslant 2t_{i-1} - 2i - 20$ and $t_0 \geqslant f(n) > 100$, leading to

$$t_i > 2^i + i^2 + 50 \tag{3.3.1}$$

by a simple induction. Thus, Stage IV lasts at most $\lceil \log_2(n) \rceil$ rounds. To show that Waiter can follow each of these rounds, we proceed by induction on i: Let us consider the i^{th} round of Stage IV (when $w_{t_0}, \dots, w_{t_{i-1}}$ are already given, and $(W1)$–$(W3)$ hold for $i-1$). At the end of Stage I we have $d_W(w_{t_{i-1}}) < 10$ and, since Stage II lasted two rounds, we have $d_W(w_{t_{i-1}}) < 12$ at the end of Stage II. Since afterwards (in Stage III–IV) each offered edge until the current round was incident to some w_ℓ, $\ell < t_{i-1} < \min\{2t_{i-1} - 2i, n\} - 20$, there are at least two free edges available as required by the strategy description. Once Client has claimed one of these edges, it is obvious that the Properties $(W1)$–$(W3)$ hold again for i.

Finally, consider the i^{th} round of Stage V for $i \in [k-1]$. Since $t_0 = f(n)+1$, $t_s \geqslant n-20$, and since $t_{j+1} < 2t_j$ for all $j \leqslant s$, it follows that there must be some $j \in [s]$ with $2\log_2^{(i)}(n) \leqslant t_j \leqslant 5\log_2^{(i)}(n)$. Having such a t_j fixed, it is enough to find two free edges $w_1 w_\ell$ with $t_j \leqslant \ell \leqslant t_j + 20$. This is possible, because $d_W(w_1) < 10$ at the end of Stage I and since in Stage II-IV no such edge was offered.

(2) – Finding pancyclicity: Let $H = (w_1, w_2, \dots, w_n)$ be the Hamilton cycle from Stage I. It is the edge disjoint union of two paths $P_1 = (w_1, w_2 \dots, w_{f(n)+1})$ and $P_2 = (w_{f(n)+1}, \dots, w_{n-1}, w_n, w_1)$ of lengths $f(n)$ and $n - f(n)$, respectively. Both paths are closed to cycles in Client's graph by the edge $w_1 w_{f(n)+1}$ which was claimed in Stage II. We next observe that after Stage III the following holds:

Observation 3.3.1. *For every $0 \leqslant t \leqslant f(n) - 1$ there is a path $P_1^t \subset C$ such that*

(i) $V(P_1^t) \subset V(P_1)$,

(ii) P_1^t *has length* $f(n) - t$,

(iii) w_1 *and* $w_{f(n)+1}$ *are the endpoints of* P_1^t.

Proof. For $t = 0$ and $t = f(n) - 1$ we let $P_1^{f(n)-1}$ consist of the edge $w_1 w_{f(n)+1}$ and $P_1^0 = P_1$. For every $1 \leqslant t \leqslant f(n) - 2$ Client claimed either $w_1 w_{t+2}$ or $w_{f(n)-t} w_{f(n)+1}$ in round t of Stage III, and thus we can choose either $P_1^t = (w_1, w_{t+2}, w_{t+3}, \dots, w_{f(n)}, w_{f(n)+1})$ or $P_1^t = (w_1, w_2, \dots, w_{f(n)-t-1}, w_{f(n)-t}, w_{f(n)+1})$. □

Let $w_{f(n)+1} w_{n-p}$ be the edge claimed by Client in the second round of Stage II, and observe that $50 \leqslant p < f(n)$. By closing the above paths P_1^t into cycles, either using the edge $w_1 w_{f(n)+1}$ or the path $(w_{f(n)+1}, w_{n-p}, w_{n-p+1}, \dots, w_n, w_1)$, we obtain cycles of all lengths between $3 \leqslant \ell \leqslant f(n) + p$. Hence, it remains to find cycles of all the lengths larger than $f(n) + p \geqslant f(n) + 50$. In order to do so, we will fix $0 \leqslant m \leqslant k-1$ from now on and we will explain how we find cycles of all lengths between $\log_2^{(m+1)}(n) + 50$ and $\min\{2\log_2^{(m)}(n), n\}$ in Client's graph. Running over all possible m finishes the argument, as the interval $[3, \log_2^{(k)}(n) + 50]$ and all the intervals $[\log_2^{(m+1)}(n) + 50, \min\{2\log_2^{(m)}(n), n\}]$ with $0 \leqslant m \leqslant k-1$ cover all integers from 3 to n.

Having m fixed, set $w_{k_m} = w_n$ if $m = 0$ and otherwise let w_{k_m} be the vertex w_ℓ from the m^{th} move in Stage V. By Stage IV (in case $m = 0$) or Stage V (in case $m \neq 0$) there is some index

$j_m \leqslant s$ such that $k_m - 20 \leqslant t_{j_m} \leqslant k_m$. Moreover, set $a_i := t_i - t_{i-1} - 1$ to be the number of vertices between $w_{t_{i-1}}$ and w_{t_i} on P_2, for every $i \leqslant s$. Then, at the end of Stage V the following holds:

Observation 3.3.2. *For every subset $S \subset [j_m]$ there is a path $P_2^S \subset C$ such that*

(i) $V\big(P_2^S\big) \subset V(P_2)$,

(ii) P_2^S *has length* $k_m - f(n) - \sum_{i\in S} a_i$,

(iii) w_1 *and* $w_{f(n)+1}$ *are the endpoints of* P_2^S.

Proof. If we extend the subpath $(w_{f(n)+1}, \dots, w_{k_m})$ from P_2 by Client's edge $w_{k_m}w_1$, we obtain a path P_2' of length $k_m - f(n)$. By replacing any subpath $(w_{t_{i-1}}, \dots, w_{t_i})$, where $i \leqslant j_m$, with the edge $w_{t_{i-1}}w_{t_i}$ which was claimed in Stage IV, the path P_s' can be shortened by exactly a_i edges. As this can be done for any $i \in S$, we can shorten P_2' to a path of length $k_m - f(n) - \sum_{i\in S} a_i$. This proves the observation. □

Now, by joining the path P_1^t with the path P_2^S, for any $0 \leqslant t \leqslant f(n) - 1$ and any $S \subset [j_m]$, we obtain a cycle of length $k_m - (t + \sum_{i\in S} a_i)$. We will see in the following that this will indeed give us cycles of all lengths between $\log_2^{(m+1)}(n) + 50$ and $\min\{2\log_2^{(m)}(n), n\}$. We start with the following observation.

Observation 3.3.3. *Every integer in $[f(n) - 1 + \sum_{i\in[j_m]} a_i]$ can be written in the form $t + \sum_{i\in S} a_i$ with $0 \leqslant t \leqslant f(n) - 1$ and $S \subset [j_m]$.*

Proof. Inductively, we can show that for every $0 \leqslant j \leqslant j_m$

(S) every integer in $[f(n) - 1 + \sum_{i\in[j]} a_i]$ can be written as a sum $t + \sum_{i\in S} a_i$ with $0 \leqslant t \leqslant f(n) - 1$ and $S \subset [j]$.

The induction start ($j = 0$) is obvious. So, let $j + 1 > 0$, and assume (S) to be true for j. By $(W3)$, the definition of a_j and since $t_0 = f(n) + 1$, it follows that

$$a_{j+1} = t_{j+1} - t_j - 1 \leqslant t_j - 2(j+1) \leqslant \sum_{i\in[j]} a_i + f(n) - 1\,.$$

By induction, the integers up to the last sum can be written as $t + \sum_{i\in S} a_i$ with $0 \leqslant t \leqslant f(n) - 1$ and $S \subset [j]$, adding a_{j+1} to the latter creates all integers in

$$\left[a_{j+1}, \sum_{i\in[j+1]} a_i + f(n) - 1\right] \supset \left[\sum_{i\in[j]} a_i + f(n) - 1, \sum_{i\in[j+1]} a_i + f(n) - 1\right].$$

Note that the last set contains all the remaining integers to complete the induction step. This shows (S) for $j + 1$ and finishes the proof of the observation. □

Finally, observe that $f(n) - 1 + \sum_{i \in [j_m]} a_i = t_{j_m} - j_m - 2$ and hence, by Observation 3.3.3 and by the argument immediately after Observation 3.3.2, we see that we can find cycles of all lengths between $k_m - (t_{j_m} - j_m - 2)$ and k_m. Now, note that $k_m \geqslant t_{j_m} \geqslant k_m - 20$ by the choice of k_m, and $k_m \geqslant \min\{2\log_2^{(m)}(n), n\}$ since $t_{j_m} \geqslant 2\log_2^{(m)}(n)$ by Stage V (in case when $m \neq 0$). Moreover, using that $t_{j_m} \leqslant 10\log_2^{(m)}(n)$ and (3.3.1) hold, we get $j_m \leqslant \log_2^{(m+1)}(n) + 10$. Hence, we obtain cycles of all lengths between $\log_2^{(m+1)}(n) + 50 \geqslant k_m - (t_{j_m} - j_m - 2)$ and $\min\{2\log_2^{(m)}(n), n\} \leqslant k_m$ as desired. □

Remark: In the above argument, we need that the interval $[3, \log_2^{(k)}(n) + 50]$ and all the intervals $[\log_2^{(m+1)}(n) + 50, \min\{2\log_2^{(m)}(n), n\}]$ with $0 \leqslant m \leqslant k-1$ cover all integers from 3 to n. Hence we only need to ensure that $2\log_2^{(m)}(n) \geqslant \log_2^{(m)}(n) + 50$ holds for all $m \leqslant k-1$, i.e. $\log_2^{(m)}(n) \geqslant 50$, which is given when $\log_2^{(k+2)}(n) \geqslant 2$. Thus, if we choose $H(n)$ to be the smallest integer t such that $\log_2^{(t)}(n) < 2$ holds, then the above proof gives us that the game is won within $n + \log_2(n) + H(n) + O(1)$ rounds. This is only an additive constant away from the best known general upper bound on the minimal size of pancyclic graphs as mentioned in [13].

3.4 Unbiased games involving trees

In this section we will prove Theorem 1.3.15. Based on ideas from [17] and [32], we will split the given tree T into a subgraph T' and a nice behaving structure, which will be either a large matching or a long path. In her strategy, Waiter will first force a copy of T' more or less greedily and without wasting any move, while additionally caring about the distribution of her edges. Afterwards, Waiter will force the appropriate large matching or long path while wasting at most one round.

Let T be any tree. We denote by $L = L(T)$ the set of leaves of T and by $N_T(L)$ the set of vertices which are in the neighbourhood of the leaves with regards to T. For every $x \in N_T(L)$ let $\ell(x)$ be the number of leaves which are neighbours of x in T and define $\Delta(N_T(L)) := \max_{x \in N_T(L)} \ell(x)$.

We start with the following lemma, similar to Lemma 2.1 in [48], which states that each of the trees T considered in Theorem 1.3.15 contains a large matching, where every edge is incident to a leaf, or a long *bare path*, i.e. a path where all of the inner vertices have degree two in T.

Lemma 3.4.1. *For every $\mu \in \left(0, \frac{1}{2}\right)$ there exists $\varepsilon > 0$ such that the following holds for every large enough integer n: Let T be a tree on n vertices with $\Delta(N_T(L)) \leqslant \varepsilon\sqrt{n}$, then either $|N_T(L)| \geqslant \mu\sqrt{n}$ or T contains a bare path of length at least $\mu\sqrt{n}$.*

Proof. Set $\varepsilon = \frac{\mu}{3}$ and assume that $|N_T(L)| < \mu\sqrt{n}$. We will show that in this case T needs to contain a bare path of length at least $\mu\sqrt{n}$. By our assumption, we obtain

$$|L| \leqslant \Delta(N_T(L)) \cdot |N_T(L)| < \varepsilon\mu n\,.$$

Now, if we set $T' = T - L$, we get $n' := |V(T')| \geqslant n - \mu\varepsilon n > \frac{2n}{3}$.

Let $S_i := \{v \in V(T') \mid d_{T'}(v) = i\}$ and $S_{\geqslant i} := \{v \in V(T') \mid d_{T'}(v) \geqslant i\}$ for every $i \in [n]$, and observe that $S_1 \subseteq N_T(L)$. Further, let $\mathcal{P}$ be the collection of maximal bare paths in T' and let $\tilde{T}$ be the tree obtained from T' by contracting each path in $\mathcal{P}$ to an edge. Then

$$|\mathcal{P}| = e(\tilde{T}) = v(\tilde{T}) - 1 = |S_1| + |S_{\geqslant 3}| - 1\,.$$

Also, by the Handshake Lemma it holds that

$$\begin{aligned}2(n'-1) = 2e(T') &\geqslant |S_1| + 2|S_2| + 3|S_{\geqslant 3}| \\ &= 2(|S_1| + |S_2| + |S_{\geqslant 3}|) + |S_{\geqslant 3}| - |S_1| = 2n' + |S_{\geqslant 3}| - |S_1|\,,\end{aligned}$$

leading to $|S_{\geqslant 3}| < |S_1|$ and hence

$$|\mathcal{P}| < |S_1| + |S_{\geqslant 3}| < 2|S_1| \leqslant 2|N_T(L)| < 2\mu\sqrt{n} \leqslant \mu\sqrt{6n'}\,.$$

By the Pigeonhole Principle and since each vertex from S_2 belongs to exactly one path in $\mathcal{P}$, there has to exist a bare path of length at least

$$\frac{|S_2|}{|\mathcal{P}|} = \frac{n' - |S_1| - |S_{\geqslant 3}|}{|\mathcal{P}|} \geqslant \frac{n' - \mu\sqrt{6n'}}{\mu\sqrt{6n'}} > \mu\sqrt{n}\,,$$

where the last inequality uses that $\mu < \frac{1}{2}$, $n' > \frac{2}{3}n$, and that n is large enough. □

Theorem 1.3.15 will follow from the next slightly stronger result which will be used later as well for the study of the tree-factor game.

Theorem 3.4.2. *There exists $\varepsilon > 0$ such that the following holds for every large enough integer n: Let T be a tree on n vertices and let $v \in V(T) \backslash (L \cup N_T(L))$ be such that the following holds:*

1. $d_T(v) \leqslant \frac{n}{3}$, *and*
2. $\Delta(T \backslash \{v\}) \leqslant \varepsilon\sqrt{n}$.

Moreover, let $p \in V(K_n)$. Then, in an unbiased Waiter-Client game on K_n, Waiter has a strategy to force Client to claim a copy of T within n rounds such that

(a) in Client's copy of T, the vertex $p \in V(K_n)$ represents the vertex $v \in V(T)$, and

(b) in each round of her strategy, Waiter offered either 2 edges or no edge incident to p.

Proof. Let $\mu = \frac{1}{3}$ and choose $\varepsilon < \frac{\mu}{20}$ according to Lemma 3.4.1. Then, given a tree T with the properties of the theorem above, there exists a bare path of length at least $\mu\sqrt{n}$ in T (Case A) or we have $|N_T(L)| \geqslant \mu\sqrt{n}$ (Case B). We provide different strategies for Waiter for each case.

In order to describe Waiter's strategy, we use notation similar to that from [17] and [32]. Let $S \subseteq V(T)$ be an arbitrary set, then an *S-partial embedding* of T in G is an injective mapping $f : S \to V(G)$ such that $f(x)f(y)$ is an edge in G whenever xy is an edge in T. Vertices in S are called *embedded* vertices. Let any subgraph $T' \subset T$ be given. Then a vertex $v \in f(S)$ is called *closed* with respect to T' if all the neighbours of $f^{-1}(v)$ in T' are embedded as well. Otherwise v is called *open* w.r.t. T'. With $\mathcal{O}_{T'}$ we denote the set of all vertices that are open w.r.t. T'. Moreover, the vertices of the set $A := V(G)\backslash f(S)$ are called *available*.

For both cases Waiter's strategy consists of three stages. If at any point during the game, Waiter is unable to follow her strategy, she forfeits the game (we will later see that this does not happen).

Case A – Long bare path. First we consider the case when T contains a bare path of length at least $\mu\sqrt{n}$. Let P be such a path of length $\mu\sqrt{n}$, and denote its endpoints with u and w. Let u_1 and w_1 be the neighbours of u and w in P, respectively. Then $T\backslash P$ is a forest with two tree components, which we denote with T_1 and T_2. Let $T' \supset T$ be the forest induced by $E(T_1) \cup E(T_2) \cup \{uu_1, ww_1\}$. Without loss of generality we may assume that both v and u belong to T_1.

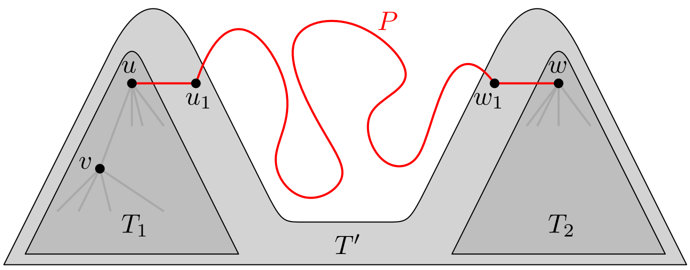

Figure 3.4.1: Setup for Case A

In broad terms, Waiter's strategy is to first force a copy of T' during Stage I and Stage II, and then force a copy of the bare path P in Stage III in such a way that a copy of T is created within n rounds. Throughout the game, she maintains a set S and an S-partial embedding f of T into K_n in order to represent the subgraph of T which currently is isomorphic to Client's graph. Initially, set $S = \{v, w\}$, $f(v) = p$, and $f(w) = q$ for arbitrary $p, q \in V(K_n)$. Waiter's strategy is split into the following stages:

Stage I: This stage lasts for $d_T(v)$ rounds in which Waiter closes the vertex v w.r.t. T. Each round is played as follows:

First Waiter fixes an arbitrary vertex $t \in N_T(v)\backslash S$. Waiter then offers two edges pa_1, pa_2 such that both edges are free and $a_1, a_2 \in A$. By symmetry, assume that Client chooses the edge pa_1. Then Waiter updates A, S, and f by removing a_1 from A, adding t to S, and setting $f(t) := a_1$.

Stage II: This stage lasts $e(T') - d_T(v)$ rounds in which Waiter's goal is to create a $V(T')$-partial embedding. Each round is played as follows:

If $S = V(T')$ holds, Waiter proceeds with Stage III. Otherwise, fix an arbitrary vertex $x \in f(S\backslash\{u_1, w_1\}) \cap \mathcal{O}_T$ and let $t = f^{-1}(x)$. Since x is open, there exists a vertex $z \in (V(T_1) \cup V(T_2) \cup \{u_1, w_1\})\backslash S$ such that $tz \in E(T')$. Waiter then offers two free edges xa_1 and xa_2 such that $a_1, a_2 \in A$. By symmetry, we may assume that Client chooses xa_1. Then Waiter updates A,

S, and f by removing a_1 from A, adding z to S, and setting $f(z) := a_1$. Afterwards, she repeats Stage II.

Stage III: When Waiter enters Stage III, Client's graph is a copy of the subgraph T'. Within $n - e(T')$ rounds, Waiter now forces a Hamilton path on $V(K_n) \backslash f(V(T_1) \cup V(T_2))$ with endpoints $f(u_1)$ and $f(w_1)$. The details of how Waiter can do this are given in the strategy discussion.

Strategy discussion:

It is obvious that if Waiter can follow the given strategy without forfeiting the game, she forces a copy of T within at most $d_T(v) + (e(T') - d_T(v)) + (n - e(T')) = n$ rounds. Hence, it remains to show that Waiter can indeed do so. However, before we study each stage separately, let us observe the following:

Observation 3.4.3. *Throughout Stages I and II, as long as Waiter can follow the proposed strategy, it holds that*

(i) $|A| \geqslant \mu\sqrt{n} - 2$ *and* $e_{C \cup W}(A) = 0$,

(ii) $d_W(x, A) \leqslant d_C(x)$ *for every* $x \in f(S)$.

Proof. Property (i) is immediately clear. The inequality $|A| \geqslant \mu\sqrt{n} - 2$ holds, since the strategy for the mentioned stages is to force a copy of T' without wasting any move and since $e(T') = e(T_1 \cup T_2) + |\{uu_1, ww_1\}| = n - \mu\sqrt{n} + 2$. The equation $e_{C \cup W}(A) = 0$ holds, since Waiter always offers only edges that intersect $f(S)$. For Property (ii), observe that $d_W(x, A)$ can only increase whenever $x \in f(S)$ (since $e_{C \cup W}(A) = 0$) and Waiter offers an edge between x and A. However, when this happens in any of the Stages I and II, Waiter actually offers two edges between x and A, which increases $d_C(x)$ by one at the same time. Hence, $d_W(x, A)$ cannot become larger than $d_C(x)$. □

In the following we show that Waiter can always follow her strategy without forfeiting the game.

Stage I: According to her strategy, Waiter needs to offer $2d_T(v) \leqslant \frac{2n}{3}$ edges at v. She can easily do so, since there exist $n - 1$ edges to choose from.

Stage II: The vertex z, which is described in the strategy, exists because of our assumption that x is an open vertex. Moreover, by Observation 3.4.3, we have $d_W(x, A) \leqslant d_C(x) \leqslant d_T(f^{-1}(x)) \leqslant \varepsilon\sqrt{n}$, which in turn means that at least $|A| - \varepsilon\sqrt{n} > \mu\sqrt{n} - 2 - \varepsilon\sqrt{n} > \frac{\mu}{2}\sqrt{n}$ edges between x and A are free. Hence there exist two free edges xa_1 and xa_2 as desired and thus Waiter can follow her strategy.

Stage III: When Waiter enters Stage III, she has successfully managed to force a copy of T'. Let $A' = V(K_n) \backslash f(V(T_1) \cup V(T_2))$ and observe that $e_{C \cup W}(A') = 0$ holds right at this moment. Indeed, according to Observation 3.4.3 we have $e_{C \cup W}(A) = 0$. Moreover, $e_{C \cup W}(\{f(u_1), f(w_1)\}, A) = 0$ holds, since in Stage II Waiter always chooses x different from

$f(u_1)$ and $f(w_1)$, which in turn ensures that, once these vertices are embedded, Waiter never offers edges incident to those vertices again.

During Stage III Waiter plays as follows. First she considers a fake round which is not played at all but where Waiter pretends that Client claimed the edge $e_C := f(u_1)f(w_1)$. Afterwards, she continues according to the strategy from Theorem 3.2.1 (with K_n being replaced by $K_n[A'] \cong K_{|A'|}$), which ensures that within $|A'|+1$ rounds she can force a Hamilton cycle in Client's graph on A' which contains the edge e_C. Since the first round was a fake round, this actually means that, within $|A'| = n - e(T')$ rounds, Waiter obtains a Hamilton path in A' as desired.

Case B – Many leaf neighbours. Next consider the case when $|N_T(L)| \geqslant \mu\sqrt{n}$ holds. Then, there exists a matching M_0 of size at least $\mu\sqrt{n}$ which consists of edges that are incident to leaves of T. Define the sets $L_0 := L \cap V(M_0)$, $D_i := \{w \in V(T) : \text{dist}_T(v,w) = i\}$, $D_{\text{odd}} := \bigcup_{i \text{ odd}} D_i$, and $D_{\text{even}} := \bigcup_{i\neq 0 \text{ even}} D_i$. By the Pigeonhole Principle there is a set $D_{\text{good}} \in \{D_{\text{odd}}, D_{\text{even}}\}$ such that $D_{\text{good}} \cap N_T(L_0)$ has size at least $\mu'\sqrt{n}$ with $\mu' := \frac{\mu}{3}$. Let $M' = \{e \in M_0 : e \cap D_{\text{good}} \neq \varnothing\}$, $L' = L \cap V(M')$, and $T' = T - L'$. By our choice of D_{good} we have that $|M'| \geqslant \mu'\sqrt{n}$, $v \notin V(M')$ and $\text{dist}_{T'}(x,y) \geqslant 2$ for every $x, y \in N_T(L')$.

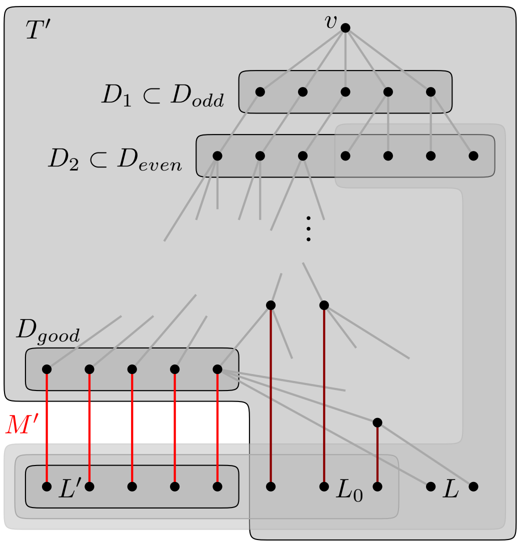

Figure 3.4.2: Setup for Case B

In broad terms, Waiter's strategy is to first force a copy of T' (Stage I and II) and then force a copy of the matching M' (Stage III) in such a way that a copy of T is created within n rounds. Throughout the game, she again maintains a set S and an S-partial embedding f of T into K_n in order to represent the subgraph of T which is currently isomorphic to Client's graph. Initially, set $S = \{v\}$ and $f(v) = p$ for an arbitrary $p \in V(K_n)$. Additionally, at any moment in the game let $S' := S \cap N_T(L')$ be the set of neighbours of leaves in L' which are already embedded.

Stage I: This stage lasts for $d_{T'}(v) = d_T(v)$ rounds in which Waiter closes the vertex v with regards to T'. Each round is played as follows:

First Waiter fixes an arbitrary vertex $t \in N_{T'}(v) \backslash S$. Then Waiter offers two edges pa_1, pa_2 such that both edges are free and $a_1, a_2 \in A$. By symmetry, assume that Client chooses the edge pa_1. Waiter then updates A, S, and f by removing a_1 from A, adding t to S, and setting $f(t) := a_1$.

Stage II: This stage lasts $e(T') - d_{T'}(v)$ rounds in which Waiter's goal is to create a $V(T')$-partial embedding, while also taking care of the distribution of her edges between the open and the available vertices. In each round she plays as follows:

If $S = V(T')$ Waiter proceeds to Stage III. Otherwise, Waiter considers the following case distinction:

Case 1. Let there be two vertices $u_1, u_2 \in \mathcal{O}_{T'}$, and let $t_1 = f^{-1}(u_1)$ and $t_2 = f^{-1}(u_2)$. By assumption, there exist vertices $z_1, z_2 \in V(T') \backslash S$ such that $t_1 z_1, t_2 z_2 \in E(T')$. Waiter then picks any vertex $a \in A$ such that au_1 and au_2 are free, where she prefers to pick vertices which satisfy $d_W(a, f(S')) \geqslant 1$, and offers au_1 and au_2 to Client. By symmetry we may assume that Client chooses au_1. Waiter then updates A, S, and f by removing a from A, adding z_1 to S, and setting $f(z_1) := a$.

Case 2. Let there be only one vertex $u \in \mathcal{O}_{T'}$, and further let $u \notin f(S')$. Let $t = f^{-1}(u)$. By assumption, there exists a vertex $z \in V(T') \backslash S$ such that $tz \in E(T')$. Waiter then picks any two vertices $a_1, a_2 \in A$ such that $a_1 u$ and $a_2 u$ are free, where she prefers to pick vertices which satisfy $d_W(a_i, f(S')) \geqslant 1$ for $i \in [2]$, and offers these two edges to Client. By symmetry we may assume that Client chooses $a_1 u$. Waiter then updates A, S, and f by removing a_1 from A, adding z to S, and setting $f(z) := a_1$.

Case 3. Let there be only one vertex $u \in \mathcal{O}_{T'}$, and moreover we have that $u \in f(S')$. Let $t = f^{-1}(u)$. By assumption, there exists at least one vertex $z \in V(T') \backslash S$ such that $tz \in E(T')$, and among such vertices Waiter chooses z such that $d_{T'}(z)$ is maximal. Waiter then picks any two vertices $a_3, a_4 \in A$ such that $a_3 u$ and $a_4 u$ are free and such that $d_W(a_3, f(S')) = d_W(a_4, f(S')) = 0$, and offers these two edges to Client. By symmetry we may assume that Client chooses $a_3 u$. Waiter then updates A, S, and f by removing a_3 from A, adding z to S, and setting $f(z) := a_3$.

Afterwards, Waiter repeats Stage II.

Stage III: When Waiter enters Stage III, Client's graph is a copy of the subgraph T'. Within $e(M') + 1$ rounds, Waiter now forces a perfect matching between $V(K_n) \backslash f(V(T'))$ and $f(N_T(L'))$. The details of how Waiter can do this are given in the strategy discussion.

Strategy discussion:

It is obvious, that if Waiter can follow the given strategy without forfeiting the game, she forces a copy of T within at most $d_T(v) + e(T') - d_T(v) + e(M') + 1 = e(T) + 1 = n$ rounds. Hence, it remains to show that Waiter can indeed do so. To do this, we first define a vertex u to be a *stopping vertex* if $u \in f(S)$ and if for the vertex $t = f^{-1}(u)$ it holds that for every

$y \in N_{T'}(t) \backslash S$ we have $d_{T'}(y) = 1$. We first observe the following which will help us later to show that Waiter can follow the proposed strategy.

Observation 3.4.4. *Throughout Stages I and II, as long as Waiter can follow the proposed strategy, it holds that*

(i) $|A| \geqslant \mu' \sqrt{n} - 2$ *and* $e_{C \cup W}(A) = 0$,

(ii) $d_W(x, A) \leqslant d_C(x)$ *for every* $x \in f(S)$, *and*

(iii) $d_W(x, f(S')) \leqslant 1$ *for every* $x \in A$.

(iv) Assume that so far $\mathcal{O}_{T'}$ *did not consist solely of a stopping vertex. Then* $e_W(f(S'), A) \leqslant \varepsilon\sqrt{n} + 1$ *holds, and moreover, if* $e_W(f(S'), A) = \varepsilon\sqrt{n} + 1$ *holds at the end of any round, then at the end of the following round it holds that* $e_W(f(S'), A) \leqslant \varepsilon\sqrt{n}$.

Proof. Property (i) is proven analogously to Property (i) from Observation 3.4.3.

For Property (ii), observe that $d_W(x, A)$ can only increase when $x \in f(S)$ (since $e_{C \cup W}(A) = 0$) and Waiter offers an edge between x and a vertex $a \in A$. If the latter happens in Case 1 of Stage II (with x being one of the vertices u_1 and u_2), then the vertex a is removed from A by the update for that case and therefore $d_W(x, A)$ does not increase at all. Otherwise, in Stage I or in Case 2 or Case 3 of Stage II, Waiter actually offers two edges between x and A, which leads to an increase of $d_C(x)$ by one at the same time. Hence, $d_W(x, A)$ cannot become larger than $d_C(x)$.

Next let us consider Property (iii). During Stage I, Waiter did not claim any edges between A and $f(S')$, since $v \notin N_T(L')$. Hence, $d_W(x, f(S')) = 0$ holds for every $x \in A$ at the end of Stage I. Now we proceed by induction, looking at any round in Stage II. In Case 1, Waiter w.l.o.g. gets the edge au_2 with $u_2 \in f(S)$ and $a \in A$, but then a is moved from the set A to the set $f(S)$ (and hence maybe to the set $f(S')$) by the update of that case. However, since $e_W(a, A) = 0$ holds by Property (i) for the previous round, we conclude that $d_W(x, f(S'))$ stays unchanged for every x which remains in the set A. In Case 2 or Case 3 of Stage II, Waiter w.l.o.g. gets the edge ua_2 or ua_4 with $u \in f(S)$, Client gets the edge ua_1 or ua_3, while $a_i \in A$ for every $i \in [4]$, and then a_1 or a_3 is moved from the set A to the set $f(S)$ (and hence maybe to the set $f(S')$) by the update of that case. But then, in Case 2, since $e_W(a_1, A) = 0$ holds by Property (i) for the previous round, and since $u \notin f(S')$ by assumption of that case, we conclude that $d_W(x, f(S'))$ does not increase for any x which remains in the set A. Moreover, in Case 3, since $e_W(a_3, A) = 0$ holds analogously and $d_W(a_4, f(S')) = 0$ was true at the end of the previous round (by the choice of a_4 in that case), we conclude that after following Case 3 we have $d_W(a_4, f(S')) = 1$ and $d_W(x, f(S'))$ does not increase for any $x \neq a_4$ which remains in the set A. Hence, in either case, it holds that $d_W(x, f(S')) \leqslant 1$ for every vertex $x \in A$.

It remains to verify Property (iv). According to the previous discussion for Property (iii) we see that $e_W(f(S'), A)$ can only increase in Case 3 of Stage II, and (if this happens) it increases exactly by one. Hence, it is enough to show that if $e_W(f(S'), A) = \varepsilon\sqrt{n} + 1$ holds at the end of any round r and if at this moment $\mathcal{O}_{T'}$ does not consist of just a stopping vertex, then $e_W(f(S'), A) \leqslant \varepsilon\sqrt{n}$ will hold at the end of round $r+1$. So, assume the aforementioned conditions hold. Then, round r was played according to Case 3 of Stage II. That is, at the beginning of round r there was only one vertex $u \in f(S)$ which was open with regards to T', and moreover $u \in f(S')$. By assumption, the vertex u was not a stopping vertex, which means that for $t = f^{-1}(u)$ we could find a vertex $y \in N_{T'}(t)\backslash S$ such that $d_{T'}(y) \geqslant 2$. In round r, Waiter played according to Case 3 of Stage II, thus the vertex z with $tz \in E(T')$ (for the strategy described in Case 3) was chosen such that $d_{T'}(z) \geqslant 2$. Waiter then fixed vertices $a_3, a_4 \in A$ such that a_3u and a_4u were free and she offered these two edges to Client. By symmetry, we may assume that Client chose a_3u and the other edge a_4u was added to Waiter's graph. Then Waiter updated A, S, and f by removing a_3 from A, adding z to S, and setting $f(z) = a_3$. Using that $d_{T'}(z) \geqslant 2$, we conclude that a_3 needs to be open with regards to T' at the end of round r. Moreover, using Property (i), we also get that $d_W(a_3, A) = 0$ holds at this moment. For round $r+1$, there are two possible cases which may occur now.

The first case is that u is still open with regards to T' at the beginning of round $r+1$. Then this round is played according to Case 1 with $\{u_1, u_2\} = \{u, a_3\}$. If Waiter can follow her strategy, we already know from the discussion of Property (iii) that $d_W(x, f(S'))$ stays unchanged for every x which remains in the set A. However, by the strategy of Case 1, it also happens that Waiter picks a vertex $a \in A$ with $d_W(a, f(S')) \geqslant 1$ such that u_1a and u_2a are free. Note that such a vertex a exists, since by Property (iii) and under the assumption that $e_W(f(S'), A) = \varepsilon\sqrt{n} + 1$ there exist $\varepsilon\sqrt{n} + 1$ vertices $a \in A$ with $d_W(a, f(S')) \geqslant 1$, while

$$d_W(u, A) + d_W(a_3, A) \overset{\text{(ii)}}{\leqslant} d_C(u) + 0 \leqslant d_T(f^{-1}(u)) \leqslant \varepsilon\sqrt{n}\ .$$

At the end of round $r+1$ the vertex a is removed from A, and hence $e_W(A, f(S'))$ is reduced by $d_W(a, f(S')) \geqslant 1$.

The second case is that u is not open with regards to T' at the beginning of round $r+1$, and hence a_3 is the only open vertex with regards to T' at that point. Since $t = f^{-1}(u) \in N_T(L')$ holds and $tz \in E(T')$ holds for the vertex $z = f^{-1}(a_3)$, we know that $z \notin N_T(L')$ by the choice of M' and L'. Hence $a_3 \notin f(S')$ and thus, in round $r+1$, Waiter plays according to Case 2 (where she sets $u := a_3$), which means, that Waiter offers two edges a_1a_3 and a_2a_3 such that $a_1, a_2 \in A$ and such that $d_W(a_i, f(S')) \geqslant 1$ holds for $i \in [2]$ (which is possible since $d_W(a_3, A) = 0$ and since there exist $\varepsilon\sqrt{n} + 1$ vertices $a \in A$ with $d_W(a, f(S')) \geqslant 1$). By symmetry we may assume that Client chooses a_1a_3. Then a_1 is removed from A by the update of that case, making sure that $e_W(A, f(S'))$ is reduced by $d_W(a_1, f(S')) \geqslant 1$. □

With Observation 3.4.4 in hand, we can easily check that Waiter can follow the proposed strategy without forfeiting the game.

Stage I: According to her strategy, Waiter needs to offer $2d_T(v) \leqslant \frac{2n}{3}$ edges which are incident to v. She can easily do so, since there exist $n-1$ edges to choose from.

Stage II: Assume that Waiter needs to make a move according to Stage II, while she could follow her strategy in all of the previous rounds. Further, let us first assume that $\mathcal{O}_{T'}$ does not solely consist of a stopping vertex yet. In Case 1, when there exist $u_1, u_2 \in \mathcal{O}_{T'}$, we know that

$$d_W(u_1, A) + d_W(u_2, A) \overset{\text{(ii)}}{\leqslant} d_C(u_1) + d_C(u_2) \leqslant d_{T'}\big(f^{-1}(u_1)\big) + d_{T'}\big(f^{-1}(u_2)\big) \leqslant 2\varepsilon\sqrt{n} < |A|\,,$$

where the last inequality uses Property (i) and $\varepsilon < \frac{\mu}{20} < \frac{\mu'}{6}$. Hence, Waiter can find a vertex $a \in A$ such that u_1a and u_2a are free, and therefore she can follow her strategy in that case. In Case 2 and Case 3, when there exists a unique vertex $u \in \mathcal{O}_{T'}$ which is not a stopping vertex, we similarly conclude that $d_W(u, A) + e_W(A, f(S')) < |A| - 2$ by Properties (i)–(iv), and hence Waiter can find vertices a_1, a_2 or a_3, a_4 as required to follow her strategy.

Now let us assume that at some point $\mathcal{O}_{T'}$ consists solely of a stopping vertex u. Then, in order to finish Stage II, only the vertex u needs to get closed with regards to T'. Since this takes at most $d_{T'}\big(f^{-1}(u)\big) \leqslant \varepsilon\sqrt{n}$ rounds played according to Case 2 or Case 3, while the Properties (ii) and (iv) were true before that point, we know that until the end of Stage II, $e_W(f(S'), A)$ and $d_W(u, A)$ cannot exceed $2\varepsilon\sqrt{n} + 1$. But then we can analogously observe that

$$d_W(u, A) + e_W(f(S'), A) < 2\left(2\varepsilon\sqrt{n} + 1\right) < |A| - 2$$

holds, so it follows that Waiter can find vertices a_1, a_2 or a_3, a_4 as desired by her strategy.

Stage III: When Waiter enters Stage III, Client's graph is a copy of the subgraph T'. The sets $A := V(K_n)\backslash f(V(T'))$ and $B := f(S') = f(N_T(L'))$ both have sizes at least $\mu'\sqrt{n} = e(M')$. Moreover $e_W(A, B) < 2\varepsilon\sqrt{n}$ holds, as explained in the discussion of Stage II, and $e_C(A, B) = 0$. Thus the conditions for Theorem 3.1.1 are fulfilled, and (by following the strategy from Theorem 3.1.1) Waiter can force a perfect matching between A and B within $e(M') + 1$ rounds. □

Having Theorem 3.4.2 at hand, we are now able to prove Theorem 1.3.15 and Theorem 1.3.17.

Proof of Theorem 1.3.15. Let n be large enough, and let T be any tree on n vertices with maximum degree at most $\varepsilon\sqrt{n}$. The upper bound $\tau_{WC}(\mathcal{F}_T, 1) \leqslant n$ follows from Theorem 3.4.2; the lower bound $\tau_{WC}(\mathcal{F}_T, 1) \geqslant n - 1$ holds trivially since $e(T) = n - 1$.

If T is a path on n vertices, then $\tau_{WC}(\mathcal{F}_T, 1) = n - 1$. Indeed, in the strategy given for Theorem 3.2.1, Waiter forces a Hamilton path in the first round of Stage III, which is the $(n-1)^{\text{st}}$ round in the game. This shows that the lower bound in Theorem 1.3.15 is tight.

If T is a tree obtained from a path on $n - 4$ vertices by connecting two further vertices to each of its endpoints, then $\tau_{WC}(\mathcal{F}_T, 1) = n$. Indeed, if Waiter would want to force a copy of T within n rounds, then for some edge $e \in E(T)$ she would need to force a copy of $T - e$ within $n - 2$ rounds. However, since a unique edge extends this copy of $T - e$ to a copy of T, Client can easily prevent this copy of T in round $n - 1$, because Waiter needs to offer two edges. This shows that the upper bound is tight as well. □

Proof of Theorem 1.3.17. Let T be any tree on k vertices, and let H by a T-factor on n vertices with $k|n$. Then $\tau_{WC}(\mathcal{F}_{n,T-fac}, 1) \geqslant e(H) = \frac{k-1}{k}n$. In order to prove the upper bound on $\tau_{WC}(\mathcal{F}_{n,T-fac}, 1)$, let H' by an arbitrary tree on $n' = n+1$ vertices, obtained from H by adding one further vertex v' and adding exactly one edge between v' and each copy of T in H. Waiter then pretends to play on $K_{n'} \supset K_n$ with $V(K_{n'})\backslash V(K_n) = \{p'\}$. She plays according to the strategy given for Theorem 3.4.2 (with $v := v'$ and $p := p'$), and whenever this strategy makes her offer two edges incident to p', she only pretends to play that round. This way, she forces a copy of $H = H' - v'$ in the game on K_n, wasting at most one round, and hence she wins within $\frac{k-1}{k}n + 1$ rounds.

For the tightness of both bounds, we can use the same trees as in the proof of Theorem 1.3.15. For large enough k, if T is a path on k vertices and H is a T-factor, Waiter can win within $\frac{k-1}{k}n$ rounds. Before the game starts, she just splits the vertex set into $\frac{n}{k}$ sets of size k, and then on each of these parts she forces a Hamilton path without wasting a move. On the other hand, if T is a tree obtained from a path on $k-4$ vertices by connecting two further vertices to each of its endpoints, then analogously to the proof of Theorem 1.3.15 we get $\tau_{WC}(\mathcal{F}_{n,T-fac}, 1) = \frac{k-1}{k}n + 1$. □

3.5 Unbiased triangle factor game

In this section we present the proof of Theorem 1.3.19. However, before doing so, let us first prove the following lemma.

Lemma 3.5.1. *Consider an unbiased Waiter-Client game on K_{12}, and fix any two vertices $u, v \in V(K_{12})$. Then Waiter has a strategy that forces Client to create two vertex disjoint triangles with the following additional properties within seven turns:*

1. *Both u and v are in one of the two triangles, respectively.*
2. *All edges within the set of six vertices, that are not part of a triangle, as well as the edge uv have not been offered.*

Proof. Throughout the proof we will often use the fact that Waiter can offer two edges from the same vertex to two new vertices. Then, by symmetry, Client's choice does not affect the rest of Waiter's strategy. First Waiter forces Client to create two vertex disjoint cherries rooted at u and v respectively, which she can do because of the above remark and since we play on a total of 12 vertices, but in total only edges between 10 vertices are offered. Note that this leaves four vertices (which are not part of a cherry) connected to either u or v by an edge of Waiter. Suppose Client's graph now has the edge set $\{ub_1, ub_2, vb_3, vb_4\}$. Next Waiter offers b_1b_2 and b_3b_4, forcing Client to close a triangle. Without loss of generality, we may assume that Client chooses b_1b_2, closing a triangle containing u. Afterwards, Waiter offers two edges between v and both vertices to which no edge was offered yet, and denote the edge Client claimed by vw. Finally she offers the edges wb_3 and wb_4, forcing Client to close a second triangle, which this time contains v. Note that both of the properties are satisfied as claimed. □

Having Lemma 3.5.1 at hand, we will now proceed to prove Theorem 1.3.19.

Proof of Theorem 1.3.19. For the lower bound on $\tau_{WC}(\mathcal{F}_{n,K_3-fac}, 1)$ we will provide a strategy for Client. Throughout the game Client maintains a set of *marked* vertices $M \subset V$, which is initially empty. Moreover, we consider the sets $X_v := \{xy : xv, vy \in E(C)\}$ for every $v \in V(K_n)$ and the set $X = \bigcup_v X_v$ which consists of those edges which would close a triangle in Client's graph. In the following we describe Client's strategy.

In any round of the game suppose that Waiter offers two edges x_1y_1 and x_2y_2 to Client. Then Client chooses his edge according to the following case distinction.

Case 1: Suppose at least one of the offered edges belongs to $E(K_n)\backslash X$. Then Client arbitrarily chooses such an edge.

Case 2: Suppose otherwise that there exist $z_1, z_2 \in V(K_n)$ such that $x_1y_1 \in X_{z_1}$ and $x_2y_2 \in X_{z_2}$. Client then considers three subcases:

(a) If $z_1 \notin M$, Client chooses the edge x_2y_2 and adds z_1 to the set M.

(b) Otherwise, if $z_2 \notin M$, Client chooses the edge x_1y_1 and adds z_2 to the set M.

(c) Otherwise, if $z_1, z_2 \in M$, then Client chooses his edge arbitrarily.

It is obvious that Client can always follow the proposed strategy. Hence, it remains to show that it prevents a triangle factor for at least $\frac{13}{12}n$ rounds. We start with the following observation.

Observation 3.5.2. *At the end of the game, let $T = \{t_1, \ldots, t_{\frac{n}{3}}\}$ be a triangle factor in C and let $U = \{t \in T : t \cap M = \varnothing\}$ be the set of triangles, for which all three vertices are not marked. Then the following properties hold:*

(a) $\forall m \in M$ *we have* $d_C(m) \geqslant 3$.

(b) $|M| \geqslant |U|$.

Proof. To show (a), fix a vertex $m \in M$ and consider the turn in which Client added m to M. Then, according to Case 2 of the aforementioned strategy, at that point Client was offered an edge xy with $xm, my \in E(C)$, but did not add this edge to his graph, i.e. the triple $\{m, x, y\}$ does not form a triangle in C. However, since m needs to be in a triangle, $d_C(m) \geqslant 3$ follows.

To show (b), fix a triangle $t \in U$ and consider the turn in which Client completed this triangle. Then Client must have played according to Case 2(a) or Case 2(b), i.e. a case in which Client adds a vertex to M. Hence, $|M| \geqslant |U|$ follows immediately. □

Suppose that a triangle factor $T = \{t_1, \ldots, t_{\frac{n}{3}}\}$ is created in Client's graph. In order to conclude that at least $\frac{13}{12}n$ rounds have been played, we consider two cases. Assume first that right at this moment $|U| \geqslant \frac{n}{6}$. Then we have

$$2 \cdot |E(\mathcal{C})| = \sum_{v \in V} d_C(v) = \sum_{v \in M} d_C(v) + \sum_{v \in V \setminus M} d_C(v)$$
$$\geqslant 3 \cdot |M| + 2 \cdot |V \setminus M| = |M| + 2 \cdot |V| \geqslant \frac{n}{6} + 2n = \frac{13n}{6},$$

where the first inequality follows from Observation 3.5.2(a) and since every vertex belongs to a triangle, and the second inequality follows from Observation 3.5.2(b). Assume $|U| < \frac{n}{6}$ next, then

$$2 \cdot |E(\mathcal{C})| = \sum_{v \in V} d_C(v) = \sum_{t \in T} \sum_{v \in t} d_C(v) \geqslant 6 \cdot |U| + 7 \cdot \left(\frac{n}{3} - |U|\right) > \frac{7n}{3} - \frac{n}{6} = \frac{13n}{6},$$

where the first inequality follows from Observation 3.5.2(a). In any case, we obtain $|E(C)| \geqslant \frac{13}{12}n$.

Let us briefly mention, that this lower bound was recently improved by Dvořak (see [26]), who found a better strategy for Client, which made it possible to prolong the game even further. In fact, he could show that this new strategy for Client matches the upper bound which we will prove in the next step.

For the upper bound on $\tau_{WC}(\mathcal{F}_{n,K_3-fac}, 1)$ we will provide a strategy for Waiter. Therefore, let n_0 be a large enough even integer such that Waiter has a strategy to force a copy of K_{48} in the unbiased Waiter-Client game on K_{n_0}. Such an integer exists according to Beck [6]. Now, playing on K_n, fix any set of vertices $W \subset V(K_n)$ with $|W| = n_0$. In the following we describe Waiter's strategy. If at any point during the game, Waiter is unable to follow the strategy, she forfeits the game (we will later see that this does not happen). Waiter's strategy consists of the following three stages:

Stage I: Playing on $K_n[W]$ only, Waiter forces Client to create a clique of size 48.

Stage II: When Waiter enters Stage II, there exists a set $K \subset W$ of size 48 such that $C[K] \cong K_{48}$. Let $S = W \setminus K$ and $T = V \setminus W$. Waiter forces Client to create a large family of vertex disjoint triangles and she updates S and T by always removing the vertices of these triangles. As long as $|T| \geqslant 12$, she plays in sequences of at most 7 moves as follows:

Waiter first arbitrarily chooses 12 vertices in $S \cup T$, where she chooses exactly 2 vertices from S if $S \neq \varnothing$. Among these 12 vertices she fixes two vertices u, v, with $u, v \in S$ if $S \neq \varnothing$, and then she plays according to the strategy from Lemma 3.5.1 on these 12 vertices. By doing this, two triangles are created in Client's graph. Waiter then removes the vertices which belong to these triangles from S and T, respectively.

Stage III: When Waiter enters Stage III, we have $|T| < 12$. Then for each of the vertices $v \in T$ Waiter picks pairwise disjoint sets K_v of four vertices in K and offers the edges from $E_{K_n}(v, K_v)$ in pairs.

It is clear that if Waiter can follow the proposed strategy without forfeiting the game, she forces a triangle factor within at most $\binom{n_0}{2} + \frac{7}{6}n + 2 \cdot 12 = \frac{7}{6}n + O(1)$ rounds. Therefore, it remains to be verified that Waiter can indeed follow the proposed strategy.

Strategy discussion:

The strategy in Stage I can be followed by the choice of n_0. Note that when Stage I is over, no edge from $E_{K_n}(T, S \cup T)$ has been offered yet.

For Stage II assume that before Waiter plays a sequence of moves as described in her strategy it is still true that all edges in $E_{K_n}(T, S \cup T)$ are free. It then follows from Lemma 3.5.1 that Waiter can play her next moves according to the strategy of Stage II. Just note that in the case when $S \neq \varnothing$ it can happen that $uv \in E_{K_n}(S)$ has already been offered before; but this does not cause any problem, since for Lemma 3.5.1 Waiter does not need to offer uv at all. Finally also note that when the two triangles are created, by (2) from Lemma 3.5.1 and by the update in Stage II it follows that $E_{K_n}(T, S \cup T)$ again consists only of free edges. Hence, Waiter can repeatedly apply Lemma 3.5.1 and follow her strategy for Stage II.

Afterwards, when Waiter enters Stage III, it holds that $S = \varnothing$ and $|T| < 12$. Since $E_{K_n}(K, T)$ consists solely of free edges at this point, Waiter can offer edges as desired. □

3.6 Biased games

To finish this chapter we will consider biased games, in particular we will study the perfect matching game and the Hamiltonicity game once again, but this time in their biased version (see Theorem 1.3.4 and Theorem 1.3.11, respectively). In the proofs of those two theorems we will make use of the following result due to Bednarska-Bzdęga, Hefetz, Krivelevich, and Łuczak [9].

Theorem 3.6.1 (Theorem 1.4(ii) in [9]). *There exists a positive constant $c \in (0, 1)$ and an integer n_0 such that the following holds. If $n \geqslant n_0$ and $b \leqslant cn$, then playing a b-biased Waiter-Client game on $E(K_n)$, Waiter has a strategy to force Client to claim a spanning pancyclic graph.*

We will consider the Hamiltonicity game first, and our first proof in this section will be a proof of Theorem 1.3.11.

Proof of Theorem 1.3.11. Let c and n_0 be given according to Theorem 3.6.1. We set

$$C_0 = 100 \max\{c^{-1}, n_0\}, \quad \delta_0 = 0.1c, \quad \delta = 0.01 \min\{\delta_0, C_0^{-1}\}, \quad C = C_0\delta^{-1}. \tag{3.6.1}$$

We let $b \leqslant \delta n$ from now on, and whenever necessary we will assume n to be large enough.

In the following we will describe a strategy for Waiter in the b-biased Waiter-Client game on K_n, and afterwards we will show that it is a strategy with which Waiter forces Client to claim a Hamilton cycle within at most $n + Cb$ rounds. Whenever Waiter is not able to follow the proposed strategy, she forfeits the game (we will later see that this does not happen). The strategy is split into five stages.

Stage I: Within $n - C_0b - 1$ rounds, Waiter forces a path $P = (a_1, \ldots, a_{n-C_0b})$ on $n - C_0b$ vertices according to the following rule:

Initially set $P = \{a_1\}$ for an arbitrary vertex $a_1 \in V(K_n)$. Assume that after $i - 1$ rounds Waiter has already forced a path $P = (a_1, \ldots, a_i)$ on i vertices. Then in round i Waiter selects

$b+1$ vertices $x_1, \ldots, x_{b+1}$ from $V \backslash V(P)$ which have the smallest degree in her graph. Then she offers the edges $\{a_i x_j : j \in [b+1]\}$ to Client, of which he needs to pick one. Then Waiter updates the path such that $P = (a_1, \ldots, a_{i+1})$ with $a_{i+1} := x_j$.

Once P has reached length $n - C_0 b - 1$, Waiter proceeds with Stage II.

Stage II: Let P denote the path that Client claimed by the end of Stage I, and set $R = V \backslash V(P)$. Playing only on $K_n[R]$, within at most Cb rounds, Waiter forces a Hamilton cycle of $K_n[R]$. The details of how she can do this, can be found later in the strategy discussion. Afterwards, Waiter proceeds with Stage III.

Stage III: This stage lasts one round. Let $P = (a_1, \ldots, a_{n-C_0 b})$ be the path from Stage I and let H be the Hamilton cycle from Stage II. Waiter now picks $b+1$ vertices $x_1, \ldots, x_{b+1} \in V(H)$ such that $a_1 x_j$ is free for every $j \in [b+1]$. Then she offers all of these $b+1$ edges to Client, who has to claim $a_1 x_j$ for some $i \in [b+1]$. From now on, set $\tilde{x} = x_j$ and let x be one of the neighbours of x_j on H. Next Waiter proceeds with Stage IV.

Stage IV: This stage lasts exactly b rounds in which Waiter forces a few Hamilton paths on $V(K_n) \backslash R$ using Pósa rotations [60]. More precisely, let $P_0 = P = (a_1, \ldots, a_{n-C_0 b})$ be the path from Stage I, and set $v_0 = a_{n-C_0 b}$. By playing only on $K_n[V(P_0)]$, for $i \in [b]$ Waiter ensures that immediately after the i^{th} round in Stage IV Client's graph contains a path P_i such that the following properties hold:

$(A1)$ $V(P_i) = V(P_0)$,

$(A2)$ P_i has endpoints a_1 and $v_i \in V \backslash \{v_0, \ldots, v_{i-1}\}$,

$(A3)$ $v_i x$ is free.

The details of how Waiter can do this can be found later in the strategy discussion. Afterwards, Waiter proceeds with Stage V.

Stage V: Within one round, Waiter forces Client's graph to contain a Hamilton cycle of K_n by offering edges between endpoints of P_i and x. The exact details of how she can do this can be found later in the strategy discussion.

If Waiter can follow the proposed strategy without forfeiting the game, then it is obvious that she forces a Hamilton cycle within at most $(n - C_0 b - 1) + Cb + 1 + b + 1 < n + Cb$ rounds. Therefore, it remains to show that Waiter can indeed always follow the proposed strategy.

Strategy discussion:

Stage I: Consider the round $i \in [n - C_0 b - 1]$ in Stage I. Up to this point Waiter already forced a path $P = (a_1, \ldots, a_i)$. Since in the previous rounds Waiter only offered edges which are incident to at least one of the vertices a_j with $j < i$, we know that before Waiter's i^{th} turn all the edges between a_i and $V \backslash V(P)$ are free. Moreover, since $|V \backslash V(P)| \geqslant C_0 b$, Waiter can easily find and offer $b+1$ edges as required by the strategy.

Stage II: Let $P = (a_1, \ldots, a_{n-C_0 b})$ denote the path which Waiter has forced at the end of Stage I. Since so far she only offered edges which are incident to at least one of the vertices a_j with $j < n - C_0 b$, we know that at the beginning of Stage II all the edges inside $R = V \backslash V(P)$ are still free. Let $\tilde{n} := |R| = C_0 b$ and, using (3.6.1), observe that $b = C_0^{-1} \tilde{n} < c\tilde{n}$ and $\tilde{n} > n_0$.

According to Theorem 3.6.1, Waiter has a strategy for playing a $(\tilde{b} : 1)$ game on $K_n[R]$ with bias $\tilde{b} = c\tilde{n}$ in such a way that Client is forced to obtain a pancyclic graph. Thus, following that strategy with bias $b < \tilde{b}$ (by pretending to add $\tilde{b} - b$ extra edges to Waiter's graph in each round), Waiter can force a Hamilton cycle on R within at most

$$\left\lceil \frac{\binom{|R|}{2}}{\tilde{b}+1} \right\rceil < \frac{|R|^2}{\tilde{b}} = \frac{\tilde{n} \cdot C_0 b}{c\tilde{n}} = C_0 c^{-1} b \overset{(3.6.1)}{<} Cb$$

rounds, as promised in the strategy description.

Stage III: When Waiter enters Stage III, it holds that $d_{C \cup W}(a_1) = b + 1$, since only in the very first rounds she offered edges incident to a_1. The number of available edges between a_1 and R is at least $|R| - d_{C \cup W}(a_1) = C_0 b - (b+1) > b + 1$ by the choice of C_0. Hence, Waiter can offer edges as required for this stage of the proposed strategy.

Stage IV: Before we will show that Waiter can follow Stage IV, let us first observe that at this point none of the vertices has a degree that is too large.

Observation 3.6.2. *At the beginning of Stage IV it holds that $d_{C \cup W}(v, V(P_0)) < \delta_0 n$ for every vertex $v \in V(K_n)$.*

Proof. When Waiter forces the path P in Stage I, she always prefers to offer edges from the current endpoint a_i to the vertices of smallest Waiter-degree in $V \backslash V(P)$. This way she makes sure that the Waiter-degrees among the vertices in $V \backslash V(P)$ differ by at most 1 throughout Stage I. Since Stage I lasts $n - Cb - 1$ rounds, it holds that $e(W) \leqslant n(b+1)$ throughout Stage I. In particular, all vertices $v \in V \backslash V(P)$ then satisfy

$$d_{C \cup W}(v) < \left\lceil \frac{2e(W)}{|V \backslash V(P)|} \right\rceil < \frac{2n(b+1)}{C_0 b} \overset{(3.6.1)}{<} 0.5\delta_0 n\,.$$

Now consider the beginning of Stage IV. It holds that $d_{C \cup W}(a_1) = 2(b+2) < 3\delta n < 0.1\delta_0 n$, since there were only two rounds in which Waiter offered edges at a_1. For every vertex $v \in V(P) \backslash \{a_1\}$ we then have $d_{C \cup W}(v) < 0.5\delta_0 n + (b+1) < \delta_0 n$, since after v was added to P there was only one round in which Waiter offered edges incident to v. Moreover, for every remaining vertex v (i.e. $v \in R$), we have $d_{C \cup W}(v, V(P_0)) < 0.5\delta_0 n + 1 < \delta_0 n$ since these vertices belong to $V \backslash V(P)$ at the end of Stage I and since afterwards, until the end of Stage III, the edge va_1 may be the only edge between v and $V(P)$ that got offered. This proves the observation. □

Having this observation at hand, we now show how Waiter can force the desired paths in Stage IV. We will do this in such a way that the Properties $(A\,1)$–$(A\,3)$ as well as the following property hold:

$(D1)$ $d_{C \cup W}(v, V(P_i)) < \delta_0 n + i$ for every $v \in V(P_i) \backslash \{v_0, \ldots, v_{i-1}\}$.

We proceed by induction on i. For $i = 0$ the path $P_0 = P$ from Stage I trivially satisfies $(A1)$ and $(A2)$. Property $(D1)$ follows from Observation 3.6.2. Moreover, Property $(A3)$ holds because of the following reason: In Stage I every offered edge is incident to at least one of the vertices in $V(P_0)\backslash\{v_0\}$, in Stage II every edge is disjoint from $V(P_0)$, and in Stage III every edge is incident to $a_1 \notin \{v_0, x\}$. Thus, v_0x has not been offered yet.

So, let $i > 0$ then. Let P_{i-1} be the path given by induction and consider the i^{th} move in Stage IV. For every vertex $v \in V(P_{i-1})\backslash\{v_{i-1}\}$ denote with v^+ the unique neighbour of v in P_{i-1} with $\text{dist}_{P_{i-1}}(v^+, v_{i-1}) = \text{dist}_{P_{i-1}}(v, v_{i-1}) - 1$. Set

$$\begin{aligned} B_1 &:= \left\{y \in V(P_{i-1}) : \ y^+x \in C \cup W\right\}, \\ B_2 &:= \left\{y \in V(P_{i-1}) : \ y^+ = v_j \text{ for some } j < i\right\}, \\ B_3 &:= \{y \in V(P_{i-1}) : \ yv_{i-1} \in C \cup W\}. \end{aligned}$$

By Observation 3.6.2 and since in Stage IV Waiter only offers edges in $V(P_0)$, we obtain $|B_1| < \delta_0 n$. Since $i \leqslant b$, we have $|B_2| \leqslant i \leqslant b < \delta_0 n$. Using Property $(D1)$ we get $|B_3| < \delta_0 n + i \leqslant 2\delta_0 n$. Hence, also using (3.6.1), we conclude that

$$\begin{aligned} |V(P_0)\backslash(B_1 \cup B_2 \cup B_3)| &\geqslant n - C_0 b - 4\delta_0 n \geqslant n - C_0\delta n - 4\delta_0 n \geqslant n - 0.01n - 0.4n \\ &\geqslant b + 1, \end{aligned}$$

provided that n is large enough. Waiter's strategy now is to offer $b + 1$ edges of the form yv_{i-1} with $y \in V(P_0)\backslash(B_1 \cup B_2 \cup B_3)$, which is possible since $y \notin B_3$ implies that yv_{i-1} is free. Client then needs to claim one of these edges; by abuse of notation let us denote this edge with yv_{i-1}.

Then set $v_i := y^+$ and let P_i be the path induced by $(E(P_{i-1}) \cup \{yv_{i-1}\})\backslash\{yy^+\}$. Property $(A1)$ is trivially satisfied. Moreover, P_i has endpoints a_1 and $y^+ = v_i$, and since $y \notin B_2$ we have that Property $(A2)$ holds as well. Property $(A3)$ is guaranteed since $y \notin B_3$. To show Property $(D1)$ observe the following: In her i^{th} move of Stage IV, Waiter only offers edges incident to v_{i-1}. Thus, the degrees $d_{C\cup W}(v, V(P_{i-1})) = d_{C\cup W}(v, V(P_i))$ can increase by at most 1 for every $v \neq v_{i-1}$.

Stage V: Let $v_0, \ldots, v_b$ be the distinct endpoints of the paths $P_0, \ldots, P_b$ from Stage IV. Each of the $b + 1$ edges $v_i x$ with $0 \leqslant i \leqslant b$ is free, and therefore Waiter can offer those in her next move. Let $v_j x$ be the edge claimed by Client in this round, and let $f = a_1\tilde{x}$ be the edge which Client claimed in Stage III. Then

$$(E(P_j) \cup E(H) \cup \{f, v_j x\})\backslash\{x\tilde{x}\}$$

is a Hamilton cycle of K_n that consists only of edges claimed by Client. □

Finally, we finish this section with the proof of Theorem 1.3.4.

Proof of Theorem 1.3.4. Creating a perfect matching under the assumption of Theorem 3.6.1, is rather straightforward. Let δ_0, δ, C_0, and C be defined as in the previous proof. For roughly $0.5(n - C_0 b)$ rounds, Waiter can force a large matching, by playing according to Stage I from the previous proof where a long path was created, and just faking every second move (i.e. pretending to make the move, but not playing at all). Then, as in Stage II of the previous proof, she forces a Hamilton cycle on the remaining vertices within Cb rounds, thus a perfect matching is created. □

3.7 Concluding remarks

Winning as fast as possible. As already mentioned in the introduction, there are quite a few games that Waiter wins (almost) perfectly fast. Also, for almost every game that we considered in this chapter, we were able to prove that Waiter can win at least asymptotically fast. On the other hand, for the triangle factor game we know that this game is not won asymptotically fast.

For every game except for the triangle factor game we presented proofs for matching upper and lower bounds. The obvious gap in our result about triangle factors was recently closed by Dvořak [26], who proved that in fact it holds that $\tau_{WC}(\mathcal{F}_{n,K_3-fac}, 1) = \frac{7}{6}n + o(n)$.

Minimum degree k game. A different game that we did not consider is the *minimum degree k game* $\mathcal{D}_{n,k}$ played on K_n, in which the winning sets consist of all spanning subgraphs H with $\delta(H) \geqslant k$. We considered the unbiased and biased version of the perfect matching and Hamiltonicity game in this thesis, which covers the minimum degree 1 and minimum degree 2 games. We also have an argument that would show that Waiter can win the unbiased game with winning sets $\mathcal{D}_{n,k}$ within $\frac{kn}{2} + O(1)$ rounds, basically by creating perfect matchings on appropriately chosen subgraphs of K_n. We wonder whether this can be improved as follows:

Problem 3.7.1. *For $k > 2$, show that $\tau_{WC}(\mathcal{D}_{n,k}, 1) = \lfloor \frac{kn}{2} \rfloor + 1$. Moreover, determine $\tau_{WC}(\mathcal{D}_{n,k}, b)$ asymptotically when $b > 1$.*

Note that Maker can win the unbiased Maker-Breaker version of the game $\mathcal{D}_{n,k}$ within at most $\lfloor \frac{kn}{2} \rfloor + 1$ rounds, as was shown by Ferber and Hefetz in [31], therefore we are curious to know whether this holds in the Waiter-Client setup as well.

Clique factor game. Another game which we did not consider in this thesis is the *k-clique factor game* for $k \geqslant 3$. The triangle factor game covers the case $k = 3$, but recently Dvořak [27] found the following upper and lower bounds for the general k-clique factor game:

Theorem 3.7.2 (Theorem 1.2 in [27])**.** *There exist functions $n_0(k), C(k)$ such that one has*

$$2^{k/3-o(k)}n \leqslant \tau_{WC}(\mathcal{F}_{n,K_k-fac}, 1) \leqslant \frac{2^k}{k}n + C(k)$$

for $n \geqslant n_0(k)$ and divisible by k.

Further results involving trees. In Section 3.4 we found a fast winning strategy for Waiter in the case where she wants to force a copy of a given spanning tree that fulfils some maximum degree condition. When we do not intend to win as fast as possible, we can even relax this maximum degree condition. Waiter's fast winning strategy that we studied in this thesis enables her to force trees with one vertex of degree at most $\frac{n}{3}$, but every other vertex needs to have a degree bounded from above by $O(\sqrt{n})$. Recently, in [2], we found a strategy that allows Waiter to create trees with maximum degree almost $\frac{n}{3}$, but with no restriction on the number of vertices with very high degrees (see Theorem 1.3.16). But the strategy is also slower than the strategy from Theorem 1.3.15 and requires Waiter to play for $O(\sqrt{n})$ additional rounds.

Also, in [1], we studied what Waiter might be able to achieve in a slightly different setting, where we lower the upper bound on the maximum degree a little bit further and do not value speed at all. We were able to show that in this setting Waiter can claim not just one specific tree, but she can even claim a graph which contains every tree T with $\Delta(T) \leqslant \frac{cn}{\ln(n)}$ (for some constant c) as a subgraph. This result can be summarised in the following theorem:

Theorem 3.7.3 (Theorem 1.2 in [1])**.** *There exists a constant $c > 0$ such that the following holds for every large enough integer n. In the $(1:1)$ Waiter-Client game on K_n, Waiter has a strategy to force Client to claim a graph which contains a copy of every tree T with n vertices and maximum degree $\Delta(T) \leqslant \frac{cn}{\ln(n)}$.*

We were also able to prove the same result for the Maker-Breaker version of this game and thus could show that Maker/Waiter can play as least as good as the random graph intuition suggest. We do not know yet, if this result is best possible. While we conjecture that this is actually the best Maker can do, we believe that Waiter should even be able to claim a graph which contains every tree with a linear maximum degree.

To finish this section (and also this chapter), the last question to think about would be what happens in the biased version of the fixed spanning tree game, when the bias depends on n.

Problem 3.7.4. *Determine $\tau_{WC}(\mathcal{F}_T, b)$ asymptotically when $b > 1$.*

Chapter 4

Connector-Breaker and Walker-Breaker Games

In this chapter we will study Connector-Breaker games and Walker-Breaker games on random graphs. Since Connector and Walker have similar rulesets for their turns (basically Walker is an even more restricted Connector, see Subsection 1.2.7 and Subsection 1.2.8), we will combine our results for both games in this chapter, which we split into two larger sections.

Our main goal in this chapter is to prove Theorem 1.3.20, which we will accomplish by proving both Theorem 1.3.21 and Theorem 1.3.22. Let us note that from these two results Theorem 1.3.20 follows trivially. We will present proofs for each of the two theorems in their own section.

In the first section (Section 4.1) we will study Connector-Breaker games. Our goal is to present a proof for Theorem 1.3.21, and thus we will focus on a strategy for Breaker. We will define sets of so-called *bad vertices* for Connector and provide an algorithm to find these vertices in a random graph $G \sim G_{n,p}$. These bad vertices will then be merged into a *bad structure*. Then we will present a strategy for Breaker, which makes use of such a bad structure to show that after Connector's first turn and if $p = n^{-2/3-o(1)}$ Breaker can choose a vertex in a specific way, which he then prevents Connector from ever reaching, thus making sure that Connector will not be able to claim any spanning structure in G. The results of this section are joint work with Dennis Clemens and Laurin Kirsch [21].

In the second section (Section 4.2) we will consider Walker-Breaker games. This time, we will focus on a strategy for Walker and present a proof for Theorem 1.3.22. We will define *good structures* in a random graph $G \sim G_{n,p}$ that Walker can use to reach any vertex in G in a constant number of turns, even if Breaker already claimed many edges. The strategy for Walker makes use of these structures to not just claim a Hamilton cycle, but even to claim a subgraph of G that behaves almost like a random graph, if $p = n^{-2/3+o(1)}$. This yields an even stronger result, which we will sum up in Theorem 4.2.7. Lastly, we will use this result in conjunction with a result by Lee and Sudakov [53] about local resilience (Theorem 2.4.2) to deduce Theorem 1.3.22. The results of this section are joint work with Dennis Clemens and Pranshu Gupta [19].

4.1 Connector-Breaker games: Breaker's strategy

In this section we will consider a $(2:2)$ Connector-Breaker game played on $G \sim G_{n,p}$ with $p = n^{-2/3-\varepsilon}$ and $\varepsilon \in (0,1)$. Our goal is to find a strategy for Breaker, which he can follow to a.a.s. isolate a vertex from Connector's graph. This vertex, which we will denote by x, is chosen by Breaker after Connector's first turn.

First we iteratively define a set B^x of vertices that are *bad* for Connector with respect to the goal of reaching x. This set of bad vertices (which we will also call a *bad structure*) will be key to Breaker's strategy. If x is carefully chosen (which we will manage to do later), then Breaker has a strategy to make sure that Connector in her move either does not even reach B^x, or if she reaches B^x, then Breaker can immediately destroy all potential threads. More details will be given later.

4.1.1 Defining bad vertices

First let us present Algorithm 1, which describes how B^x is constructed.

Algorithm 1: Bad vertex set B^x for given vertex x

Input : graph G and vertex $x \in V(G)$

Output : number of iterations r_x, bad vertex set $B^x = \bigcup_{k \leqslant r_x} B_k^x$

$B_1^x := N_G(x)$;

$B^x := B_1^x$;

for $i \geqslant 2$ **do**

    $B_i^x := \{v \notin B^x \cup \{x\} : \ d_G(v, B^x) \geqslant 2\}$;

    $B^x \leftarrow B^x \cup B_i^x$

  if $B_i^x = \varnothing$

    then halt with output $B_1^x, \dots, B_{i-1}^x, B^x$ and $r_x = i-1$;

end

Basically, this algorithm creates a structure by iteratively adding layers to the existing structure, starting from the vertex x. The first layer of our bad structure then becomes $N_G(x)$, and then a new layer is created by taking each vertex which has at least two neighbours in the existing bad structure and adding it to this layer, which will then be added to the bad structure. Then the process will start over for the next layer, until at some point there are no more vertices which have a degree of two or more into the bad structure.

For now let us just state the algorithm here. An in-depth analysis will follow in Subsection 4.1.3. Let us also state the following lemma, which will be crucial for Breaker's strategy.

Lemma 4.1.1. *Let n be a large enough integer and let $\varepsilon \geqslant \frac{7\ln(\ln(n))}{\ln(n)}$. For $p = n^{-2/3-\varepsilon}$ generate $G \sim G_{n,p}$. Then a.a.s. G satisfies the following property: For every set $M \subset V(G)$ of size 3, there exists a vertex x such that Algorithm 1 produces a set B^x of vertices and a sequence $(B_1^x, \dots, B_{r_x}^x)$ of disjoint subsets of B^x such that the following holds:*

$(B1)$ *$B_1^x = N_G(x)$ and $e_G(B_1^x) = 0$,*

$(B2)$ *for every $2 \leqslant i \leqslant r_x$ and every vertex in $v \in B_i^x$ we have $d_G\big(v, \bigcup_{k \leqslant i} B_k^x\big) = 2$,*

$(B3)$ *for every vertex $v \in V \backslash (B^x \cup \{x\})$ it holds that $d_G\big(v, B^x\big) \leqslant 1$,*

$(B4)$ *$B^x \cap (M \cup N_G(M)) = \varnothing$.*

This lemma basically tells us, that for any three vertices of G there exists a vertex x and a corresponding bad structure B^x, such that every vertex in B^x has degree exactly two into some lower levels, and there also exist no edges inside of the layers of B^x. Furthermore, this structure does not overlap with the three vertices from the beginning or with their neighbourhoods, and every vertex outside of B^x has at most one edge incident to the bad structure (with the exception of x).

We postpone the proof of the above lemma to Subsection 4.1.3 as well and recommend to read Breaker's strategy first.

4.1.2 The strategy for Breaker

This subsection is dedicated to the proof of Theorem 1.3.21. Let Connector and Breaker play any $(2:2)$ game on $G \sim G_{n,p}$. We will show that, under the condition that the property described in Lemma 4.1.1 holds, Breaker has a strategy that isolates a vertex from Connector's graph. Let V_C^r denote the set of vertices that are covered by Connector's edges at the end of round r. Immediately after Connector's first move, we have $|V_C^1| = 3$ and thus, by the property from Lemma 4.1.1 (applied with $M = V_C^1$), we find a vertex x such that Algorithm 1 produces a set B^x of vertices and a sequence $(B_1^x, \ldots, B_{r_x}^x)$ of disjoint subsets of B^x such that the Properties $(B1)$–$(B4)$ hold with $M = V_C^1$. Notice that at this point $x \notin V_C^1 \cup N_G(V_C^1)$ holds, according to $(B4)$ and since $N_G(x) \subset B^x$.

In order to simplify notation, let $B_0^x := \{x\}$ and set $B_{<i}^x := \bigcup_{\ell=0}^{i-1} B_\ell^x$ as well as $B_{\leqslant i}^x := \bigcup_{\ell=0}^{i} B_\ell^x$. Breaker's strategy is to make sure that for each round r, immediately after his move the following property holds for every free edge vw:

$(Q1)$ If there exists $0 \leqslant i \leqslant r_x$ such that $v \in (N_G(V_C^r) \backslash V_C^r) \cap B_i^x$ and $w \in V_C^r$ hold, then $w \in B_{<i}^x$.

Let us first observe that Breaker keeps x isolated in Connector's graph, if he is able to maintain $(Q1)$ for every free edge after each of his moves. Assume this is not the case, i.e. there is some round r in which Connector reaches vertex x. Then immediately after Breaker's $(r-1)^{\text{st}}$ move, we have that $(Q1)$ holds for every free edge, and still $x \notin V_C^{r-1}$. From this it follows that immediately before Connector's r^{th} move there cannot be a free edge xw with $w \in V_C^{r-1}$. Indeed, otherwise we would need $x \in (N_G(V_C^{r-1}) \backslash V_C^{r-1}) \cap B_0^x$, and by $(Q1)$ we would get $w \in B_{<0}^x = \varnothing$, a contradiction. Thus, in order to reach x during round r, Connector would need do claim a path (w, v, x) of length 2, starting at some vertex $w \in V_C^{r-1}$ and ending in x. It then follows that $v \in (N_G(V_C^{r-1}) \backslash V_C^{r-1}) \cap B_1^x$. However, using $(Q1)$ for the free edge wv at the end of round $r-1$, this would give $w \in B_{<1}^x = B_0^x$ and therefore $x = w$, a contradiction.

Hence, we know that Connector cannot reach x as long as Breaker always restores $(Q1)$ for every free edge. It thus remains to show that Breaker can indeed do so. We proceed by induction.

For the first round, observe that immediately after Connector's first move, there is no edge between V_C^1 and $B^x \cup \{x\}$, according to Property $(B4)$ (with $M = V_C^1$). Thus, Property $(Q1)$ holds for every free edge at the end of this round, independent of what Breaker claims in his first move, as there does not exist any edge vw as described in that property.

Now let us assume, that $(Q1)$ is satisfied immediately after Breaker's $(r-1)^{\text{st}}$ move for every free edge, and let us explain how Breaker is able to restore $(Q1)$ in the next round. Without loss of generality we may assume that in round r Connector reaches exactly two new vertices, say w_1 and w_2, i.e. $V_C^r = V_C^{r-1} \cup \{w_1, w_2\}$.

If there exist at most two free edges after Connector's r^{th} move that fail to satisfy Property $(Q1)$ (with $V_C = V_C^r$), then Breaker can claim these edges and therefore easily ensures that $(Q1)$ holds for every free edge at the end of round r. So, assume for a contradiction that immediately after Connector's r^{th} move there are at least three free edges that do not satisfy $(Q1)$. All of these edges need to be incident to w_1 or w_2, since before Connector's move Property $(Q1)$ was true for every free edge (where $V_C = V_C^{r-1}$). Without loss of generality let w_2 be incident to at least two of these edges, say w_2v_1 and w_2v_2. Since for these edges $(Q1)$ fails to hold after Connector's r^{th} move, we have $v_1 \in (N_G(V_C^r) \backslash V_C^r) \cap B_{i_1}^x$ and $v_2 \in (N_G(V_C^r) \backslash V_C^r) \cap B_{i_2}^x$ for some $0 \leqslant i_1, i_2 \leqslant r_x$, while $w_2 \in V_C^r$ and $w_2 \notin B_{<i}^x$, with $i := \max\{i_1, i_2\}$. Now, since w_2 has two neighbours in $B^x \cup \{x\}$, Algorithm 1 at some point must have added w_2 to B^x. Thus, we conclude that $w_2 \in B_k^x$ for some $k \geqslant \max\{i_1, i_2\}$.

First we consider the case that in round r Connector reaches w_2 by claiming a free edge yw_2 with $y \in V_C^{r-1}$. Then $y \notin \{v_1, v_2\}$. Moreover, $w_2 \in N_G(V_C^{r-1}) \backslash V_C^{r-1}$ and, since $(Q1)$ was true for yw_2 at the end of round $r-1$ (with $V_C = V_C^{r-1}$), we conclude that $y \in B_{<k}^x$. But this means that $w_2 \in B_k^x$ has three neighbours in $B_{\leqslant k}^x$ (namely v_1, v_2, and y), a contradiction to $(B2)$.

Next we consider the case that in round r Connector reaches w_2 in another way than in the first case. This can be done by claiming a path (y, w_1, w_2) with $y \in V_C^{r-1}$ and $w_1 \in N_G(V_C^{r-1}) \backslash V_C^{r-1}$. According to Property $(B2)$ we know that $w_2 \in B_k^x$ has exactly two neighbours in $B_{\leqslant k}^x$, and these neighbours need to be v_1 and v_2. It follows that the third edge, which does not satisfy $(Q1)$ immediately before Breaker's r^{th} move, cannot be incident to w_2 and thus needs to be of the form v_3w_1 with $v_3 \in (N_G(V_C^r) \backslash V_C^r) \cap B_{i_3}^x$ for some $0 \leqslant i_3 \leqslant r_x$. Then $v_3, w_2 \in B^x$ are two neighbours of w_1 and hence Algorithm 1 must have added w_1 to B^x at some point, say $w_1 \in B_t^x$. Since again $w_2 \in B_k^x$ has exactly two neighbours in $B_{\leqslant k}^x$ and these are v_1 and v_2, we must have $w_1 \notin B_{\leqslant k}^x$, i.e. $t > k$. But now, by induction, Property $(Q1)$ was true for the free edge yw_1 at the end of round $r-1$, and thus $y \in B_{<t}^x$. Moreover, because we assumed v_3w_1 to be an edge not satisfying $(Q1)$ after Connector's r^{th} move, we have $w_1 \notin B_{<i_3}^x$ and thus $i_3 \leqslant t$. Hence, we obtain that the three neighbours w_2, v_3, y of $w_1 \in B_t^x$ belong to $B_{\leqslant t}^x$, since we have $w_2 \in B_k^x \subset B_{\leqslant t}^x$, $v_3 \in B_{i_3}^x \subset B_{\leqslant t}^x$, and $y \in B_{<t}^x$. This again leads to a contradiction to $(B2)$. □

4.1.3 Analysis of Algorithm 1

The goal of this subsection is to prove Lemma 4.1.1. We will not prove this lemma directly, but instead we will prove the slightly more general technical Lemma 4.1.2 (stated below), from which Lemma 4.1.1 follows. For Lemma 4.1.2 we are going to apply Algorithm 1 to a set $A = \{x_1, \dots, x_t\}$ of vertices, from which we will carefully choose one to obtain a vertex x as promised by Lemma 4.1.1. That is, we first fix x_1 and apply Algorithm 1 in order to determine the set B^{x_1}, then we repeat the algorithm for x_2, and so on. Amongst other properties we will obtain that it is very likely that all the sets B^{x_j} are pairwise disjoint and satisfy certain degree conditions. To simplify notation we set

$$B^{(j,i)} := \bigcup_{\ell<j} B^{x_\ell} \cup \bigcup_{k \leqslant i} B_k^{x_j} \,. \tag{4.1.1}$$

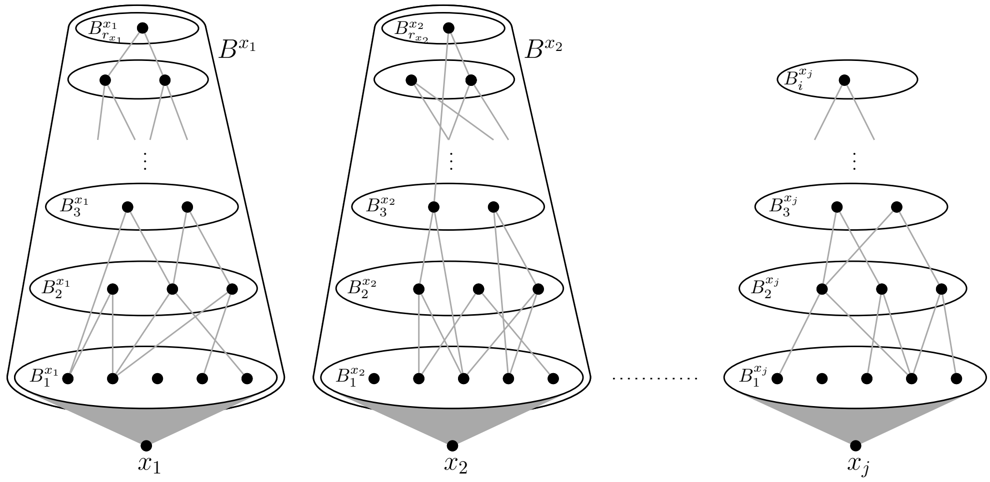

Figure 4.1.1: Structure of $B^{(j,i)}$

That is, $B^{(j,i)}$ is the set of all bad vertices that are determined immediately after $B_i^{x_j}$ is created. In particular, $B^{(t,r_{x_t})} = \bigcup_{x \in A} B^x$ is the union of all bad vertices after the algorithm was applied to all vertices x_j. Moreover, we let

$$a(j,i) := \begin{cases} (j, i-1)\,, & i \neq 1 \\ (j-1, r_{x_{j-1}})\,, & i = 1 \end{cases}$$

denote the pair coming immediately before (j,i) in lexicographic order, for $(j,i) \neq (1,1)$.

Lemma 4.1.2 (Breaker's Technical Lemma)**.** *Let n be a large enough integer, let $\varepsilon \geqslant \frac{7\ln(\ln(n))}{\ln(n)}$ and let $t \in \mathbb{N}$ be any constant. For $p = n^{-2/3-\varepsilon}$ generate a random graph $G \sim G_{n,p}$. Then with probability at least $1 - n^{-\varepsilon/4}$ there exists a set $A = \{x_1, \dots, x_t\} \subset V(G)$ of size t, such that successively applying Algorithm 1 for $x_1, \dots, x_t$ the following holds for every $j \in [t]$ and $i \leqslant \tilde{r}_j := \min\{r_{x_j}, \lceil \frac{1}{\varepsilon} \rceil\}$:*

$(P1)$ $x_j \notin B^{a(j,1)} \cup N_G\big(B^{a(j,1)}\big)$,

$(P2)$ $|B_i^{x_j}| < n^{(1-i\varepsilon)/3}$,

$(P3)$ $e(B_i^{x_j}) = 0$,

$(P4)$ $B_i^{x_j} \cap \big(N_G\big(B^{a(j,1)}\big) \cup B^{a(j,1)}\big) = \varnothing$,

$(P5)$ *if we define* $N_{(j,i)}^s := \big\{v \in V\backslash B^{(j,i)} : d_G\big(v, B^{(j,i)}\big) \geqslant s\big\}$ *then*

$$|N_{(j,i)}^s| \leqslant \big(2j\varepsilon^{-1} + i\big)\, n^{(3-s(1+\varepsilon))/3} \quad \textit{for every } s \in \{0,1,2,3\},$$

$(P6)$ *for every* $k \in [t]$ *we have* $r_{x_k} = \tilde{r}_k$.

Before proving Lemma 4.1.2, let us first show how it implies Lemma 4.1.1.

Proof of Lemma 4.1.1. Apply Lemma 4.1.2 with $t = 7$. Then a.a.s. we find a set $A = \{x_1, \dots, x_t\}$ as promised by this lemma. Now, fix any set $M \subset V(G)$ of size 3. Since $|A| = 7$, it will be enough to verify the following two statements.

(i) Every vertex $x \in A$ satisfies $(B1)$–$(B3)$.

(ii) At most six vertices $x \in A$ do not satisfy $(B4)$.

For (i), consider any $x_j \in A$. Property $(B3)$ follows immediately from the halting condition of Algorithm 1. Moreover, Property $(B1)$ follows immediately by the definition of $B_1^{x_j}$ and Property $(P3)$. To see that Property $(B2)$ holds, let $v \in B_i^{x_j}$. The algorithm adds v to $B_i^{x_j}$ if $d_G\big(v, \bigcup_{k<i} B_k^{x_j}\big) \geqslant 2$. Moreover, we have $v \in V\backslash B^{a(j,i)}$, because of Property $(P4)$ and since $B_i^{x_j} \cap \big(\bigcup_{k<i} B_i^{x_j}\big) = \varnothing$ according to the algorithm. Now, using the Properties $(P5)$ and $(P6)$, and provided that n is large enough, we deduce that $|N_{a(j,i)}^3| < n^{-\varepsilon/2}$ and thus $v \notin N_{a(j,i)}^3 = \varnothing$. This yields $d_G\big(v, \bigcup_{k<i} B_k^{x_j}\big) \leqslant d_G\big(v, B^{a(j,i)}\big) \leqslant 2$. Finally, using that $e_G(B_i^{x_j}) = 0$ according to Property $(P3)$, we deduce that $d_G\big(v, \bigcup_{k\leqslant i} B_k^{x_j}\big) = 2$, proving $(B2)$.

Let us prove (ii) next. For any $k < j$, we have $B^{x_k} \subset B^{a(j,1)}$ by Definition (4.1.1) and since $B^{x_k} = \bigcup_{i\leqslant r_{x_k}} B_i^{x_k}$ by Algorithm 1. Thus, by using Property $(P4)$ we conclude that B^{x_j} and B^{x_k} are disjoint. Moreover, since $B^{x_k} \subset B^{a(j,i)}$ we also obtain that $N_G(B^{x_k}) \subset N_G(B^{a(j,i)})$. Thus, by using Property $(P4)$ once again, we get that G does not have any edges between B^{x_j} and B^{x_k}. As a consequence of this property we know that every vertex v that is adjacent to but not contained in B^{x_j} for some $j \in [7]$ needs to be an element of $V\backslash B^{(t,r_{x_t})}$. However, according to Property $(P5)$ and since $r_{x_j} \leqslant \lceil \frac{1}{\varepsilon} \rceil$ holds by Property $(P6)$, we obtain $N_{(t,r_{x_t})}^3 = \varnothing$ for large enough n. This implies that every vertex of $V\backslash B^{(t,r_{x_t})}$ is adjacent to at most two of the sets B^{x_j} with $j \in [7]$.

We conclude that at most three of the pairwise disjoint sets B^{x_j} may contain a vertex of M. If a vertex $v \in M$ belongs to some set B^{x_j} with $j \in [7]$, then $v \notin B^{x_k} \cup N_G(B^{x_k})$ for every $k \neq j$. If otherwise a vertex $v \in M$ belongs to $V\backslash B^{(t,r_{x_t})}$, then it is adjacent to at most two of the sets B^{x_j}. Hence, there are at most six vertices $x \in A$ such that $M \cap (B^x \cup N_G(B^x)) \neq \varnothing$. This proves statement (ii). □

Proof of Lemma 4.1.2. For the proof of Lemma 4.1.2 we expose the edges of $G \sim G_{n,p}$ step by step with respect to the given algorithm, and only during the process we choose the vertices of A randomly. To be more precise, we proceed as follows: We first choose x_1 uniformly at random from $V(G) = [n]$ and then apply Algorithm 1 for x_1. Once Algorithm 1 has been applied for x_{j-1} and $B^{x_{j-1}}$ is determined, we choose the next vertex x_j uniformly at random from $[n]$ and apply Algorithm 1 for x_j. While doing this, we always expose only those edges which have not been exposed yet and which are needed to determine the next set $B_i^{x_j}$ in the algorithm. For example, when we apply the algorithm for x_1, we first expose only the edges incident to x_1 so that we are able to determine $B_1^{x_1}$. Once this set is fixed, we expose all edges incident to $B_1^{x_1}$ that have not been exposed yet, so that we can find $B_2^{x_1}$. Then we expose all edges incident to $B_2^{x_1}$ that have not been exposed yet, and so on.

For the analysis of the algorithm, we consider the pairs (j,i), with $j \in [t]$ and $i \in [\tilde{r}_j]$, in lexicographic order. We consider the following event:

$E_{(j,i)}$: for all pairs until and including (j,i) the Properties $(P1)$–$(P5)$ hold, and the Property $(P6)$ is true for all $k < j$.

We will show that

$$\mathbb{P}\left(\overline{E_{(j,i)}}\middle|E_{a(j,i)}\right) < 5n^{-\varepsilon/3} \tag{4.1.2}$$

holds for every pair $E_{(j,i)}$, where $E_{a(1,1)}$ is the event which is always true. Before going into detail, let us first prove that Lemma 4.1.2 follows, once (4.1.2) is proven.

Claim 4.1.3. *If (4.1.2) holds, then* $\mathbb{P}\left(E_{(t,r_{x_t})} \text{ and } r_{x_t} = \tilde{r}_t\right) \geqslant 1 - n^{-\varepsilon/4}$.

Proof. Observe first that for every $j \in [t]$ the events $E_{(j,r_{x_j})}$ and $E_{(j,\tilde{r}_j)}$ are equivalent. Indeed, by definition $E_{(j,r_{x_j})}$ implies $E_{(j,\tilde{r}_j)}$, since $r_{x_j} \geqslant \tilde{r}_j$. Now, let $E_{(j,\tilde{r}_j)}$ be given and let us explain why $E_{(j,r_{x_j})}$ follows then. If we assume that the latter does not hold, then $\tilde{r}_j \neq r_{x_j}$, and by definition of $\tilde{r}_j$ we then have $\tilde{r}_j = \lceil \frac{1}{\varepsilon} \rceil < r_{x_j}$. Applying $(P2)$ for $(j,\tilde{r}_j)$, which is given under assumption of $E_{(j,\tilde{r}_j)}$, we obtain $B_{\tilde{r}_j}^{x_j} = \varnothing$. But this means that Algorithm 1, when processed for vertex x_j, must have stopped already, i.e. $r_{x_j} < \tilde{r}_j$, a contradiction.

Moreover, by looking at the above argument more carefully, we see that whenever one of the events $E_{(j,r_{x_j})}$ and $E_{(j,\tilde{r}_j)}$ holds, we must have $r_{x_j} = \tilde{r}_j \leqslant \lceil \frac{1}{\varepsilon} \rceil$.

For every $j \in [t]$ we now conclude that

$$\begin{aligned}\mathbb{P}\left(\overline{E_{(j,r_{x_j})}}\right) = \mathbb{P}\left(\overline{E_{(j,\tilde{r}_j)}}\right) &\leqslant \sum_{i=1}^{\tilde{r}_j} \mathbb{P}\left(\overline{E_{(j,i)}}\middle|E_{a(j,i)}\right) + \mathbb{P}\left(\overline{E_{a(j,1)}}\right) \\ &\overset{(4.1.2)}{\leqslant} \tilde{r}_j \cdot 5n^{-\varepsilon/3} + \mathbb{P}\left(\overline{E_{(j-1,r_{x_{j-1}})}}\right) < \frac{10}{\varepsilon}n^{-\varepsilon/3} + \mathbb{P}\left(\overline{E_{(j-1,r_{x_{j-1}})}}\right).\end{aligned}$$

Applying the above inequality recursively we finally obtain

$$\mathbb{P}\left(\overline{E_{(t,r_{x_t})} \text{ and } r_{x_t} = \tilde{r}_t}\right) = \mathbb{P}\left(\overline{E_{(t,r_{x_t})}}\right) < t \cdot \frac{10}{\varepsilon}n^{-\varepsilon/3} < n^{-\varepsilon/4}$$

as claimed. □

It thus remains to prove (4.1.2). We start with a few observations.

Observation 4.1.4. *If Algorithm 1 adds a vertex v to the set $B_i^{x_j}$, then $d_G\big(v, B_{i-1}^{x_j}\big) \geqslant 1$ and $d_G\big(v, \bigcup_{k\leqslant i-1} B_k^{x_j}\big) \geqslant 2$.*

Proof. Algorithm 1 adds a vertex v to $B_i^{x_j}$ if $d_G\big(v, \bigcup_{k\leqslant i-1} B_k^{x_j}\big) \geqslant 2$ and only if v was not already added to some $B_k^{x_j}$ with $k < i$. However, the latter ensures $d_G\big(v, \bigcup_{k\leqslant i-2} B_k^{x_j}\big) \leqslant 1$ and thus $d_G\big(v, B_{i-1}^{x_j}\big) \geqslant 1$. ☐

Observation 4.1.5. *If $E_{a(j,i)}$ holds, then the following is true:*

(i) $\big|\bigcup_{k\leqslant i-1} B_k^{x_j}\big| \leqslant |B^{a(j,i)}| < n^{1/3}$ *and* $\big|\bigcup_{k\leqslant i-1} N_G\big(B_k^{x_j}\big)\big| \leqslant \big|N_G\big(B^{a(j,i)}\big)\big| < n^{2/3-\varepsilon}$.

(ii) If Algorithm 1 adds a vertex v to $B_i^{x_j}$, then $v \in V\backslash B^{a(j,i)}$.

Proof. If $E_{a(j,i)}$ holds, we obtain

$$|B^{a(j,i)}| \overset{(4.1.1)}{\leqslant} \sum_{k<j}\sum_{\ell\leqslant r_k} |B_\ell^{x_k}| + \sum_{\ell<i} |B_\ell^{x_j}| \overset{(P2)}{\leqslant} \sum_{k<j}\sum_{\ell\leqslant r_k} n^{(1-\ell\varepsilon)/3} + \sum_{\ell<i} n^{(1-\ell\varepsilon)/3} < 2jn^{(1-\varepsilon)/3} < n^{1/3}$$

and

$$|B^{a(j,i)} \cup N_G(B^{a(j,i)})| \overset{(2.3.1)}{\leqslant} 2jn^{(1-\varepsilon)/3} + 2jn^{(1-\varepsilon)/3} \cdot 2n^{1/3-\varepsilon} < n^{2/3-\varepsilon},$$

provided that n is large enough, where the first inequality follows by the upper bound on the degree in G. Thus, (i) follows.

For (ii) observe that according to the algorithm no vertex from $\bigcup_{k\leqslant i-1} B_k^{x_j}$ can be added to $B_i^{x_j}$. Moreover, using Property $(P4)$, no vertex in $B^{a(j,1)}$ has a neighbour in $B_{i-1}^{x_j}$ (or in $\{x_j\}$ in the case when $i = 1$, because of $(P1)$), while every vertex being added to $B_i^{x_j}$ needs to have such a neighbour according to Observation 4.1.4 (or since $B_1^{x_j} = N_G(x_j)$ if $i = 1$). It thus follows that no vertex from $B^{a(j,i)} = B^{a(j,1)} \cup \bigcup_{k\leqslant i-1} B_k^{x_j}$ is added to $B_i^{x_j}$. ☐

Now we will prove (4.1.2) for each $j \in [t]$ and $i \in [\tilde{r}_j]$ by showing that under the condition of $E_{a(j,i)}$ each of the Properties $(P1)$–$(P5)$ in Lemma 4.1.2 fails to hold for the pair (j, i) with probability smaller than $n^{-\varepsilon/3}$. This is obviously enough to show (4.1.2) when $i > 1$. To get (4.1.2) for $i = 1$, recall that under the condition of $E_{a(j,1)} = E_{(j-1,r_{j-1})}$ we also have that $r_{x_{j-1}} = \tilde{r}_{j-1}$ (as shown in the proof of Claim 4.1.3), making sure that $(P6)$ holds for $(j, 1)$ as well. We discuss each of the Properties $(P1)$–$(P5)$ separately.

Property $(P1)$**:** The statement is trivially true for $j = 1$. So, let $j > 1$ and let us condition on $E_{a(j,1)}$. The vertex x_j is chosen uniformly at random from $[n]$ after Algorithm 1 has been applied for $x_1, \ldots, x_{j-1}$ and $B^{a(j,1)}$ was determined. Now, conditioned on $E_{a(j,1)}$, we have $|B^{a(j,1)} \cup N_G(B^{a(j,1)})| < n^{2/3}$ due to Observation 4.1.5. It thus follows that

$$\mathbb{P}\left((P1) \text{ fails for } j \;\middle|\; E_{a(j,1)}\right) < n^{-1/3} < n^{-\varepsilon/3}. \tag{4.1.3}$$

Property $(P2)$**:** Let $j \in [t]$. First we consider the case when $i = 1$. Then $B_1^{x_j} = N_G(x_j)$, and using Claim 2.3.4 we have

$$\mathbb{P}\left((P2) \text{ fails for } B_1^{x_j} \middle| E_{a(j,1)}\right) < \exp\left(-n^{1/3-2\varepsilon}\right) < n^{-\varepsilon}.$$

So, let $i > 1$ from now on and consider the moment immediately after $B_{i-1}^{x_j}$ was determined, i.e. when all remaining edges incident to $B_{i-1}^{x_j}$ become exposed in order to determine $B_i^{x_j}$.

When we condition on $E_{a(j,i)}$, only vertices from $v \in V \backslash B^{a(j,i)}$ can be added to $B_i^{x_j}$ according to Observation 4.1.5. Moreover, before $B_{i-1}^{x_j}$ was determined, for every vertex $v \in V \backslash B^{a(j,i)}$ all the edges towards $B_{i-1}^{x_j}$ have not been exposed so far. Now, if a vertex $v \in V \backslash B^{a(j,i)}$ is added to $B_i^{x_j}$ then by Observation 4.1.4 one of the following two cases needs to happen:

(i) $d_G\left(v, B_{i-1}^{x_j}\right) \geqslant 2$, or

(ii) $d_G\left(v, B_{i-1}^{x_j}\right) = 1$ and $d_G\left(v, \bigcup_{k<i-1} B_k^{x_j}\right) \geqslant 1$.

Conditioned on $E_{a(j,i)}$, the expected number of vertices in (i) is smaller than $n \cdot |B_{i-1}^{x_j}|^2 \cdot p^2 \overset{(P2)}{<} n \cdot n^{2(1-(i-1)\varepsilon)/3} \cdot n^{-4/3-2\varepsilon} = n^{1/3-2(i+2)\varepsilon/3}$. For (ii), observe that $d_G\left(v, \bigcup_{k<i-1} B_k^{x_j}\right) \geqslant 1$ means that $v \in \bigcup_{k<i-1} N_G\left(B_k^{x_j}\right)$. Since we have $\left|\bigcup_{k<i-1} N_G\left(B_k^{x_j}\right)\right| < n^{2/3-\varepsilon}$ according to Observation 4.1.5, we get that the expected number of vertices in $V \backslash B^{a(j,i)}$ satisfying (ii) is at most $n^{2/3-\varepsilon} \cdot |B_{i-1}^{x_j}| \cdot p \overset{(P2)}{<} n^{2/3-\varepsilon} \cdot n^{(1-(i-1)\varepsilon)/3} \cdot n^{-2/3-\varepsilon} = n^{1/3-(i+5)\varepsilon/3}$. Summing up, we get that the (conditional) expected size of $B_i^{x_j}$ is at most

$$n^{(1-2(i+2)\varepsilon)/3} + n^{(1-(i+5)\varepsilon)3} < n^{(1-(i+4)\varepsilon)/3}$$

and thus, using Markov's inequality (Lemma 2.3.5), we obtain

$$\mathbb{P}\left((P2) \text{ fails for } (j,i) \middle| E_{a(j,i)}\right) < \frac{n^{(1-(i+4)\varepsilon)/3}}{n^{(1-i\varepsilon)/3}} = n^{-4\varepsilon/3} < n^{-\varepsilon}. \tag{4.1.4}$$

Property $(P3)$**:** Again we condition on the event $E_{a(j,i)}$. By Observation 4.1.4 all of the vertices we add to $B_i^{x_j}$ need to come from $V \backslash B^{a(j,i)}$. Thus, when $B_i^{x_j}$ is determined, none of the edges in $E(B_i^{x_j})$ has been exposed yet. With probability at least $1 - n^{-4\varepsilon/3}$ we get $|B_i^{x_j}| < n^{(1-i\varepsilon)/3}$, according to (4.1.4). If we condition on the latter, the expectation of $e_G(B_i^{x_j})$ is smaller than $|B_i^{x_j}|^2 \cdot p \overset{(P2)}{<} n^{2(1-i\varepsilon)/3} \cdot n^{-2/3-\varepsilon} = n^{-(i+3)\varepsilon/3} \leqslant n^{-4\varepsilon/3}$. Thus, using Markov's inequality (Lemma 2.3.5), we obtain

$$\mathbb{P}\left((P3) \text{ fails for } (j,i) \middle| E_{a(j,i)}\right) \leqslant n^{-4\varepsilon/3} + \frac{n^{-4\varepsilon/3}}{1} < n^{-\varepsilon/3}.$$

Property $(P4)$: Let $i=1$. If $j=1$ then the statement is trivially true. Otherwise, we know from (4.1.3) that under the condition on the event $E_{a(j,i)}$ we have that $x_j \notin B^{a(j,1)} \cup N_G\big(B^{a(j,1)}\big)$ with probability at least $1-n^{-1/3}$. This implies $B_1^{x_j} \cap B^{a(j,1)} = \varnothing$ and thus it remains to show that it is unlikely that a vertex from $N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)}$ will be added to $B_1^{x_j}$. Note that before $B_1^{x_j}$ is determined, if $x_j \notin B^{a(j,1)} \cup N_G\big(B^{a(j,1)}\big)$ none of the edges between $N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)}$ and x_j has been exposed so far. Thus, using Observation 4.1.5, we (conditionally) expect at most $|N_G\big(B^{a(j,1)}\big)| \cdot p < n^{2/3-\varepsilon} \cdot n^{-2/3-\varepsilon} = n^{-2\varepsilon}$ vertices in $\big(N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)}\big) \cap B_1^{x_j}$. Using Markov's inequality (Lemma 2.3.5) again, it follows that

$$\mathbb{P}\left((P4) \text{ fails for } (j,1) \middle| E_{a(j,1)}\right) < n^{-1/3} + \frac{n^{-2\varepsilon}}{1} < n^{-\varepsilon}\,.$$

Let $i>1$ then. Under the assumption of $E_{a(j,i)}$, we have that $B_{i-1}^{x_j} \cap N_G\big(B^{a(j,1)}\big) = \varnothing$ according to $(P4)$. But then, according to Observation 4.1.4, no vertex from $B^{a(j,1)}$ is added to $B_i^{x_j}$, giving that $B_i^{x_j} \cap B^{a(j,1)} = \varnothing$. It thus remains to show that it is unlikely that a vertex from $N_G(B^{a(j,1)})\backslash B^{a(j,1)}$ will be added to $B_i^{x_j}$. Using that $B_{i-1}^{x_j} \cap B^{a(j,1)} = \varnothing$ by $(P4)$, we note that before $B_i^{x_j}$ is determined, none of the edges between $N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)}$ and $B_{i-1}^{x_j}$ has been exposed so far. Now (by applying Observation 4.1.4) a vertex $v \in N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)}$ is added to $B_i^{x_j}$ if

(i) $d_G\big(v, B_{i-1}^{x_j}\big) \geqslant 2$, or

(ii) $d_G\big(v, B_{i-1}^{x_j}\big) = 1$ and $d_G\big(v, \bigcup_{k<i-1} B_k^{x_j}\big) \geqslant 1$.

Hereby, again using Observation 4.1.5 as well as $(P2)$, the (conditional) expected number of vertices in (i) is at most $|N_G\big(B^{a(j,i)}\big)| \cdot |B_{i-1}^{x_j}|^2 \cdot p^2 < n^{2/3-\varepsilon} \cdot n^{2(1-(i-1)\varepsilon)/3} \cdot n^{-4/3-2\varepsilon} = n^{(2-2i)\varepsilon/3-3\varepsilon} < n^{-3\varepsilon}$. For (ii), observe that $N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)} \subset V\backslash B^{a(j,i)}$ holds since $\bigcup_{k<i} B_k^{x_j}$ and $N_G\big(B^{a(j,1)}\big)$ are disjoint due to Property $(P4)$. Therefore, if $d_G\big(v, \bigcup_{k<i-1} B_k^{x_j}\big) \geqslant 1$ and $v \in N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)}$, it holds that $v \in N^2_{a(j,i)}$. Using $(P5)$ and $(P6)$, we have that $|N^2_{a(j,i)}| < n^{1/3}$. Thus, the (conditional) expected number of vertices satisfying (ii) is bounded from above by $n^{1/3} \cdot |B_{i-1}^{x_j}| \cdot p \overset{(P2)}{<} n^{1/3} \cdot n^{(1-(i-1)\varepsilon)/3} \cdot n^{-2/3-\varepsilon} = n^{-(i+2)\varepsilon/3} \leqslant n^{-4\varepsilon/3}$. Summing up, we expect at most

$$n^{-3\varepsilon} + n^{-4\varepsilon/3} < n^{-\varepsilon}$$

vertices in $B_i^{x_j} \cap \big(N_G\big(B^{a(j,1)}\big)\backslash B^{a(j,1)}\big)$. By Markov's inequality (Lemma 2.3.5) we obtain

$$\mathbb{P}\left((P4) \text{ fails for } (j,i) \middle| E_{a(j,i)}\right) < \frac{n^{-\varepsilon}}{1} = n^{-\varepsilon}\,.$$

Property $(P5)$: Now, to prove $(P5)$, we first consider the case when $(j,i) = (1,1)$. The bound on $|N^0_{(1,1)}|$ is trivially true. So let $s \geqslant 1$. Immediately after $B^{(1,1)} = N_G(x_1)$ is determined, none of the edges between $V\backslash B^{(1,1)}$ and $B^{(1,1)}$ has been exposed yet. Moreover, according to Claim 2.3.4, with probability at least $1-\exp(-n^{1/3-2\varepsilon})$ we have $|B^{(1,1)}| < n^{(1-\varepsilon)/3}$. Thus, if we condition on that bound, the expected size of $N^s_{(1,1)}$ is at most $n \cdot \big(|B^{(1,1)}| \cdot p\big)^s < n^{(3-s(1+4\varepsilon))/3}$. We use Markov's inequality (Lemma 2.3.5) again to conclude that

$$\mathbb{P}\left((P5) \text{ fails for } (1,1)\right) \leqslant \sum_{s\in[3]} \frac{n^{(3-s(1+4\varepsilon))/3}}{(2\varepsilon^{-1}+1)\, n^{(3-s(1+\varepsilon))/3}} < n^{-\varepsilon/3}\,.$$

So let $(j,i) \neq (1,1)$ from now on. Again, the bound on $|N^0_{(j,i)}|$ is trivially true. Under condition of $E_{a(j,i)}$ we have $|B^{x_j}_i| < n^{(1-i\varepsilon)/3}$ with probability at least $1 - n^{-4\varepsilon/3}$, according to (4.1.4). From now on, we condition on this property to hold. Given that $E_{a(j,i)}$ holds, we additionally get

$$|N^s_{a(j,i)}| \overset{(P5)}{\leqslant} \begin{cases} \left(2j\varepsilon^{-1} + i - 1\right) n^{(3-s(1+\varepsilon))/3} & \text{if } i > 1 \\ \left(2(j-1)\varepsilon^{-1} + r_{x_{j-1}}\right) n^{(3-s(1+\varepsilon))/3} \overset{(P6)}{<} \left(2j\varepsilon^{-1} + i - 1\right) n^{(3-s(1+\varepsilon))/3} & \text{if } i = 1 \end{cases}$$

for every $s \in \{0,1,2,3\}$. Thus,

$$\begin{aligned} |N^s_{(j,i)}| &= |N^s_{(j,i)} \cap N^s_{a(j,i)}| + |N^s_{(j,i)} \backslash N^s_{a(j,i)}| \\ &\leqslant \left(2j\varepsilon^{-1} + i - 1\right) n^{(3-s(1+\varepsilon))/3} + |N^s_{(j,i)} \backslash N^s_{a(j,i)}| \,. \end{aligned} \tag{4.1.5}$$

Now, for $s \in [3]$, if a vertex v ends up being in $N^s_{(j,i)} \backslash N^s_{a(j,i)}$, then by definition we have that $v \in V \backslash B^{(j,i)} \subset V \backslash B^{a(j,i)}$ and $d_G\big(v, B^{a(j,i)}\big) = t$ for some $t < s$. But this means that $v \in N^t_{a(j,i)}$, and in order to be added to $N^s_{(j,i)}$ the vertex v needs to receive at least $s - t$ edges towards $B^{x_j}_i$ (which get exposed only after $B^{x_j}_i$ has been determined, since $v \in V \backslash B^{(j,i)}$). We conclude that the (conditional) expected size of $|N^s_{(j,i)} \backslash N^s_{a(j,i)}|$ is at most

$$\begin{aligned} \sum_{t<s} |N^t_{a(j,i)}| \cdot \left(|B^{x_j}_i| \cdot p\right)^{s-t} &\leqslant \left(2j\varepsilon^{-1} + i - 1\right) \sum_{t<s} n^{(3-t(1+\varepsilon))/3} \left(n^{(-1-(i+3)\varepsilon)/3}\right)^{s-t} \\ &= \left(2j\varepsilon^{-1} + i - 1\right) \sum_{t<s} n^{(3-s(1+\varepsilon)-(i+2)(s-t)\varepsilon)/3} \\ &\leqslant \left(2j\varepsilon^{-1} + i - 1\right) \sum_{t<s} n^{(3-s(1+\varepsilon)-3\varepsilon)/3} < n^{(3-s(1+\varepsilon)-2.5\varepsilon)/3}\,, \end{aligned}$$

where the first inequality uses $(P2)$ and $(P5)$, and the last inequality uses that $i \leqslant \tilde{r}_j \leqslant 2\varepsilon^{-1}$. Thus, with Markov's inequality (Lemma 2.3.5) and union bound, we obtain

$$\mathbb{P}\left(\exists s \in [3]: \; |N^s_{(j,i)} \backslash N^s_{a(j,i)}| > n^{(3-s(1+\varepsilon)-\varepsilon)/3}\right) < \frac{n^{(3-s(1+\varepsilon)-2.5\varepsilon)/3}}{n^{(3-s(1+\varepsilon)-\varepsilon)/3}} = n^{-1.5\varepsilon/3} = n^{-\varepsilon/2}\,.$$

Combining this with (4.1.5), we see that with (conditional) probability at least $1 - n^{-\varepsilon/2}$ we have

$$|N^s_{(j,i)}| \leqslant \left(2j\varepsilon^{-1} + i - 1\right) n^{(3-s(1+\varepsilon))/3} + n^{(3-s(1+\varepsilon)-\varepsilon)/3} < \left(2j\varepsilon^{-1} + i\right) n^{(3-s(1+\varepsilon))/3}$$

for every $s \in [3]$, and thus

$$\mathbb{P}\left((P5) \text{ fails for } (j,i) \big| E_{a(j,i)}\right) < n^{-4\varepsilon/3} + n^{-\varepsilon/2} < n^{-\varepsilon/3}\,.$$

This finishes the proof of Lemma 4.1.2. □

4.2 Walker-Breaker games: Walker's strategy

In this section we consider a $(2:2)$ Walker-Breaker Hamiltonicity game played on $G \sim G_{n,p}$ with $p = n^{-2/3+\varepsilon}$ and $\varepsilon \in (0,1)$. Our goal is to prove Theorem 1.3.22 by finding a strategy for Walker that she can follow to a.a.s. create a Hamilton cycle.

Instead of proving this theorem directly, we will prove an even stronger statement (Theorem 4.2.7, which we will state later), which quintessentially says that Walker can create a graph which behaves like a random graph with a few missing edges. From this result we will then deduce Theorem 1.3.22 with an application of Theorem 2.4.2.

The following subsections contain all of the helpful tools which Walker has available. The arguably most useful ones are so-called *good structures* $\mathcal{S}_k$ for Walker, which she can use to reach any vertex of G within a constant number of turns. We will give a formal definition of those structures in Subsection 4.2.1, but before doing so let us first present a short overview of how Waiter's overall strategy looks like.

Walker's strategy consists of three sub-goals, which can roughly be summed up as follows: She wants to create a large star at her starting vertex, she wants to connect new vertices to her graph, and she wants to turn her graph into a graph which behaves almost like a random graph. She will always play a few turns to get closer to one of her sub-goals, and then she will move on to the next sub-goal. More details will be given later, but let us briefly say why she wants to achieve those sub-goals and where the good structures come into play.

For Walker it will be necessary to build a large star, because she needs to reach the starting vertices of good structures whenever necessary for her strategy. She has to claim this star early in the game, since then she has these edges available, even later on in the game.

These good structures are then used by Walker to reach any vertex she wants in the graph. Therefore, we will show that if $p \geqslant n^{-2/3+\varepsilon}$ Walker can in fact find so many of these good structures in $G_{n,p}$ that for each vertex, and even after Breaker has claimed many edges incident to many of those good structures, there will still be sufficiently many of those structures available for Walker, which contain no Breaker edges at all. Waiter is then able to use these structures to connect new vertices (which she chooses in a specific way) to her current graph.

Lastly, Walker will turn her graph into a graph which behaves similar to a random graph, by finding exposure vertices (which are again chosen in a specific way), walking to the exposure vertices, and then claiming random edges incident to those vertices.

With an overview of Walker's strategy at hand, let us next define those good structures.

4.2.1 Good Structures $\mathcal{S}_k$

During the game Walker is often confronted with the following situation: her current position is some vertex $a \in V(G)$, and by using some potential function argument she decides for a vertex $x \in V(G)$ that she wants to reach next. In order to guarantee that Walker can indeed reach this vertex, we make use of copies of a *good structure* $\mathcal{S}_k$, which we will finally define formally.

Definition 4.2.1 (Good structures)**.** Given any positive integer k, let $\mathcal{T}_k$ denote a perfect 3-ary tree $\mathcal{T}_k$ of depth k. Starting from $\mathcal{T}_k$, the *good structure* $\mathcal{S}_k$ is created as follows: we subdivide every edge of $\mathcal{T}_k$ with one vertex, and afterwards unify all of its leaves into a single vertex.

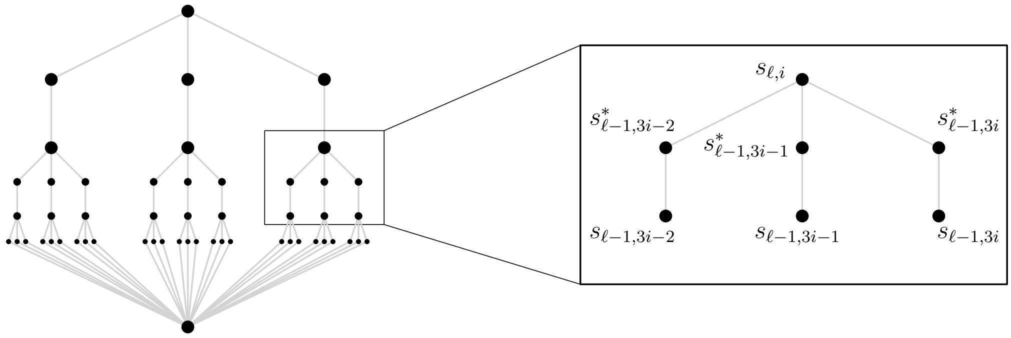

Figure 4.2.1: The left part of the picture shows the structure $\mathcal{S}_3$, while the right part depicts the notation of vertices in a small subtree between main levels ℓ and $\ell - 1$.

The left side of Figure 4.2.1 shows $\mathcal{S}_3$. Whenever we want to refer to the graphs $\mathcal{T}_k$ and $\mathcal{S}_k$, we make use of the following labelling. For the graph $\mathcal{T}_k$, the root is denoted by $s_{k,1}$, and for every $\ell \in [k]$, and $i \in [3^{k-\ell}]$, we let the three children of $s_{\ell,i}$ be $s_{\ell-1,3i-2}$, $s_{\ell-1,3i-1}$, and $s_{\ell-1,3i}$. For $\mathcal{S}_k$ we keep this labelling, and we set $s_0 := s_{0,i}$ for every $i \in [3^k]$ to be the identifying vertex. When talking about the subdividing vertices, we write $s^*_{\ell-1,3i-j}$ for the middle vertex of the path between vertices $s_{\ell,i}$ and $s_{\ell-1,3i-j}$, for every $\ell \in [k]$, $i \in [3^{k-\ell}]$, and $j \in \{0, 1, 2\}$. See also the right side of Figure 4.2.1.

Moreover, in light of the strategy described in Lemma 4.2.3, we sometimes say that $\mathcal{S}_k$ *starts* in $s_{k,1}$ and *ends* in s_0. Furthermore, for every $\ell \in [k]$ we call $L_\ell := \{s_{\ell,i} : i \in [3^{k-\ell}]\}$ the *main level* ℓ, and we use $I(k) = \{(\ell, i) : \ \ell \in [k], i \in [3^{k-\ell}]\}$ for the set of indices over all vertices belonging to main levels. Similarly, for every $\ell \in \{0\} \cup [k-1]$ we call $L^*_\ell := \{s^*_{\ell,i} : i \in [3^{k-\ell}]\}$ the *secondary level* ℓ, and we use $I^*(k) = \{(\ell, i) : \ \ell \in [k] \cup \{0\}, i \in [3^{k-\ell}]\}$ for the set of indices over all vertices belonging to secondary levels.

As a first property of these good structures, let us calculate their number of vertices and edges. We claim that the following holds.

Claim 4.2.2. *For every positive integer k we have $v(\mathcal{S}_k) = 2 \cdot 3^k - 1$, $e(\mathcal{S}_k) = 3^{k+1} - 3$, and the tree $\mathcal{S}_k - s_0$ has 3^k leaves.*

Proof. By construction $\mathcal{S}_k - s_0$ is a tree with $|L^*_0| = 3^k$ leaves and with

$$v(\mathcal{S}_k - s_0) = \sum_{\ell=1}^{k} |L_\ell| + \sum_{\ell=0}^{k-1} |L^*_\ell| = \sum_{\ell=1}^{k} 3^{k-\ell} + \sum_{\ell=0}^{k-1} 3^{k-\ell} = 2 \cdot 3^k - 2\,,$$

where the last equality can be proven by a simple induction on k. In particular, $v(\mathcal{S}_k) = 2 \cdot 3^k - 1$. The edge set of $\mathcal{S}_k$ consists of all edges of the tree $\mathcal{S}_k - s_0$ and all edges between s_0 and L^*_0. Hence, $e(\mathcal{S}_k) = v(\mathcal{S}_k - s_0) - 1 + |L^*_0| = 2 \cdot 3^k - 2 - 1 + 3^k = 3^{k+1} - 3$. □

The next lemma indicates how Walker can use copies of $\mathcal{S}_k$ in her strategy.

Lemma 4.2.3. *Let $k \geqslant 1$, and consider a $(2 : 2)$ Walker-Breaker game on the structure $\mathcal{S}_k$ with Breaker being the first player and Walker's starting position being $s_{k,1}$. Then Walker has a strategy $S_{structure}$ to reach the vertex s_0 within k rounds.*

Proof. We prove this lemma by induction on k. For $k = 1$, the structure $\mathcal{S}_k$ consists of three edge-disjoint paths of length 2 between $s_{1,1}$ and s_0, and by assumption $s_{1,1}$ is Walker's starting position. Since Breaker can only block two of these paths within the first round, Walker can find a free path with which she reaches s_0 within her first move.

For $k > 1$ consider the structure $\mathcal{S}_k$ starting in $s_{k,1}$ and ending in s_0. This structure consists of three copies of the structure $\mathcal{S}_{k-1}$ starting in one of the vertices $s_{k-1,i}$, $i \in [3]$, and ending in s_0, and the three paths of the form $s_{k,1}s^*_{k-1,i}s_{k-1,i}$. After Breaker's first move, at least one of the paths $s_{k,1}s^*_{k-1,i}s_{k-1,i}$ as well as the copy of $\mathcal{S}_{k-1}$ starting from $s_{k-1,i}$ still do not contain any edge claimed by Breaker. Walker claims this path and thus reaches $s_{k-1,i}$ within one move. According to our induction hypothesis Walker can then reach s_0 starting from $s_{k-1,i}$ within $k-1$ further moves and thus has a strategy to reach s_0 within k moves. □

4.2.2 Finding copies of $\mathcal{S}_k$ in $G_{n,p}$

By Claim 4.2.2 the density of $\mathcal{S}_k$ is slightly below 3/2, and hence for $p \geqslant n^{-2/3}$ we know that a random graph $G \sim G_{n,p}$ a.a.s. contains copies of $\mathcal{S}_k$ (see e.g. [11] for subgraph containment in $G_{n,p}$). Moreover, increasing our probability slightly to $p \geqslant n^{-2/3+\varepsilon}$ and having that $k = k(\varepsilon)$ is a sufficiently large constant, it can be verified that in expectation for any two vertices $v, x \in V(G)$ there is a copy of $\mathcal{S}_k$ such that v is the copy of $s_{k,1}$ and x is the copy of s_0. Now, throughout the game, Walker aims to find such copies which additionally are free of Breaker's edges and which can be reached from her current position a within one round. In order to guarantee that Walker can indeed find such copies, we want to make sure that there exists a collection of many copies of $\mathcal{S}_k$ with the additional property that any Breaker edge can only belong to a comparatively small number of these copies. This property will help later to ensure that Walker can play without facing a situation in which she wants to add a new vertex x to her graph but cannot find a good structure with which she can still reach x.

In order to prove the existence of the desired collections of structures we use a recursive approach. Because we want to keep independence whenever needed in the probabilistic analysis, we initially split the vertex set of $G \sim G_{n,p}$ into several blocks $B_t(s)$, for every $s \in V(\mathcal{S}_k - s_0)$ and $t \in [2]$, and some left-over R; then we aim for copies of $\mathcal{S}_k$ for which the copy of any vertex $s \in V(\mathcal{S}_k - s_0)$ is an element of one of the corresponding blocks $B_t(s)$. In order to make our statements precise, we use the following definitions.

Definition 4.2.4. Let a positive integer k, a graph G, and a family $\mathcal{B} := \{B(s) : \ s \in V(\mathcal{S}_k - s_0)\}$ of pairwise disjoint subsets of $V(G)$ be given. Furthermore, let $x \in V(G)$, $e \in E(G)$, $v \in B(s_{k,1})$, and $\ell \in [k]$. Then we define the following:

- A copy S of $\mathcal{S}_k$ in G is called a $(\mathcal{B}, x)$-*structure* if x is the copy of s_0 in S and if for every $s \in V(\mathcal{S}_k - s_0)$ there is a vertex in $B(s)$ which is the copy of s in S.
- We say that e *sees the vertex* x with respect to $\mathcal{B}$ if there exists a $(\mathcal{B}, x)$-structure containing e.

- We say that e *is relevant for* v with respect to $(\mathcal{B}, x)$ if there exists a $(\mathcal{B}, x)$-structure containing both v and e.

- We say that e *appears between levels* $\ell - 1$ *and* ℓ with respect to $\mathcal{B}$ if there exists a vertex $s \in L^*_{\ell-1}$ such that e is incident with a vertex in $B(s)$, and if ℓ is the smallest integer with this property.

- We say that e *appears below level* ℓ with respect to $\mathcal{B}$ if for some $\ell' \leqslant \ell$ the edge e appears between levels $\ell' - 1$ and ℓ'.

Now, we can state our main technical lemma, which the strategy of Walker builds on. In order to avoid rounding signs in its proof we restrict the possible choices of ε. More precisely, we let $\mathcal{R}$ be the set of real numbers $\varepsilon \in (0, 1)$ such that $\log_3(2\varepsilon^{-1} + 12) - 2$ is a positive integer. Note that, by monotonicity and since $\inf(\mathcal{R}) = 0$, it is enough to prove Theorem 1.3.22 for $\varepsilon \in \mathcal{R}$.

Lemma 4.2.5 (Walker's Technical Lemma)**.** *For every $\varepsilon \in \mathcal{R}$ there exists a positive integer k such that the following holds. Let the structure $\mathcal{S}_k$ be given with the labelling from Subsection 4.2.1. Further, let $p \geqslant n^{-2/3+\varepsilon}$ and $G \sim G_{n,p}$. Then a.a.s. there exist a vertex $a \in V(G)$, a partition $V(G)\backslash\{a\} = V_1 \cup V_2$, families $\mathcal{B}_t = \{B_t(s) : s \in V(\mathcal{S}_k - s_0)\}$ of pairwise disjoint subsets of V_t each of size $\frac{n}{3^{k+10}}$ for every $t \in [2]$, and a set $R \subseteq V\backslash(\bigcup_{s \in V(\mathcal{S}_k - s_0)}(B_1(s) \cup B_2(s)) \cup \{a\})$ such that the following properties hold:*

(S) <u>*Sizes:*</u> *$|R| \geqslant \frac{n}{2}$, $|B_t(s)| = \frac{n}{3^{k+10}}$ for every $s \in V(\mathcal{S}_k - s_0)$ and $t \in [2]$, and $|N(a, R)| \geqslant n^{1/3+0.5\varepsilon}$.*

(C) <u>*Candidate sets:*</u> *Let $t \in [2]$. For all $x \in V_{3-t}$ there exist candidate sets $C^x(s) \subseteq B_t(s)$ for every $s \in V(\mathcal{S}_k - s_0)$ such that:*

  (C1) <u>*Number of candidates:*</u> *$|C^x(s)| = n^{(3^{\ell+1}-3)\varepsilon} \ln^{\pm 3^{3\ell}}(n)$ for every $\ell \in [k]$ and $s \in L_\ell$.*

  (C2) <u>*Neighbourhoods for main levels:*</u> *For every $(\ell, i) \in I(k)$, every vertex $v \in C^x(s_{\ell,i})$ has a neighbour in each of the sets $C^x(s^*_{\ell-1,j})$ with $3i - 2 \leqslant j \leqslant 3i$.*

  (C3) <u>*Neighbourhoods for secondary levels:*</u> *For every $(\ell, i) \in I^*(k)$, every vertex $v \in C^x(s^*_{\ell,i})$ has a neighbour in $C^x(s_{\ell,i})$, where we set $C^x(s_{0,i}) = \{x\}$.*

  Set $C^x := C^x(s_{k,1})$ from now on.

(E) <u>*Edge appearances:*</u> *Let $t \in [2]$. For every edge $e \in E(G)$ the following holds:*

  (E1) If e appears below level $k - 1$ with respect to $\mathcal{B}_t$, then e sees at most $\ln^2(n)$ vertices $x \in V_{3-t}$ with respect to $\mathcal{B}_t$.

  (E2) If e appears between levels $k - 1$ and k with respect to $\mathcal{B}_t$, then e sees at most $n^{1/3+0.1\varepsilon}$ vertices $x \in V_{3-t}$ with respect to $\mathcal{B}_t$.

(R) <u>Relevance of edges:</u> *Let $t \in [2]$ and $x \in V_{3-t}$. For any set Z_1 of edges appearing below level $k-1$ with respect to $\mathcal{B}_t$ and any set Z_2 of edges appearing between levels $k-1$ and k with respect to $\mathcal{B}_t$, let $C^x[Z_1, Z_2]$ denote the vertices in C^x for which no edge of $Z_1 \cup Z_2$ is relevant with respect to $(\mathcal{B}_t, x)$. Then the following holds: If $|Z_1| = \ln^4(n)$ and $|Z_2| = n^{1/3+0.5\varepsilon}$, and if $A \subseteq N(a, R)$ has size $|A| = n^{1/3}$, then $e_G\big(A, C^x[Z_1, Z_2]\big) \geqslant n^{1/3+1.5\varepsilon}$.*

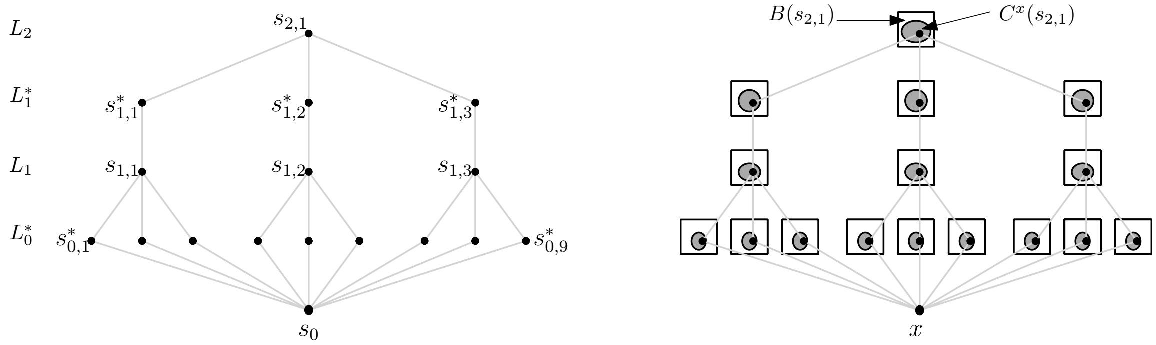

Figure 4.2.2: The left picture shows the structure $\mathcal{S}_2$. The right picture shows a $(\mathcal{B}, x)$-structure with respect to blocks $B(s)$ depicted by rectangles, and candidate sets $C^x(s)$ depicted by grey areas.

The proof of Lemma 4.2.5 follows from standard applications of probabilistic tools. Details of the full proof can be found in Subsection 4.2.4. For now, let us briefly give a reason why we care about the above properties. Set $B(s) = B_1(s) \cup B_2(s)$ for every $s \in V(\mathcal{S}_k) - s_0$ and $\mathcal{B} = \{B(s) : \ s \in V(\mathcal{S}_k) - s_0\}$. Property (C) promises that for every $x \in V(G)\backslash\{a\}$ we can find a collection of $(\mathcal{B}, x)$-structures starting in any vertex of $C^x(s_{k,1})$.

Observation 4.2.6. *Let $a \in V(G)$, a partition $V(G)\backslash\{a\} = V_1 \cup V_2$, families $\mathcal{B}_t$, and a set R be given according to Lemma 4.2.5 such that property (C) holds. Then for every $t \in [2]$, $x \in V_{3-t}$, and $v \in C^x(s_{k,1}) \cap B_t(s_{k,1})$ there exists a $(\mathcal{B}_t, x)$-structure starting in v and ending in x.*

Let $x \in V_{3-t}$. Then, by fixing any vertex $v_{k,1} := v \in C^x(s_{k,1}) \cap B_t(s_{k,1})$, we can in fact find a copy of $\mathcal{S}_k$ starting in $v_{k,1}$ and ending in x as follows: By (C2) we know that $v_{k,1}$ has neighbours in each of the sets $C^x(s^*_{k-1,i}) \subset B_t(s^*_{k-1,i})$. Picking one such vertex $v^*_{k-1,i}$ from each of these sets, we have fixed copies of the vertices $s^*_{k-1,i}$ of $\mathcal{S}_k$. Then, by (C3) we know that each $v^*_{k-1,i}$ has a neighbour in $C^x(s_{k-1,i}) \subset B_t(s_{k-1,i})$; hence we can pick copies of the vertices $s_{k-1,i}$. This process continues, always switching between (C2) and (C3) until we reach $C^x(s_{0,i}) = \{x\}$, so that x becomes the copy of s_0 and a $(\mathcal{B}_t, x)$-structure is found.

Note that (C1) ensures that we have many possible candidates in $C^x(s_{k,1})$ to start with, which in turn means that we have many such structures. Additionally, property (E) helps when we want to show that in each move, Breaker can only block a reasonably small number of $(\mathcal{B}, x)$-structures. Moreover, property (R) comes in handy when we want to ensure that Walker, being at her starting position a, can always reach some $(\mathcal{B}, x)$-structure with no Breaker edges within one round. More precisely, the sets Z_1 and Z_2 mentioned in the property represent edges claimed by Breaker, and the set $C^x[Z_1, Z_2]$ represents those candidates in C^x from which Walker can still reach the vertex x by following the strategy from Lemma 4.2.3. Once we know that $e_G\big(A, C^x[Z_1, Z_2]\big)$ is large, we can ensure that Walker has enough options to reach one of the mentioned candidates within one round. More details are given in Subsection 4.2.3 where we first exhibit a partially randomized strategy for Walker and then prove that a.a.s. if she follows that strategy she wins the game.

4.2.3 The strategy for Walker

The overall goal of this subsection is to prove Theorem 1.3.22. As already mentioned before, instead of proving this theorem directly, we will first prove the more general Theorem 4.2.7, which states that Walker can even claim a graph with a (p, ε)-resilient graph property.

Theorem 4.2.7. *Let $\varepsilon \in (0,1)$, $p \geqslant n^{-2/3+\varepsilon}$, and $\mathcal{P} = \mathcal{P}_n$ be a monotone increasing graph property that is (p, ε)-resilient. Then, playing a $(2:2)$ Walker-Breaker game on the edges of a random graph $G \sim G_{n,p}$, Walker a.a.s. has a strategy to occupy a graph with property $\mathcal{P}$.*

Due to monotonicity, we can restrict our attention to $\varepsilon \in \mathcal{R}$, meaning that $\log_3(2\varepsilon^{-1} + 12) - 2$ is a positive integer. Hence we can apply Lemma 4.2.5 to obtain some output $k \in \mathbb{N}$ (we will later see that $k = \log_3(2\varepsilon^{-1} + 12) - 2$). Thus, for the graph $G \sim G_{n,p}$, we can always condition on the properties promised by Lemma 4.2.5 and Claim 2.3.3. That is, before the game starts we fix a vertex $a \in V(G)$, a set R, and families $\mathcal{B}_t$ with $t \in [2]$ as described in the lemma. In particular, we assume that all of the properties (S), (C), (E), and (R) hold, which (amongst other things) provides us with families of $(\mathcal{B}_t, x)$-structures as mentioned in Observation 4.2.6. Moreover, for notational reasons, we set $B(s) = B_1(s) \cup B_2(s)$ for every $s \in V(\mathcal{S}_k) - s_0$ and $\mathcal{B} = \{B(s) : s \in V(\mathcal{S}_k - s_0)\}$. Whenever necessary, we assume that n is large enough.

In the following we will describe Walker's strategy, which combines a deterministic strategy with randomized moves. We will first describe the overall idea, introduce necessary notation, and give a substrategy that is used later on. Afterwards, we will explain the full strategy of Walker. Finally, in the strategy discussion we will show that Walker can always follow the described strategy and that, against any strategy of Breaker, Walker a.a.s. manages to occupy a graph with property $\mathcal{P}$. It thus follows that Breaker cannot have a winning strategy, which always prevents Walker from occupying a graph with property $\mathcal{P}$, and hence Walker must have a deterministic strategy to win the game (see Zermelo's Lemma, e.g. [6]).

Our setup

The overall idea of Walker's strategy in order to create a spanning graph is as follows: as long as Walker's graph is not spanning, she consecutively chooses a vertex x and finds a free copy of a good structure $\mathcal{S}_k$ to reach x.[1] Because of Walker's movement restrictions, throughout the game we want to make sure that the following holds.

[1] This is a similar approach to the one that was used in the $(2:2)$ Connector-Breaker connectivity game for Connector's strategy in [21]. However, the more restricted movement possibilities for Walker require several

1. We make sure that Walker's graph keeps having a small diameter.
2. We play a substrategy (Lemma 4.2.8) to make sure that Walker is always able to reach starting points of appropriate copies of $\mathcal{S}_k$.
3. We apply a weight function argument to have control on the number of structures that Breaker blocks.

Additionally, in order to obtain Theorem 4.2.7, we combine the above approach with a randomized strategy, for which we can almost directly reuse the analysis from [33], because the diameter of Walker's graph is kept small.

In the strategy (which we will make more precise later on) Walker alternates between three different sequences of moves, denoted by Sequences I – III. These different sequences allow us to maintain three different goals:

(I) occupy suitable paths of length 2 starting at the starting vertex a,

(II) ensure that for every $x \notin V(W)$ there exist $(\mathcal{B}, x)$-structures the edges of which are available,

(III) occupy a graph which "behaves almost like a random graph" (details below).

Let us remark at this point that achieving the goals (I) and (II) would already be sufficient to prove that Walker wins the connectivity game. As long as she maintains these goals, the main idea of her strategy is the following: Let x be any vertex that Walker wants to add next to her component. By (II) she can find some $(\mathcal{B}, x)$-structure S_x, the edges of which are available, and by (I) there is a path of length 2 with which Walker can reach the top vertex of S_x. Once this top vertex is reached, Walker can use the edges of S_x to reach x in a constant number of turns (Lemma 4.2.3) and hence she can add it to her component.

Property (I) will be guaranteed by Sequence I of Walker's strategy. Following this strategy, Walker makes sure to create a large star with centre a, and moreover she makes sure to claim suitable edges starting at the leaves of this star, hence creating paths of length 2. These paths are created in such a way that Walker can always reach vertices from certain candidate sets $C^x[Z_1, Z_2]$ as described in (R). More details about this are given by Lemma 4.2.8.

Property (II) will be guaranteed by Sequence II of Walker's strategy. Roughly speaking, the idea is as follows: For every $x \in V(G)$, having $t \in [2]$ such that $x \in V_{3-t}$, we can consider the set E_x consisting of all edges which see the vertex x with respect to $\mathcal{B}_t$. Using the Continuous Box game, Walker then ensures that Breaker does not claim too many elements of E_x as long as $x \notin V(W)$. Hence, we are able to find $(\mathcal{B}_t, x)$-structures which have not been blocked by Breaker so far.

changes. Both the local strategy on a copy of the specific structure $\mathcal{T}_k$ in [21] and finding a starting point of a free copy of $\mathcal{T}_k$, used the fact that at anytime during the game Connector could claim edges incident to any vertex she previously reached. As Walker does not have this option available, we need a different structure $\mathcal{S}_k$ and have to be more careful in order to find appropriate copies of $\mathcal{S}_k$ throughout the game.

Finally, property (III) will be guaranteed by Sequence III of Walker's strategy. Using that part of the strategy, Walker generates a random graph $H \sim G_{\ln^{-1}(n)}$ while the game is in progress, i.e. $H \sim G_{n,q}$ with $q = p\ln^{-1}(n) = \frac{n^{-2/3+\varepsilon}}{\ln(n)} = \omega(n^{-2/3})$. Using a partially randomized strategy, Walker then ensures that a.a.s.

$$d_{W\cap H}(v) \geqslant (1-\varepsilon)d_H(v) \tag{4.2.1}$$

holds for every $v \in V(H)$ by the end of the game. Since the desired property $\mathcal{P}$ is (p,ε)-resilient, we then have that $W \cap H$ must have property $\mathcal{P}$ a.a.s. and thus Walker wins. This part of the strategy is motivated by [33] and has similarly been used for Walker-Breaker games in [23, 36].

Let us mention here that the final step to deduce Theorem 1.3.22 is to look at Theorem 2.4.2. Since the property that a random graph contains a Hamilton cycle is $(p, \frac{1}{2}-\varepsilon)$-resilient if our probability is large enough (which is the case), the graph Walker claimed during the game will a.a.s. contain a Hamilton cycle.

In the following we provide a few more details of that strategy adapted to our setting.

While the game is in progress, Walker tosses a coin on every edge of G independently at random, where the probability of success equals $\ln^{-1}(n)$. If the coin toss for an edge e shows a success, Walker adds the edge e to the random graph H, and moreover, Walker claims the edge if that is possible.

In order to decide on which edge Walker tosses a coin, she always identifies some exposure vertex v (to be defined later in the strategy description) and makes sure that she can reach the vertex v within a small number of rounds. In order to choose an appropriate exposure vertex, Walker plays an auxiliary MinBox game in parallel, namely MinBox$(n, 4pn, 0.5\ln^{-1}(n), 16k+28)$. In this simulated $(1 : 16k+28)$ Maker-Breaker game, in which Walker imagines playing as Maker, we have a box J_v of size $4pn$ for every vertex $v \in V(G)$.

Then, once an exposure vertex v is identified and Walker has reached this vertex, she starts to toss a coin on edges which are incident to v, but only for the ones for which she has not tossed a coin yet, and she stops when the first success happens, and claims that edge (if possible). Of course, it may happen that none of her coin tosses is a success, and in this case we say that her move is a failure of type I, and Walker just makes an arbitrary move instead. It may also happen that Walker's coin toss is a success on an edge which was already claimed by Breaker in an earlier round so that Walker cannot claim it. Similarly to the previous case, Walker instead makes an arbitrary move and we denote it as failure of type II.

Analogously to [33] our final goal is to prove that a.a.s. at every vertex v only a relatively small number of failures happen, which then yields (4.2.1).

For the analysis of such an argument, we say that Walker exposes an edge $e \in E(G)$ if she tosses a coin on it, and we set $U_v \subseteq N_G(v)$ to be the set of all neighbours w of v for which the edge vw has not been exposed yet. Moreover, we introduce counters $f_I(v)$ and $f_{II}(v)$ in order to keep track on the number of failures of type I and type II that involve edges incident with v. Initially, we set $f_I(v) = f_{II}(v) = 0$ for every vertex $v \in V(G)$. In order to ensure that the number of failures of type II does not become too large, we apply Theorem 2.2.5 for

MinBox$(n, 4pn, 0.5\ln^{-1}(n), 16k+28)$. Here $w_B(J_v)$ is made to be related to $d_B(v)$ for every $v \in V(G)$, while $w_M(J_v)$ is made to be related to the number of edges incident to v on which the coin toss was successful. Initially, $w_B(J_v) = w_M(J_v) = 0$ for every $v \in V(G)$. Later we will realize that it is very likely that Walker is done with all coin tosses on the edges incident to a fixed vertex v before Breaker has claimed too many of these edges. This in turn helps to bound the number of failures of type II.

A substrategy (Sequence I)

In this segment we prepare for Sequence I of our main strategy and prove the following lemma.

Lemma 4.2.8. *Let n, b be positive integers and let n be large enough. Assume that a $(2:b)$ Walker-Breaker game on some graph G is in progress and that Walker has already claimed the edges of a star of size $n^{1/3}$ with centre a and A being the set of leaves. Assume further that there exist (not necessarily disjoint) subsets $C_1, \dots, C_s \subseteq V(G) \backslash (A \cup \{a\})$, with $s \leqslant \exp(n^{1/3+\varepsilon})$, such that for each $i \in [s]$ there exist at least $n^{1/3+1.1\varepsilon}$ available edges between A and C_i. Then Walker has a strategy $\mathcal{S}_{paths}$ that satisfies the following:*

(i) The strategy proceeds in sequences of two moves, which always start and end in the vertex a, and in which Walker claims exactly one edge from $E_G(A, \bigcup_{i\in[s]} C_i)$.

(ii) The strategy ensures that Walker claims at least one element from each of the sets $E_G(A, C_i)$. In particular, as long as Walker plays according to this strategy, there must always be an edge in $E_G(A, C_i)$ that is either free or taken by Walker.

Proof. Assume that at any moment in the game Walker's position is the vertex a and she wants to claim some free edge $e = xy \in E_G(A, \bigcup_{i\in[s]} C_i)$ with $x \in A$. Since, by assumption of the lemma, Walker has already claimed the edge ax, she can play a sequence of two moves as follows: first go from a to y via x, and secondly return to a by using the same edges. If Walker continues playing like this, her strategy already satisfies (i). Moreover, since Walker can claim one free edge of $E_G(A, \bigcup_{i\in[s]} C_i)$ arbitrarily while Breaker claims at most $2b$ edges in the meantime, Walker can imagine playing an auxiliary $(2b:1)$ Maker-Breaker game on the board $E_G(A, \bigcup_{i\in[s]} C_i)$ with the family of winning sets being $\mathcal{F} = \{E(A, C_i) : i \in [s]\}$, but taking over the role of Breaker since for (ii) she wants to occupy an edge in each of the winning sets. In order to show that Walker can succeed in doing so, we can apply Beck's winning criterion (Theorem 2.2.1):

$$\sum_{F\in\mathcal{F}} 2^{-|F|/(2b)} = \sum_{i\in[s]} 2^{-(e_G(A,C_i))/(2b)} \leqslant e^{n^{1/3+\varepsilon}} \cdot 2^{-(n^{1/3+1.1\varepsilon})/(2b)} = o(1)\,.$$

This proves the lemma. □

Strategy description

Let us now provide more details for Walker's strategy. Throughout the game, Walker alternates between the following sequences of moves which always start and end in the vertex a. To keep track of her star with centre a (Sequence I), we let N_a denote a set of vertices $w \in N(a, R)$ for which aw is already a Walker's edge. Initially $N_a = \varnothing$. Moreover, once $|N_a| = n^{1/3}$ holds, the set is never updated again (even if Walker claims another edge incident with a).

In order to apply the Continuous Box game (Sequence II), we define weights for the edges and vertices of G as follows. For every edge $e \in E(G)$ and $t \in [2]$ we define

$$\text{weight}_t(e) := \begin{cases} \ln^{-2}(n) & \text{if } e \text{ is an edge below level } k-1 \text{ w.r.t. } \mathcal{B}_t \\ n^{-1/3-0.1\varepsilon} & \text{if } e \text{ is an edge between level } k-1 \text{ and } k \text{ w.r.t. } \mathcal{B}_t, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, for every $x \in V(G)\backslash\{a\}$ and $t \in [2]$ such that $x \in V_{3-t}$, we dynamically define

$$\text{weight}(x) := \sum_{\substack{e \in B \text{ sees } x \\ \text{w.r.t. } \mathcal{B}_t}} \text{weight}_t(e) ,$$

where the sum is taken over all edges e that see x with respect to $\mathcal{B}_t$ and which have been claimed by Breaker already. Initially weight$(x) = 0$ for every $x \in V(G)\backslash\{a\}$.

Finally, in light of property (R) from Lemma 4.2.5, we let $C^x[Z_1, Z_2]$ be the set of vertices in C^x for which no edge of Z_1 or Z_2 is relevant with respect to $(\mathcal{B}_t, x)$. Moreover, we set

$$\text{GOOD}_x := \left\{ (Z_1, Z_2) : \begin{array}{l} Z_1 \text{ consists of edges below level } k-1 \text{ w.r.t. } \mathcal{B}_t, \\ Z_2 \text{ consists of edges between level } k-1 \text{ and } k \text{ w.r.t. } \mathcal{B}_t, \\ |Z_1| = \ln^4(n), \quad \text{and} \quad |Z_2| = n^{1/3+0.5\varepsilon}. \end{array} \right\} .$$

We are now ready to give the full description of Walker's strategy.

Sequence I: If $|N_a| = n^{1/3}$ and Walker has an edge between N_a and any set $C^x[Z_1, Z_2]$ with $(Z_1, Z_2) \in \text{GOOD}_x$, she proceeds with Sequence II immediately. Otherwise, she distinguishes two cases.

Case 1: Let $|N_a| < n^{1/3}$. Then Walker proceeds as follows:

(i) Walker identifies a vertex $w \in N(a, R)$ such that aw is free.

(ii) She walks from a to w (hence claiming aw) and back to a.

N_a is updated by adding w to this set. Walker continues with Sequence II afterwards.

Case 2: Let $|N_a| = n^{1/3}$. Walker then plays a sequence of two moves according to strategy $\mathcal{S}_{paths}$ with bias $8k + 14$, with $A = N_a$, and $C_1, \ldots, C_s$ being replaced with all sets $C^x[Z_1, Z_2]$ such that $(Z_1, Z_2) \in \text{GOOD}_x$. Afterwards she proceeds with Sequence II.

Sequence II: If Walker's graph is already spanning, Walker proceeds with Sequence III immediately. Otherwise, Walker plays as follows: Let x be a vertex with $x \notin V(W)$ for which weight(x) is largest and let $t \in [2]$ be such that $x \in V_{3-t}$. Walker includes x in her graph by the following steps:

(i) Walker identifies a vertex $v \in C^x(s_{k,1})$ such that the following properties hold:

– there is a path P_v of length 2 between a and v, the edges of which are available,

– there is a $(\mathcal{B}_t, x)$-structure $\mathcal{S}_{v,x}$ starting in v, the edges of which are available.

The existence of such a vertex is proven later in the strategy discussion.

(ii) For her first move, Walker walks along P_v to reach the vertex v.

(iii) For her next k moves, Walker follows strategy $S_{\text{structure}}$ on the structure $\mathcal{S}_{v,x}$ until she reaches x.

(iv) Finally, Walker takes at most $k+1$ further moves to return to vertex a and then proceeds with Sequence III.

Sequence III: Let $b := 8k + 14$. If Walker has already tossed a coin on every edge, she stops playing. Otherwise she does the following. Before her move, Walker updates the simulated game $\text{MinBox}(n, 4pn, 0.5\ln^{-1}(n), 2b)$: Let $e_1, \ldots, e_s$ be the edges that Breaker claimed since the last time when Walker identified an exposure vertex. Then, for each vertex $v \in V(G)$ increase $w_B(J_v)$ by the cardinality of $\{i \leqslant s : \; v \in e_i\}$. Walker now plays as follows:

Case 1: After the update, let there be a free active box in $\text{MinBox}(n, 4pn, 0.5\ln^{-1}(n), 2b)$. Walker first chooses an exposure vertex $x \in V(G)$ for which J_x is a free active box in $\text{MinBox}(n, 4pn, 0.5\ln^{-1}(n), 2b)$ and such that

$$\text{dang}(J_x) := w_B(J_x) - 2b \cdot w_M(J_x)$$

is largest. Then she increases $w_M(J_x)$ by one in the simulated MinBox game (i.e. in the MinBox game she pretends to claim an element in the box J_x), and afterwards plays on G exactly like in Sequence II with the following additional step (which lasts for 1 round) between steps (iii) and (iv), after she reaches the exposure vertex x.

Once Walker's position is the exposure vertex x, Walker starts with the exposure process. For this, she first fixes a random ordering $\pi : [|U_x|] \to U_x$ of the vertices in U_x. According to that ordering, she then tosses a coin on the vertices of U_x independently at random, such that for each coin toss the probability of success equals $\ln^{-1}(n)$. Moreover, she tosses a coin until either the first success happens or until she tossed a coin on every element of U_x without any success. According to these different outcomes, Walker considers the following subcases in her strategy.

- *No success (failure of type I):* If none of the coin tosses happens to be a success, Walker declares this exposure round as a failure of type I. In particular, she increases the counter $f_I(x)$ by one, and makes an arbitrary move after which she ends in the vertex x again. In the auxiliary game $\text{MinBox}(n, 4pn, 0.5\ln^{-1}(n), 2b)$ she imagines to receive $2pn\ln^{-1}(n)$ elements of the box J_x (or all of the remaining free elements of J_x if there are less than $2pn\ln^{-1}(n)$), so that J_x is no longer a free active box. Moreover, in the Walker-Breaker game on G, she sets $U_x = \varnothing$ and removes x from every set U_w with $w \neq x$.

- *Success without failure:* Assume that for some $k \in \mathbb{N}$, the vertex $\pi(k)$ is the first vertex in U_x for which the coin toss is a success, and assume further that $x\pi(k)$ is still a free edge. Then Walker walks along this edge in both directions and hence claims $x\pi(k)$. Afterwards, the following updates happen: for every $i \in [k]$, the vertex x is removed from $U_{\pi(i)}$ and the vertex $\pi(i)$ is removed from U_x. Additionally, in the MinBox game, Maker claims an arbitrary free element of the box $J_{\pi(k)}$, i.e. $w_M(J_{\pi(k)})$ is increased by one.
- *Success with failure (failure of type II):* Assume that for some $k \in \mathbb{N}$, the vertex $\pi(k)$ is the first vertex in U_x for which the coin toss is a success, and assume further that $x\pi(k)$ is not a free edge any more. Then Walker makes an arbitrary move after which she ends in x again, and she declares this exposure round as a failure of type II. Afterwards, the following updates happen: Walker increases both counters $f_{II}(x)$ and $f_{II}(\pi(k))$ by one. For every $i \in [k]$, the vertex x is removed from $U_{\pi(i)}$ and the vertex $\pi(i)$ is removed from U_x. Additionally, in the MinBox game, Maker claims an arbitrary free element of the box $J_{\pi(k)}$, i.e. $w_M(J_{\pi(k)})$ is increased by one.

Case 2: After the update, let there be no free active box in MinBox$(n, 4pn, 0.5\ln^{-1}(n), 2b)$. Then Walker tosses a coin on every edge $uv \in E(G)$ on which she did not toss a coin yet. If the coin toss for uv is successful, she increases $f_{II}(u)$ and $f_{II}(v)$ by one. Afterwards, Walker stops playing the game.

Strategy discussion:

The goal of the strategy discussion is to prove that Walker can always follow the described strategy, and that by following this strategy, Walker a.a.s. ensures that (4.2.1) holds by the end of the game. As explained earlier in our setup, this is enough to conclude Theorem 4.2.7.

We start by proving some statements that are guaranteed to hold as long as Walker can follow her strategy.

Claim 4.2.9 (Maintaining properties I)**.** *As long as Walker can follow the strategy described in the strategy description segment, the following holds: Breaker claims at most* $8k + 14$ *edges between any two consecutive sequences of the same type (I, II, or III).*

Proof. Each application of Sequence I, II, and III lasts at most $2, 2k + 2$, and $2k + 3$ rounds for Walker, respectively. Therefore, the number of edges claimed by Breaker between two consecutive sequences of the same type are at most $2 \cdot (2 + 2k + 2 + 2k + 3) = 8k + 14$. □

Claim 4.2.10 (Maintaining properties II)**.** *As long as Walker can follow the strategy described in the strategy description segment, the following holds: For every* $x \in V(G)\backslash V(W)$*, it holds that* weight$(x) < \ln^2(n)$*. In particular, taking* $t \in [2]$ *such that* $x \in V_{3-t}$*, Breaker claims at most* $\ln^4(n)$ *edges below level* $k-1$ *that see* x *with respect to* $\mathcal{B}_t$*, and claims at most* $n^{1/3+\varepsilon/2}$ *edges between level* $k-1$ *and* k *that see* x *with respect to* $\mathcal{B}_t$*.*

Proof. While the Walker-Breaker game is in progress, consider an auxiliary Continuous Box game with a box F_x for each vertex $x \in V(G)\backslash\{a\}$. Let CMaker and CBreaker play as follows:

(1) If Breaker claims an edge e in the Walker-Breaker game on G, then do the following: for every $t \in [2]$ and $x \in V_{3-t}$ such that e sees x w.r.t. $\mathcal{B}_t$, CMaker adds weight$_t(e)$ to the box F_x.

(2) If Walker chooses a vertex x according to Sequence II, i.e. such that weight(x) is largest, then do the following: at the beginning of the sequence, i.e. already before Walker starts to play according to the steps (i)–(iv) of Sequence II, CBreaker destroys box F_x in the auxiliary Continuous Box game.

Then we observe the following. With every edge e that Breaker claims, CMaker adds a total weight of at most 2 over all the boxes. This follows from the definition of the weights and property (E) in Lemma 4.2.5. Indeed, let e be any edge that Breaker claims, and let $t \in [2]$. If e is an edge below level $k-1$ w.r.t. $\mathcal{B}_t$, then weight$_t(e) = \ln^{-2}(n)$ by definition, and by (E1) from Lemma 4.2.5 we know that e sees at most $\ln^2(n)$ vertices $x \in V_{3-t}$ w.r.t. $\mathcal{B}_t$. Similarly, if e is an edge between levels $k-1$ and k w.r.t. $\mathcal{B}_t$, then weight$_t(e) = n^{-1/3-0.1\varepsilon}$ by definition, and by (E2) from Lemma 4.2.5 we know that e sees at most $n^{1/3+0.1\varepsilon}$ vertices $x \in V_{3-t}$ w.r.t. $\mathcal{B}_t$. Hence, in any case, a total weight of at most 1 is added over all boxes F_x with $x \in V_{3-t}$, and hence at most 2 over all boxes.

Since between any two applications of Sequence II Breaker claims at most $8k + 14$ edges, we obtain that CMaker adds a total weight of at most $2 \cdot (8k + 14) = 16k + 28$ over all boxes before CBreaker destroys the next box. Hence, we can consider the bias to be $(16k + 28 : 1)$ for this auxiliary game. Moreover, by the description in (2) we also know that CBreaker follows the strategy $\mathcal{S}$ described shortly before Lemma 2.2.4. Therefore, Lemma 2.2.4 guarantees that CMaker never manages to obtain more weight than $(16k + 28) \cdot (\ln(n) + 1)$ in any box which is still not destroyed. We conclude that we have

$$\text{weight}(x) \leqslant (16k + 28) \cdot (\ln(n) + 1) + (16k + 28) < \ln^2(n)$$

as long as $x \notin V(W)$. Note that the additional factor of $(16k + 28)$ is added to also have a bound for all the intermediate rounds in the Walker-Breaker game in which we do not make an update in the Continuous Box game.

Now, fix any $x \notin V(W)$ and let $t \in [2]$ be such that $x \in V_{3-t}$. Since every edge below level $k-1$ that sees x with respect to $\mathcal{B}_t$ has weight$_t(e) = \ln^{-2}(n)$, it follows from weight$(x) < \ln^2(n)$ that Breaker cannot have claimed more than $\ln^4(n)$ of these edges. Similarly, since every edge between level $k-1$ and k that sees x with respect to $\mathcal{B}_t$ has weight$_t(e) = n^{-1/3-0.1\varepsilon}$, it follows that Breaker cannot claim more than $n^{1/3+0.1\varepsilon}\ln^2(n) < n^{1/3+\varepsilon/2}$ of these edges. □

Having the above claim in hand, we are now able to prove that Walker can always follow the proposed strategy. We start with Sequence I.

Claim 4.2.11 (Following Sequence I). *Walker can always follow the proposed strategy of Sequence I.*

Proof. If Case 1 happens, i.e. if $|N_a| < n^{1/3}$, then this means that Sequence I happened less than $n^{1/3}$ times so far, and by Claim 4.2.9 Breaker has claimed at most $n^{1/3}(8k+14)$ edges in total. Moreover, by Property (S) of Lemma 4.2.5, we know that $|N(a,R)| \geqslant n^{1/3+\varepsilon/2}$. Hence, Walker can find a vertex w as described in step (i) of this case and thus can follow her strategy.

Otherwise, if Case 2 happens, it remains to show that Walker can indeed follow the strategy $\mathcal{S}_{paths}$ as proposed, which means that we need to check if the requirements of Lemma 4.2.8 are fulfilled. Therefore, we need to check that $|\,\mathrm{GOOD}_x\,| \leqslant \exp(n^{1/3+\varepsilon})$ and that, when Case 2 happens for the first time, for each $(Z_1, Z_2) \in \mathrm{GOOD}_x$ we have at least $n^{1/3+1.1\varepsilon}$ available edges between $C^x[Z_1, Z_2]$ and N_a. For the first requirement observe that the number of pairs $(Z_1, Z_2) \in \mathrm{GOOD}_x$ is at most

$$\begin{aligned}\binom{n^2}{\ln^4(n)} \cdot \binom{n^2}{n^{1/3+\varepsilon/2}} &< \left(n^2\right)^{\ln^4(n)} \cdot \left(n^2\right)^{n^{1/3+\varepsilon/2}} = n^{2\ln^4(n)+2n^{1/3+\varepsilon/2}} \\ &= \exp\left(\ln(n)\cdot\left(2\ln^4(n)+2n^{1/3+\varepsilon/2}\right)\right) = \exp\left(2\ln^5(n)+2\ln(n)\cdot n^{1/3+\varepsilon/2}\right) \\ &< \exp\left(n^{1/3+\varepsilon}\right)\end{aligned}$$

for large enough n. For the number of available edges, observe that at most $n^{1/3}\,(8k+14)$ edges were claimed by Breaker when Case 2 happens for the first time, while Property (R) ensures that $e_G\big(N_a, C^x[Z_1, Z_2]\big) \geqslant n^{1/3+1.5\varepsilon}$ for every $(Z_1, Z_2) \in \mathrm{GOOD}_x$. Hence, more than $n^{1/3+1.1\varepsilon}$ of these edges are still free. □

By following Sequence I, Walker additionally maintains the following property.

Claim 4.2.12 (Maintaining properties III)**.** *From the moment when $|N_a| = n^{1/3}$ holds for the first time, the following is always true: Let $t \in [2]$ and $x \in V_{3-t}$. Let $(Z_1, Z_2) \in \mathrm{GOOD}_x$, then there is a path (a, y, v) of length 2 such that both its edges are available and $y \in N_a$ and $v \in C^x[Z_1, Z_2]$.*

Proof. Since Walker can always follow the strategy of Sequence I, she can therefore also always follow the strategy of $\mathcal{S}_{paths}$ from the moment when $|N_a| = n^{1/3}$. Hence we know that by the end of the game she will have claimed an edge between N_a and each of the mentioned sets $C^x[Z_1, Z_2]$. In particular, this means that she eventually claims a path (a, y, v) of length 2 such that $y \in N_a$ and $v \in C^x[Z_1, Z_2]$. Hence, the claim follows since all free edges and all of Walker's edges are available. □

Let us prove next that Walker can always follow her strategy when she needs to move according to Sequence II or Sequence III.

Claim 4.2.13 (Following Sequences II and III)**.** *Walker can always follow the proposed strategy of Sequences II and III.*

Proof. Let us consider Sequence II first and assume that Walker so far could always follow her strategy. Recall that the sequence starts at the vertex a. Moreover, assume further that Walker has fixed a vertex x according to the description of Sequence II, and let $t \in [2]$ such that $x \in V_{3-t}$.

We start by checking that Walker can follow step (i). For this, before the sequence starts, denote with Z_1 and Z_2 the edges of Breaker which are below level $k-1$ or between level $k-1$ and k, respectively, which see vertex x with respect to $\mathcal{B}_t$. By Claim 4.2.10, we have $(Z_1, Z_2) \in \text{GOOD}_x$. We have to consider two cases.

First assume that $|N_a| < n^{1/3}$. Then Sequence I happened less than $n^{1/3}$ times so far and, following Claim 4.2.9, at most $(8k + 14)n^{1/3}$ rounds have been played so far. In particular, we can find a set $A \subseteq N_G(a, R)$ of size $n^{1/3}$ such that all edges of $E_G(a, A)$ are free. Moreover, by Property (R) of Lemma 4.2.5 it follows that there are at least $n^{1/3+1.5\varepsilon}$ edges between A and $C^x[Z_1, Z_2]$ in G. Among these edges, there must be free edges since at most $(8k+14)n^{1/3}$ rounds have been played so far; denote one such edge with yv such that $y \in A$ and $v \in C^x[Z_1, Z_2]$. Then the path $P_v = (a, y, v)$ consists of available edges. Moreover, since $v \in C^x[Z_1, Z_2]$ and because of Observation 4.2.6, there exists a $(\mathcal{B}_t, x)$-structure $\mathcal{S}_{v,x}$ incident with v, where none of the edges is claimed by Breaker.

Next assume that $|N_a| = n^{1/3}$. Then, by Claim 4.2.12, we can find a path $P_v = (a, y, v)$ consisting of available edges such that $y \in N_a$ and $v \in C^x[Z_1, Z_2]$. Hence, Walker can follow (i) as before.

Now, step (ii) can easily be followed, since the edges of P_v are available when Walker starts the sequence. Walker can follow step (iii) because of Lemma 4.2.3. Note that for steps (i)–(iii) Walker only needs $k + 1$ turns. By returning to a using the the same edges, Walker can easily follow step (iv).

Now let us consider Sequence III. Its discussion is almost the same as the discussion of Sequence II, except that step (iv) in this sequence is new. For this step note that whenever a failure happens, i.e. either no success on an edge or a success on an edge which is not free, then the strategy requires Walker to walk along an arbitrary available edge incident with x. Let us briefly mention, that at least one such edge exists since Walker just reached x and hence occupies an edge incident with x. Otherwise, if no failure happens, i.e. success on a free edge, then by following her strategy Walker claims this free edge which is incident with her current position x. Hence, in any case, Walker can follow that part of the strategy. □

Now, since Walker can always follow the strategy of Sequences I and II, it follows that by the end of the game her graph is spanning. In order to show that the inequality (4.2.1) is very likely to hold, we need to take a closer look at the randomized moves of the additional step in Sequence III. The following four claims are analogues of Claims 3.1–3.4 in [33].

Claim 4.2.14 (Analysing random moves I)**.** *As long as Walker does not play according to Case* 2 *of Sequence III, for every $v \in V(G)$ it holds that $w_B(J_v) < 2pn$ and $w_M(J_v) < 2pn(1 + \ln^{-1}(n))$.*

Proof. The bound on $w_B(J_v)$ follows from Claim 2.3.3 and the fact that Breaker claims an element of J_v in the MinBox game if and only if he claims an edge incident to v. We observe that $w_M(J_v)$ is updated under three scenarios. It is increased by one whenever v is the exposure vertex. It is also increased by one if v is not the exposure vertex but the coin shows success on an edge incident with v. Since the maximum degree in G is a.a.s. smaller than $2pn$ (see Claim 2.3.3), these first two scenarios contribute at most $2pn$ to $w_M(J_v)$. Moreover, $w_M(J_v)$ is increased by up to $2pn \ln^{-1}(n)$ whenever Walker encounters a failure of type I at vertex v; however this can happen at most once. Hence, $w_M(J_v)$ cannot become larger than $2pn(1 + \ln^{-1}(n))$. ☐

Claim 4.2.15 (Analysing random moves II). *For every vertex $v \in V(G)$, the box J_v becomes inactive before $d_B(v) \geqslant \frac{\varepsilon pn}{5}$.*

Proof. Assume on the contrary that at some point in the game there is an active box J_v with $w_B(J_v) = d_B(v) \geqslant \frac{\varepsilon pn}{5}$. In the first round in which $w_B(J_v)$ exceeds $\frac{\varepsilon pn}{5}$ we must have $\frac{\varepsilon pn}{5} \leqslant w_B(J_v) \leqslant \frac{\varepsilon pn}{5} + b$. At that moment J_v must still be free, since $w_M(J_v) < 2pn(1+\ln^{-1}(n))$ by the previous claim and since $|J_v| = 4pn$ by definition. But then, by Theorem 2.2.5, $\text{dang}(J_v) = w_B(J_v) - 2b \cdot w_M(J_v) \leqslant 2b(\ln(n) + 1)$ which can be rearranged to yield $w_M(J_v) \geqslant \frac{w_B(J_v)}{2b} - (\ln(n) + 1) \geqslant \frac{\varepsilon pn}{10b} - (\ln(n) + 1) \geqslant 0.5 \ln^{-1}(n) \cdot 4pn$ for large enough n. However, this contradicts the assumption that J_v is still active. ☐

Claim 4.2.16 (Analysing random moves III). *A.a.s. the following happens: For every vertex $v \in V(G)$, as long as $U_v \neq \varnothing$, the box J_v is active. Hence, a.a.s. no edge of G will become exposed because of Case 2 in Sequence III.*

Proof. Applying Claim 2.3.3 for $H \sim G_{\ln^{-1}(n)}$, we have that a.a.s. every vertex $v \in V(H)$ satisfies $d_H(v) \leqslant (1 \pm 2\varepsilon)pn \ln^{-1}(n)$. In particular, the following is a.a.s. true: fixing any vertex v, there happen to be at most $(1 \pm 2\varepsilon)pn \ln^{-1}(n)$ incident edges on which the coin toss is a success. From now on, condition on this property.

Assume that the statement of the claim is wrong, i.e. at some moment in the game there is a vertex v for which we have $U_v \neq \varnothing$ and the box J_v is inactive. The assumption $U_v \neq \varnothing$ means that not every edge at v has been exposed yet, and hence no failure of type I has happened where v was the exposure vertex. The assumption of J_v being inactive leads to $w_M(J_v) \geqslant 2pn \ln^{-1}(n)$ by definition. But then, since $f_I(v) = 0$, there must have been $2pn \ln^{-1}(n)$ edges incident with v on which the coin toss was successful, which contradicts our assumption. ☐

Claim 4.2.17 (Analysing random moves IV). *A.a.s. the following happens: For every vertex $v \in V(G)$, we maintain $f_{II}(v) \leqslant 0.5\varepsilon pn \ln^{-1}(n)$.*

Proof. By Claim 4.2.16 we have that a.a.s. all edges of $H \sim G_{\ln^{-1}(n)}$ become exposed because of Case 1 in Sequence III. Moreover, by Claim 4.2.15, for every $v \in V(H)$ we have $d_B(v) \leqslant \frac{\varepsilon pn}{5}$ as long as the corresponding box J_v is active. Since $f_{II}(v)$ counts the number of failures of type II at v, i.e. the number of successes on edges incident with v which have already been claimed by Breaker, we observe that $f_{II}(v)$ is stochastically dominated by $\text{Bin}(\frac{\varepsilon pn}{5}, \ln^{-1}(n))$. Applying Chernoff's inequality (Lemma 2.3.1), we conclude that

$$\mathbb{P}\left(f_{II}(v) \geqslant 0.5\varepsilon pn \cdot \ln^{-1}(n)\right) \leqslant e^{-\varepsilon pn/(15\ln(n))} \leqslant e^{-n^{1/3}}$$

for large enough n. The claim now follows by taking a union bound over all vertices v. ☐

Finally, let us explain why the above statement implies that a.a.s. (4.2.1) holds by the end of the game. By a standard probabilistic argument (analogously to Claim 2.3.3) we a.a.s. have $d_H(v) \geqslant (1-\varepsilon)pn\ln^{-1}(n)$ for every $v \in V(G)$. Out of these edges, a.a.s. at most $0.5\varepsilon pn\ln^{-1}(n)$ are failures of type II, while the remaining edges end up in Walker's graph. This yields

$$d_{H\cap W}(v) \geqslant d_H(v) - \underbrace{0.5\varepsilon pn\ln^{-1}(n)}_{\leqslant \varepsilon d_H(v)} \geqslant (1-\varepsilon)d_H(v)$$

for every $v \in V(H)$. Thus, Theorem 4.2.7 is proven. □

4.2.4 Proof of Walker's Technical Lemma

In this subsection we give a full proof of Walker's Technical Lemma 4.2.5. In order to do so, we will first prove two simple claims that we will require later. As before, we do not intend to optimize polylogarithmic factors. Hence, in order to simplify calculations, we sometimes make generous estimates.

To reiterate, Walker's Technical Lemma reads as follows:

Lemma 4.2.5 (Walker's Technical Lemma). *For every $\varepsilon \in \mathcal{R}$ there exists a positive integer k such that the following holds. Let the structure $\mathcal{S}_k$ be given with the labelling from Subsection 4.2.1. Further, let $p \geqslant n^{-2/3+\varepsilon}$ and $G \sim G_{n,p}$. Then a.a.s. there exist a vertex $a \in V(G)$, a partition $V(G)\backslash\{a\} = V_1 \cup V_2$, families $\mathcal{B}_t = \{B_t(s) : s \in V(\mathcal{S}_k - s_0)\}$ of pairwise disjoint subsets of V_t each of size $\frac{n}{3^{k+10}}$ for every $t \in [2]$, and a set $R \subseteq V\backslash(\bigcup_{s\in V(\mathcal{S}_k - s_0)}(B_1(s) \cup B_2(s)) \cup \{a\})$ such that the following properties hold:*

(S) <u>*Sizes:*</u> *$|R| \geqslant \frac{n}{2}$, $|B_t(s)| = \frac{n}{3^{k+10}}$ for every $s \in V(\mathcal{S}_k - s_0)$ and $t \in [2]$, and $|N(a,R)| \geqslant n^{1/3+0.5\varepsilon}$.*

(C) <u>*Candidate sets:*</u> *Let $t \in [2]$. For all $x \in V_{3-t}$ there exist candidate sets $C^x(s) \subseteq B_t(s)$ for every $s \in V(\mathcal{S}_k - s_0)$ such that:*

- *(C1)* <u>*Number of candidates:*</u> *$|C^x(s)| = n^{(3^{\ell+1}-3)\varepsilon}\ln^{\pm 3^{3\ell}}(n)$ for every $\ell \in [k]$ and $s \in L_\ell$.*
- *(C2)* <u>*Neighbourhoods for main levels:*</u> *For every $(\ell, i) \in I(k)$, every vertex $v \in C^x(s_{\ell,i})$ has a neighbour in each of the sets $C^x(s^*_{\ell-1,j})$ with $3i-2 \leqslant j \leqslant 3i$.*
- *(C3)* <u>*Neighbourhoods for secondary levels:*</u> *For every $(\ell, i) \in I^*(k)$, every vertex $v \in C^x(s^*_{\ell,i})$ has a neighbour in $C^x(s_{\ell,i})$, where we set $C^x(s_{0,i}) = \{x\}$.*

Set $C^x := C^x(s_{k,1})$ from now on.

(E) <u>*Edge appearances:*</u> *Let $t \in [2]$. For every edge $e \in E(G)$ the following holds:*

- *(E1) If e appears below level $k-1$ with respect to $\mathcal{B}_t$, then e sees at most $\ln^2(n)$ vertices $x \in V_{3-t}$ with respect to $\mathcal{B}_t$.*

(E2) If e appears between levels $k-1$ and k with respect to $\mathcal{B}_t$, then e sees at most $n^{1/3+0.1\varepsilon}$ vertices $x \in V_{3-t}$ with respect to $\mathcal{B}_t$.

(R) Relevance of edges: Let $t \in [2]$ and $x \in V_{3-t}$. For any set Z_1 of edges appearing below level $k-1$ with respect to $\mathcal{B}_t$ and any set Z_2 of edges appearing between levels $k-1$ and k with respect to $\mathcal{B}_t$, let $C^x[Z_1, Z_2]$ denote the vertices in C^x for which no edge of $Z_1 \cup Z_2$ is relevant with respect to $(\mathcal{B}_t, x)$. Then the following holds: If $|Z_1| = \ln^4(n)$ and $|Z_2| = n^{1/3+0.5\varepsilon}$, and if $A \subseteq N(a, R)$ has size $|A| = n^{1/3}$, then $e_G\big(A, C^x[Z_1, Z_2]\big) \geqslant n^{1/3+1.5\varepsilon}$.

Our main tools for probabilistic arguments will be Lemma 2.3.1 and Lemma 2.3.2. As mentioned before, these inequalities by Chernoff are useful to verify that a given binomial random variable $X \sim \text{Bin}(n, p)$ is concentrated around its expectation $\mathbb{E}(X) = np$.

As promised, before we give the proof of Lemma 4.2.5, let us prove the following two claims, which we will use later on.

Claim 4.2.18. *Let $\varepsilon > 0$, let n be a large enough integer, and let $p = n^{-2/3+\varepsilon}$. Moreover, let $A, B \subset [n]$ be any disjoint sets such that $1 \leqslant |A| \leqslant p^{-1}$ and $|A||B| > n^{2/3}$. When we reveal the edges of $G \sim G_{n,p}$ on the vertex set $[n]$, then with probability at least $1 - \exp(-0.5\ln^2(n))$ it holds that*

$$|N_G(A) \cap B| = \ln^{\pm 2}(n) p |A||B| .$$

Proof. Let $\delta \in (0, 0.1)$, and whenever necessary assume n to be large enough.

At first we observe that $e_G(A, B) \sim \text{Bin}(|A||B|, p)$ with expectation $\mathbb{E}(e_G(A, B)) = |A||B|p$. Hence, using Chernoff (Lemma 2.3.1) and $|A||B|p > n^{\varepsilon}$, we get that with probability at least $1 - \exp(-n^{\varepsilon/2})$ the following holds:

$$e_G(A, B) = (1 \pm \delta)|A||B|p . \tag{4.2.2}$$

Furthermore, for every $v \in B$ we have $(v, A) \sim \text{Bin}(|A|, p)$ and $\mathbb{E}(d_G(v, A)) = p|A| \leqslant 1$. Hence, applying Chernoff (Lemma 2.3.2) and union bound, we see that with probability at least $1 - n \cdot \exp(-0.9\ln^2(n))$,

$$d_G(v, A) \leqslant 0.9\ln^2(n) \quad \text{for every } v \in B . \tag{4.2.3}$$

The probability that at least one of the above events fails can be bounded with $\exp(-0.5\ln^2(n))$ by applying a union bound. Hence, we may assume that both equation 4.2.2 and inequality 4.2.3 hold. Then, using that

$$\frac{e_G(A, B)}{\max_{v \in B} d_G(v, A)} \leqslant |N_G(A) \cap B| \leqslant e_G(A, B) ,$$

the claim follows. □

Claim 4.2.19. *Let $\delta, \varepsilon > 0$, let n be a large enough integer, and let $p = n^{-2/3+\varepsilon}$. Moreover, let M_1, M_2, M_3, B_1^*, B_2^*, B_3^*, $B \subset [n]$ be any disjoint sets such that $1 \leqslant |M_j| \leqslant n^{1/3-3\varepsilon}$ and $|B|, |B_j^*| \geqslant \delta n$ for every $j \in [3]$. When we reveal the edges of $G \sim G_{n,p}$ between these seven sets, with probability at least $1 - \exp(-0.4\ln^2(n))$ the following holds:*

(P) Let $C \subseteq B$ be the set of vertices $b \in B$ such that for every $j \in [3]$ there exists a path (b, y_j, z_j) with $y_j \in B_j^*$ and $z_j \in M_j$. Then $|C| = \ln^{\pm 13}(n) n^{6\varepsilon} \prod_{j \in [3]} |M_j|$.

Proof. Whenever necessary assume n to be large enough.

Let $A_j := N_G(M_j) \cap B_j^*$. Then C is the set of vertices $b \in B$ that have a neighbour in each of the sets A_j with $j \in [3]$. In order to obtain estimates on $|C|$ we apply Claim 4.2.18 repeatedly. To do this, we first reveal the edges of $G \sim G_{n,p}$ between M_j and B_j^* for each $j \in [3]$. Then Claim 4.2.18 (applied with sets M_j and B_j^*) yields that the following is likely to hold for every $j \in [3]$:

$$|A_j| = \ln^{\pm 2}(n) p |M_j||B_j^*| . \tag{4.2.4}$$

From now on condition on these events and note that then

$$n^{1/3} < |A_j| < p^{-1}$$

holds. Next we proceed as follows: first we reveal the edges between A_1 and B to obtain $B_1 := N_G(A_1) \cap B$; afterwards we reveal the edges between A_2 and B_1 to obtain $B_2 := N_G(A_2) \cap B_1$; lastly we reveal the edges between A_3 and B_2 to obtain $C = N_G(A_3) \cap B_2$. Each time we can apply Claim 4.2.18, provided that the two relevant sets satisfy the size conditions given by Claim 4.2.18, which will always be the case.

In the first step, applying Claim 4.2.18 with sets A_1 and B yields that the following is likely to hold:

$$|B_1| = \ln^{\pm 2}(n) p |A_1||B| . \tag{4.2.5}$$

Now, using $|A_1| > n^{1/3}$ and $|B| \geqslant \delta n$, we get $|B_1| > n^{2/3}$. If we condition on this, then A_2 and B_1 satisfy the size conditions of Claim 4.2.18, and hence in the second step we obtain that the following is likely to hold:

$$|B_2| = \ln^{\pm 2}(n) p |A_2||B_1| . \tag{4.2.6}$$

Similarly to before, using $|A_2| > n^{1/3}$ and $|B_1| > n^{2/3}$, we get $|B_2| > n^{1/3}$. So, if we condition on this, then A_3 and B_2 satisfy the size conditions of Claim 4.2.18 as well, and for the third step it is likely to hold that

$$|C| = \ln^{\pm 2}(n) p |A_3||B_2| . \tag{4.2.7}$$

Since according to Claim 4.2.18 each of the events (4.2.4)–(4.2.7) fails with probability less than $\exp(-0.5 \ln^2(n))$, when taking a union bound we get that with probability at least $1 - \exp(-0.4 \ln^2(n))$ all of the events described by (4.2.4)–(4.2.7) hold at the same time. Putting everything together we get

$$\begin{aligned} |C| &\overset{(4.2.7)-(4.2.5)}{=} \ln^{\pm 6}(n) p^3 |B| \prod_{j \in [3]} |A_j| \overset{(4.2.4)}{=} \ln^{\pm 12}(n) p^6 |B| \prod_{j \in [3]} |M_j||B_j^*| \\ &= \ln^{\pm 13}(n) p^6 n^4 \prod_{j \in [3]} |M_j| = \ln^{\pm 13}(n) n^{6\varepsilon} \prod_{j \in [3]} |M_j| , \end{aligned} \tag{4.2.8}$$

which proves our claim. □

Main proof. We now have all the necessary tools available to prove Lemma 4.2.5. Let $\varepsilon \in \mathcal{R}$ be given and set $k := \log_3(2\varepsilon^{-1} + 12) - 2$. Let $V = [n]$ be the vertex set, and before revealing any edges of $G_{n,p}$, do the following: fix an arbitrary vertex $a \in [n]$ and an equipartition $V \backslash \{a\} = V_1 \cup V_2$, as well as families $\mathcal{B}_t = \{B_t(s) : s \in V(\mathcal{S}_k - s_0)\}$ of pairwise disjoint vertex subsets of V_t each of size $\frac{n}{3^{k+10}}$. Furthermore set $V(\mathcal{B}_t) = \bigcup_{B \in \mathcal{B}_t} B$ for $t \in [2]$, and set

$$R = V \backslash (V(\mathcal{B}_1) \cup V(\mathcal{B}_2) \cup \{a\}) .$$

We now start revealing the edges of $G \sim G_{n,p}$ and prove that a.a.s. all properties listed in Lemma 4.2.5 hold.

Property (S): By assumption we have $|B_t(s)| = \frac{n}{3^{k+10}}$ for every $s \in V(\mathcal{S}_k - s_0)$ and $t \in [2]$. Since $v(\mathcal{S}_k) = 2 \cdot 3^k - 1$ by Claim 4.2.2, it follows that $|R| \geqslant \frac{n}{2}$. Moreover, by revealing only the edges between a and R, we have $|N(a, R)| \sim \text{Bin}(|R|, p)$ with expectation $|N(a, R)| = |R|p \geqslant \frac{1}{2} n^{1/3+\varepsilon}$. Hence, by Chernoff (Lemma 2.3.1), Property (S) fails with probability at most $\exp(-0.1 n^{1/3+\varepsilon})$.

Property (C): Let $t \in [2]$ and $x \in V_{3-t}$. We prove that for this particular choice of t and x, the described property in (C) fails with probability at most $\exp(-0.3 \ln^2(n))$. Applying union bound concludes the argument.

Note that for this property we only need to reveal the edges in $V(\mathcal{B}_t)$ and the edges between x and boxes $B_t(s)$ with $s \in L_0^*$, and hence checking this property is independent of (S). Set $C^x(s_0) = \{x\}$. Starting from this candidate set, we find the candidate sets $C^x(s) \subseteq B_t(s)$ with $s \in L_\ell$ and $\ell \in [k]$ iteratively, starting with L_1, then L_2, and so on, while always only revealing the necessary edges.

To be more precise, one step in the iteration looks as follows: Let $\ell \in [k]$ and assume we have already fixed the candidate sets $C^x(s)$ with $s \in L_{\ell-1}$. Then we reveal all edges between boxes $B_t(s)$ with $s \in L_{\ell-1} \cup L_{\ell-1}^* \cup L_\ell$. For every $i \in [3^{k-\ell}]$ we then let $C^x(s_{\ell,i})$ be the set of all vertices $v \in B_t(s_{\ell,i})$ such that for every $j \in \{0, 1, 2\}$ it holds that there exists a path (v, y_j, z_j) with $y_j \in B_t(s_{\ell-1,3i-j}^*)$ and $z_j \in C^x(s_{\ell-1,3i-j})$. Afterwards we let $C^x(s_{\ell-1,3i-j}^*)$ be the union over of all v_j which appear in such a path.

Properties $(C2)$ and $(C3)$ immediately hold by definition. To show that $(C1)$ is likely to hold, we apply Claim 4.2.19 (with $\delta = \frac{1}{3^{k+10}}$) along the iteration. That is, in the ℓ-th step of the above iteration, having $i \in [3^{k-\ell}]$, we set $M_j = C^x(s_{\ell-1,3i-j})$, $B_j^* = B_t(s_{\ell-1,3i-j}^*)$, and $B = B_t(s_{\ell,i})$. By Claim 4.2.19 it is then likely to hold that

$$|C^x(s_{\ell,i})| = \ln^{\pm 13}(n) n^{6\varepsilon} \prod_{j \in [3]} |C^x(s_{\ell-1,3i-j})| , \tag{4.2.9}$$

from which we may conclude inductively that

$$|C^x(s_{\ell,i})| = n^{(3^{\ell+1}-3)\varepsilon} \ln^{\pm 3^{3\ell}}(n) . \tag{4.2.10}$$

Note that in this whole iteration the above application of Claim 4.2.19 happens exactly once for each vertex $s \in V(\mathcal{S}_k - s_0)$. Thus, the probability that (4.2.9) and therefore also (4.2.10) fail for some $(\ell, i) \in I(k)$ is bounded by $v(\mathcal{S}_k) \cdot \exp(-0.4\ln^2(n)) < \exp(-0.3\ln^2(n))$, provided that each time when we apply Claim 4.2.19 the required condition on $|M_j|$ holds. But now, using (4.2.10) in every step along the way, this condition is guaranteed since for every $\ell \in [k]$ and $s \in L_{\ell-1}$ the likely size of $C^x(s)$ is bounded by

$$n^{(3^\ell-3)\varepsilon}\ln^{3^{3\ell-3}}(n) \leqslant n^{(3^k-3)\varepsilon}\ln^{3^{3\ell-3}}(n) = n^{2/9+4\varepsilon/3-3\varepsilon}\ln^{3^{3\ell-3}}(n) < n^{2/9}$$

by our choice of k, and provided that n is large enough.

Property (E): Let $t \in [2]$ and $\ell \in [k]$, and fix any edge e that appears below level ℓ with respect to $\mathcal{B}_t$. For this fixed edge we prove that (E1) and (E2) each fail with probability at most $\exp(-0.5\ln^2(n))$. Hence, applying union bound over all edges, (E) fails with probability at most $\exp(-0.4\ln^2(n))$.

By Definition 4.2.4, the edge e can only see a vertex $x \in V_{3-t}$ if it is contained in a $(\mathcal{B}_t, x)$-structure. Hence, we may assume that there exist adjacent vertices $s, s' \in V(\mathcal{S}_k - s_0)$ such that e intersects both $B_t(s)$ and $B_t(s')$, and such that $s \in L^*_{\ell'}$ for some $\ell' \leqslant \ell$. If $s' = x$, then x would be the only vertex in V_{3-t} that e could see, and hence the bounds in (E1) and (E2) would hold trivially. Therefore, we further assume that $e \cap V_{3-t} = \varnothing$. Now, for such e we have that if e sees a vertex $x \in V_{3-t}$, then there must be a subgraph $S \subseteq \mathcal{S}_k$ isomorphic to $\mathcal{S}_\ell - s_0$ such that the following properties hold:

(a) $e \in E(S)$ and the vertices of e are the copies of s and s',

(b) $V(S) \subseteq V(\mathcal{B}_t)$, and

(c) each leaf of S is a neighbour of x.

Note that the properties (a) and (b) depend only on edges in $E(V(\mathcal{B}_t))$, while (c) depends on edges in $E(V(\mathcal{B}_t), V_{3-t})$. So we can first expose all edges in $E(V(\mathcal{B}_t))$ and give an upper bound on the number of possible structures S satisfying both (a) and (b). Afterwards we expose the edges in $E(V(\mathcal{B}_t), V_{3-t})$ and bound the number of vertices x such that (c) is satisfied.

For the first part it turns out that it suffices to have a reasonable upper bound on the degrees in $V(\mathcal{B}_t)$. If we reveal the edges in $E(V(\mathcal{B}_t))$, then by a simple Chernoff argument we have that with probability at least $1 - \exp(-n^{1/3})$ every vertex $v \in V(\mathcal{B}_t)$ satisfies $d(v, V(\mathcal{B}_t)) \leqslant np$. If we condition on this bound and use $v(S_\ell - s_0) = 2 \cdot 3^\ell - 2$ from Claim 4.2.2, it follows that there are at most

$$(np)^{v(S_\ell - s_0)-2} = (np)^{2\cdot 3^\ell - 4}$$

copies S of $S_\ell - s_0$ fulfilling properties (a) and (b). Now, condition on this event to hold and expose the edges between $V(\mathcal{B}_t)$ and V_{3-t}. Then for every vertex $x \in V_{3-t}$ and any copy S for which properties (a) and (b) are true the following holds:

$$\mathbb{P}\,(\text{all leaves of } S \text{ are neighbours of } x) = p^{|L_0^*|} = p^{3^\ell}\,.$$

Taking a union bound over all relevant copies of S, we then conclude that

$$\mathbb{P}\,(\exists S \text{ with properties (a),(b): all leaves of } S \text{ are in } N(x)) \leqslant (np)^{2\cdot 3^\ell - 4} \cdot p^{3^\ell} =: p^*\,.$$

Since these events are independent for all $x \in V_{3-t}$, it follows that the random variable

$$X_e := |\{x \in V_{3-t} : \exists S \text{ with properties (a),(b) such that all leaves of } S \text{ are in } N(x)\}|$$

is stochastically dominated by the random variable $\mathrm{Bin}(|V_{3-t}|, p^*) =: Y_e$. The expectation of Y_e is

$$\mathbb{E}[Y_e] = |V_{3-t}|p^* \leqslant n \cdot (np)^{2\cdot 3^\ell - 4} \cdot p^{3^\ell} = n^{\alpha(\ell)}$$

with $\alpha(\ell) = 1 + (2 \cdot 3^\ell - 4) + (-\frac{2}{3} + \varepsilon) \cdot (3 \cdot 3^\ell - 4) = -\frac{1}{3} + \varepsilon \cdot (3^{\ell+1} - 4)$. In light of (E1) and (E2), we now distinguish two cases.

Case (E1): e is below level $k-1$ ($\ell \leqslant k-1$). In this case we have $\alpha(\ell) \leqslant -\frac{1}{3} + \varepsilon \cdot (3^k - 4) \leqslant -\frac{1}{9}$ by our choice of k. In particular, we have $\mathbb{E}[Y_e] = o(1)$ and using Chernoff (Lemma 2.3.2) we get

$$\mathbb{P}\left(Y_e \geqslant \ln^2(n)\right) \leqslant e^{-\ln^2(n)} \quad \Rightarrow \quad \mathbb{P}\left(X_e \geqslant \ln^2(n)\right) \leqslant e^{-\ln^2(n)} .$$

Hence, e fails (E1) with probability at most $\exp(-n^{1/3}) + \exp(-\ln^2(n)) < \exp(-0.5 \ln^2(n))$.

Case (E2): e is between level $k-1$ and k ($\ell = k-1$). In this case we have $\alpha(\ell) = -\frac{1}{3} + \varepsilon \cdot (3^{k+1} - 4) = \frac{1}{3}$ by our choice of k. In particular, we have $\mathbb{E}[Y_e] = n^{1/3}$ and using Chernoff (Lemma 2.3.1) we get

$$\mathbb{P}\left(Y_e \geqslant n^{1/3+0.1\varepsilon}\right) \leqslant e^{-n^{1/3}} \quad \Rightarrow \quad \mathbb{P}\left(X_e \geqslant n^{1/3+0.1\varepsilon}\right) \leqslant e^{-n^{1/3}} .$$

Hence, e fails (E2) with probability at most $\exp(-n^{1/3}) + \exp(-n^{1/3}) = 2\exp(-n^{1/3})$.

Property (R): In order to verify property (R), we first need a bound on the number of candidates in $B_t(s_{k,1})$ for a given vertex $x \in V_{3-t}$, for $t \in [2]$, that belong to a $(\mathcal{B}_t, x)$-structure that also contains a fixed vertex v. For this, we define

$$c_\ell := \begin{cases} n^{(3^{k+1}-4)\varepsilon} , & \text{if } \ell \neq k-1 , \\ n^{1/9+4\varepsilon} , & \text{if } \ell = k-1 . \end{cases}$$

Then the following claim holds.

Claim 4.2.20. *With probability at least $1 - \exp(-0.1 \ln^2(n))$ the following property holds: For every $t \in [2]$, $x \in V_{3-t}$ and every $v \in B_t(s^*_{\ell,i})$, with $\ell \in [k-1]$ and $i \in [3^{k-i}]$, there are at most c_ℓ candidates in $B_t(s_{k,1})$ which belong to a $(\mathcal{B}_t, x)$-structure that also contains v.*

Before we prove the above claim, let us explain why Property (R) follows. If we condition on (C) and the good event from Claim 4.2.20 to hold, then we observe the following: Let any $t \in [2]$, $x \in V_{3-t}$, and any edge sets Z_1 and Z_2 be given as described in (R), such that $|Z_1| = \ln^4(n)$ and $|Z_2| = n^{1/3+\varepsilon/2}$. Then the number of vertices in $B_t(s_{k,1})$ for which an edge of $Z_1 \cup Z_2$ may be relevant with respect to $(\mathcal{B}_t, x)$ is bounded by

$$|Z_1| \cdot n^{(3^{k+1}-4)\varepsilon} + |Z_2| \cdot n^{1/9+4\varepsilon} = \ln^4(n) \cdot n^{2/3} + n^{4/9+9\varepsilon/2} = o(|C^x|) .$$

In particular, we then conclude that

$$|C^x[Z_1, Z_2]| = (1 - o(1))|C^x| > n^{2/3+0.9\varepsilon} .$$

Now condition on this inequality and notice that for verifying (C) and Claim 4.2.20 we did not need to expose the edges between R and $B_1(s_{k,1}) \cup B_2(s_{k,1})$. Hence, we can expose them now. Then, for each choice of t, x, Z_1, Z_2 as described above, and each $A \subseteq N(a, R)$ of size $n^{1/3}$ we have that the random variable $e_G\big(A, C^x[Z_1, Z_2]\big)$ has distribution $\mathrm{Bin}(|A||C^x[Z_1, Z_2]|, p)$ with expectation

$$\mathbb{E}(e_G\big(A, C^x[Z_1, Z_2]\big)) = |A||C^x[Z_1, Z_2]|p > n^{1/3+1.9\varepsilon}\,.$$

Using Chernoff (Lemma 2.3.1) we obtain that the desired inequality $e_G\big(A, C^x[Z_1, Z_2]\big) \geqslant n^{1/3+1.5\varepsilon}$ fails with probability at most $\exp(-n^{1/3+1.8\varepsilon})$. Hence, taking a union bound over all possible choices of t, x, A, Z_1, and Z_2, we see that (R) fails to hold with probability at most

$$\begin{aligned} &2 \cdot n \cdot \binom{|N(a,R)|}{n^{1/3}} \cdot \binom{n^2}{\ln^4(n)} \cdot \binom{n^2}{n^{1/3+\varepsilon/2}} \cdot e^{-n^{1/3+1.8\varepsilon}} \\ &\leqslant 2 \cdot \exp\Big((1 + n^{1/3} + 2\ln^4(n) + 2n^{1/3+\varepsilon/2}) \cdot \ln(n) - n^{1/3+1.8\varepsilon}\Big) = o(1)\,. \end{aligned}$$

Thus, it remains to prove Claim 4.2.20. The proof of this claim is very similar to the proof of Property (C). Let vertices x and v be given as described by Claim 4.2.20. We will prove that for this particular choice of x and v, the described property in Claim 4.2.20 fails with probability at most $\exp(-0.3\ln^2(n))$, so that the general statement follows by a union bound over all choices of x and v. By symmetry we can assume that $v \in B_2(s^*_{\ell,1})$.

In order to find the candidates as described in Claim 4.2.20, we proceed in the same way as in the discussion of Property (C), and find candidate sets $C^x_v(s)$ which are defined as before, with the only difference that these sets have to contain v. This means that we start by setting $C^x(s_0) = \{x\}$ and then continue by iteratively finding candidate sets $C^x(s) \subseteq B_t(s)$ with $s \in L_\ell$ and $\ell \in [k]$, but from box $B_2(s^*_{\ell,1})$ the only vertex we are allowed to use is v, so we replace $B_2(s^*_{\ell,1})$ with $\{v\}$. In order to bound the sizes of all sets $C^x_v(s)$ we can proceed analogously to the discussion of (C). As hereby most of the failure probabilities appearing in the different steps of the argument are the same as in steps of (C), we do not state these bounds again, and assume that a.a.s all likely events happen to hold at the same time.

Now, In the analysis of (C), nothing changes until level ℓ is reached, since we do not come across the vertex v. In particular, it is likely to hold that

$$|C^x_v(s)| = |C^x(s)| = n^{(3^{\ell+1}-3)\varepsilon} \ln^{\pm 3^{3\ell}}(n)$$

for every $s \in L_\ell$ as in (C). Next we want to upper bound the size of $C^x_v(s_{\ell+1,1})$, i.e. the candidates from $C^x(s_{\ell+1,1})$ which are adjacent to v. For this we can proceed similarly to the proof of Claim 4.2.19. Let $A_1 := \{v\}$, $A_2 := N(C^x_v(s_{\ell,2})) \cap B_t(s^*_{\ell,2})$, and $A_3 := N(C^x_v(s_{\ell,3})) \cap B_t(s^*_{\ell,3})$. Then $C^x_v(s_{\ell+1,1})$ consists of all vertices of $B_t(s_{\ell+1,1})$ that have a neighbour in each of the sets A_i with $i \in [3]$. We then have that $|A_1| = 1$, while Claim 4.2.18 gives that for $i \in \{2, 3\}$ the following is likely to hold:

$$|A_i| = \ln^{\pm 2}(n) p |C^x_v(s_{\ell,i})| \cdot |B_t(s^*_{\ell,i})|\,.$$

Now, condition on the above event. Given any $u \in B_t(s_{\ell+1,1})$ with probability at most $|A_i|p$ it holds that u has a neighbour in A_i and hence $\mathbb{P}(u \in C_v^x(s_{\ell+1,1})) \leqslant p^3|A_2||A_3|$. It follows that the random variable $|C_v^x(s_{\ell+1,1})|$ is stochastically dominated by $\text{Bin}(|B(s_{\ell+1,1})|, p^3|A_2||A_3|)$, the expectation of which is

$$\begin{aligned} |B_t(s_{\ell+1,1})| \cdot p^3|A_2||A_3| &= \ln^{\pm 4}(n) p^5 |B_t(s_{\ell+1,1})| \prod_{i \in \{2,3\}} |C_v^x(s_{\ell,i})| \cdot |B_t(s^*_{\ell,i})| \\ &\leqslant n^{\varepsilon} p^5 n \prod_{i \in \{2,3\}} \left(n^{(3^{\ell+1}-3)\varepsilon} \cdot n \right) \\ &= n^{-1/3+\varepsilon \cdot 2 \cdot 3^{\ell+1}} . \end{aligned}$$

Hence, applying Chernoff (Lemma 2.3.2), it follows that with probability at least $1-\exp(-\ln^2(n))$ it holds that

$$|C_v^x(s_{\ell+1,1})| \leqslant n^{\varepsilon} \cdot \max\left\{1, n^{-1/3+\varepsilon\cdot 2\cdot 3^{\ell+1}}\right\} .$$

Now, we need to consider two cases: $\ell \neq k-1$ and $\ell = k-1$. In the first case, we obtain

$$|C_v^x(s_{\ell+1,1})| \leqslant n^{-2\varepsilon} |C^x(s_{\ell+1,1})| .$$

Thus, because of the restriction to v, the candidate set shrinks at least by a factor of $n^{-2\varepsilon}$. If we now replace $C^x(s_{\ell+1,1})$ with $C_v^x(s_{\ell+1,1})$ in the analysis of (C), it turns out that a.a.s. the factor $n^{-2\varepsilon}$ carries over to all candidate sets $|C_v^x(s_{j,1})|$ with $j \geqslant \ell+1$, up to maybe some polylogarithmic factors. In particular, the relevant number of vertices in level k which we want to estimate can be bounded from above by $n^{(3^{k+1}-4)\varepsilon}$ as claimed. If otherwise $\ell = k-1$ holds, we get immediately that

$$|C_v^x(s_{k,1})| = |C_v^x(s_{\ell+1,1})| \leqslant n^{\varepsilon-1/3+\varepsilon\cdot 2\cdot 3^{\ell+1}} = n^{\varepsilon-1/3+4/9+24\varepsilon/9} \leqslant n^{1/9+4\varepsilon}$$

by our choice of k, which concludes the proof. □

4.3 Concluding remarks

Making use of local resilience. In Subsection 4.2.3 we proved Theorem 4.2.7, which states that when playing a (2 : 2) Walker-Breaker game on $G \sim G_{n,p}$, if $p \geqslant n^{-2/3+\varepsilon}$ Walker a.a.s. has a strategy to claim a graph that satisfies a given (p,ε)-resilient graph property $\mathcal{P}$. Since the containment of a Hamilton cycle is a (p,ε)-resilient graph property, Walker is able to claim a Hamilton cycle. But, by applying other known results on local resilience in random graphs, we can immediately deduce further results on (2 : 2) Walker-Breaker games on $G_{n,p}$. For instance, when $p \geqslant n^{-1/2+\varepsilon}$, then Walker a.a.s. has strategies to obtain a pancyclic spanning graph [51] or the square of an almost spanning cycle [59]. Due to a recent result of Fischer et al. [34] we even believe that in the mentioned range Walker can claim the square of a Hamilton cycle. We would like to state this as a problem for the interested reader.

Problem 4.3.1. *Prove the following: Let $\varepsilon \in (0,1)$. Then, for $p \geqslant n^{-1/2+\varepsilon}$, playing a (2 : 2) Walker-Breaker game on the edges of a random graph $G \sim G_{n,p}$, Walker a.a.s. has a strategy to occupy the square of a Hamilton cycle.*

Note that the strategy given in Subsection 4.2.3 is made in such a way that from some moment on Walker can play almost like Maker in a Maker-Breaker game. Whenever she wants to claim an arbitrary edge at some chosen vertex v, she is able to reach that vertex within a constant number of rounds and then claim her desired edge. Having such a strategy at hand, we are able to transfer the argument of Ferber, Krivelevich, and Naves [33] to create a random subgraph $H \sim G_{n,q}$ from which Walker a.a.s. claims all edges except at most an ε-fraction of edges at every vertex. But when we consider the square of a Hamilton cycle, in order to apply the result of [34] this argument is not sufficient. Since Fischer at al. [34] considered triangle resilience instead of local resilience, Walker would need to ensure to lose at most an ε-fraction of triangles at every vertex.

Considering different biases. When we consider the more general $(m : b)$ Connector-Breaker or Walker-Breaker game on $G_{n,p}$, then it turns out that in contrast to the usual Maker-Breaker setting the threshold probability for creating a spanning tree or Hamilton cycle highly depends on the biases m and b. The good structures considered in Subsection 4.2.3 can be generalised for any two biases m, b, which will have different threshold probabilities at which those structures will appear in $G \sim G_{n,p}$. But although most ideas from this thesis can be generalised to these doubly biased games, new ideas for Breaker's side are required. We conjecture that the threshold probability in the $(m : b)$-biased version is of order $n^{-(m(b+1)-b)/(m(b+1))+o(1)}$.

Adding more constraints to Breaker. Instead of allowing Breaker to pick his edges freely, we can instead consider the variants of Connector-Breaker and Walker-Breaker games in which Breaker also needs to play as Connector or Walker, respectively, as was suggested e.g. in [29] and [35]. We have not considered this variant, but we would be interested in how it behaves compared to the usual Maker-Breaker game setting, as well as to the Connector-Breaker game and the Walker-Breaker game, respectively.

Considering different graph properties. Lastly, it would be interesting to consider even more graph properties, and to study the relation between the Connector/Walker-Breaker games and their Maker-Breaker game analogues. For example, consider the H-game, where Maker (or Connector/Walker) wins if she claims all the edges of a copy of a given graph H of constant size. Following [10] and the approach given in [23] it turns out that for all the three types of games the threshold bias for a $(1 : b)$ game played on K_n is of the same order, namely $\Theta(n^{1/m_2(H)})$, with

$$m_2(H) = \max_{\substack{J \subset H, \\ v(J) \geqslant 3}} \frac{e(J) - 1}{v(J) - 2}$$

being the maximum 2-density of H. This is in contrast to the connectivity game and Hamiltonicity game discussed in this thesis, where the three variants of these games have widely different threshold probabilities. We wonder whether in the unbiased or $(2 : 2)$ biased H-game on $G \sim G_{n,p}$ it also holds that the threshold probabilities for winning either variant are of the same order.

Bibliography

[1] G. Adamski, S. Antoniuk, M. Bednarska-Bzdęga, D. Clemens, F. Hamann, and Y. Mogge, *Tree universality in positional games*, arXiv preprint, arXiv:2312.00503 (2023).

[2] G. Adamski, S. Antoniuk, M. Bednarska-Bzdęga, D. Clemens, F. Hamann, and Y. Mogge, *Creating spanning trees in Waiter-Client games*, arXiv preprint, arXiv:2403.18534 (2024).

[3] P. Allen, J. Böttcher, Y. Kohayakawa, H. Naves, and Y. Person, *Making spanning graphs*, arXiv preprint, arXiv:1711.05311 (2017).

[4] N. Alon and J. H. Spencer, **The Probabilistic Method**, Wiley, New-York, 2008.

[5] J. Balogh, R. Martin, and A. Pluhár, *The diameter game*, Random Structures & Algorithms 35(3) (2009), 369–389.

[6] J. Beck, **Combinatorial Games: Tic-Tac-Toe Theory**, Encyclopedia of Mathematics and its Applications 114, Cambridge University Press, 2008.

[7] J. Beck, *Remarks on positional games. I*, Acta Mathematica Hungarica 40(1–2) (1982), 65–71.

[8] M. Bednarska-Bzdęga, *On weight function methods in Chooser-Picker games*, Theoretical Computer Science 475 (2013), 21–33.

[9] M. Bednarska-Bzdęga, D. Hefetz, M. Krivelevich, and T. Łuczak, *Manipulative Waiters with Probabilistic Intuition*, Combinatorics, Probability and Computing 25(6) (2016), 823–849.

[10] M. Bednarska and T. Łuczak, *Biased positional games for which random strategies are nearly optimal*, Combinatorica 20 (2000), 477–488.

[11] B. Bollobás, **Random Graphs**, Academic Press, London, 1985.

[12] B. Bollobás and A. Thomason, *Threshold functions*, Combinatorica 7 (1987), 35–38.

[13] J.A. Bondy, *Pancyclic graphs I*, Journal of Combinatorial Theory 11 (1971), 80–84.

[14] N. Brustle, S. Clusiau, V. V. Narayan, N. Ndiaye, B. Reed, and B. Seamone, *The Speed and Threshold of the Biased Perfect Matching Game*, Procedia Computer Science 195 (2021), 190–199.

[15] N. Brustle, S. Clusiau, V. V. Narayan, N. Ndiaye, B. Reed, and B. Seamone, *The Speed and Threshold of the Biased Hamilton Cycle Game*, Procedia Computer Science 195 (2021), 200–211.

[16] V. Chvátal and P. Erdős, *Biased Positional Games*, Annals of Discrete Mathematics 2 (1978), 221–229.

[17] D. Clemens, A. Ferber, R. Glebov, D. Hefetz, and A. Liebenau, *Building spanning trees quickly in Maker-Breaker games*, SIAM Journal on Discrete Mathematics 29(3) (2015), 1683–1705.

[18] D. Clemens, P. Gupta, F. Hamann, A. Haupt, M. Mikalački, and Y. Mogge, *Fast Strategies in Waiter-Client Games*, The Electronic Journal of Combinatorics 27(3) (2020), P3.57.

[19] D. Clemens, P. Gupta, and Y. Mogge, *Walker-Breaker Games on* $G_{n,p}$, The Electronic Journal of Combinatorics 31(4) (2024), P4.14.

[20] D. Clemens, F. Hamann, Y. Mogge, and O. Parczyk, *Waiter-Client Games on Randomly Perturbed Graphs*, Extended Abstracts EuroComb 2021 (2021), 397–403.

[21] D. Clemens, L. Kirsch, and Y. Mogge, *Connector-Breaker Games on Random Boards*, The Electronic Journal of Combinatorics 28(3) (2021), P3.10.

[22] D. Clemens and M. Mikalački, *How fast can Maker win in fair biased games?*, Discrete Mathematics 341(1) (2018), 51–66.

[23] D. Clemens and T. Tran, *Creating cycles in Walker-Breaker games*, Discrete Mathematics, 339(8) (2016), 2113–2126.

[24] J. Corsten, A. Mond, A. Pokrovskiy, C. Spiegel, and T. Szabó, *On the odd cycle game and connected rules*, European Journal of Combinatorics 89 (2020), 103140.

[25] A. Csernenszky, C.I. Mándity, and A. Pluhár, *On Chooser-Picker positional games*, Discrete Mathematics 309 (2009), 5141–5146.

[26] V. Dvořak, *Waiter–Client triangle-factor game on the edges of the complete graph*, European Journal of Combinatorics 96 (2021), 103356.

[27] V. Dvořak, *Waiter-Client clique-factor game*, Discrete Mathematics 346(1) (2023), 113191.

[28] P. Erdős and J.L. Selfridge, *On a combinatorial game*, Journal of Combinatorial Theory Series A 14(3) (1973), 298–301.

[29] L. Espig, A. Frieze, M. Krivelevich, and W. Pegden, *Walker-Breaker Games*, SIAM Journal on Discrete Mathematics 29(3) (2015), 1476–1485.

[30] A. Ferber, R. Glebov, M. Krivelevich, and A. Naor, *Biased games on random boards*, Random Structures & Algorithms 46(4) (2015), 651–676.

[31] A. Ferber and D. Hefetz, *Weak and strong k-connectivity games*, European Journal of Combinatorics 35 (2014), 169–183.

[32] A. Ferber, D. Hefetz, and M. Krivelevich, *Fast embedding of spanning trees in biased Maker-Breaker games*, European Journal of Combinatorics 33 (2012), 1086–1099.

[33] A. Ferber, M. Krivelevich, and H. Naves, *Generating random graphs in biased Maker-Breaker games*, Random Structures & Algorithms 47(4) (2015), 615–634.

[34] M. Fischer, N. Škorić, A. Steger, and M. Trujić, *Triangle resilience of the square of a Hamilton cycle in random graphs*, Journal of Combinatorial Theory Series B 152 (2022), 171–220.

[35] J. Forcan and M. Mikalački, *On the WalkerMaker-WalkerBreaker games*, Discrete Applied Mathematics 279 (2020), 69–79

[36] J. Forcan and M. Mikalački, *Spanning Structures in Walker-Breaker Games*, Fundamenta Informaticae 185(1) (2022)

[37] H. Gebauer and T. Szabó, *Asymptotic Random Graph Intuition for the Biased Connectivity Game*, Random Structures & Algorithms 35 (2009), 431–443.

[38] A.W. Hales and R.I. Jewett, *Regularity and Positional Games*, Classic Papers in Combinatorics, Birkhäuser Boston (2009), 320–327.

[39] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, *A sharp threshold for the Hamilton cycle Maker-Breaker game*, Random Structures & Algorithms 34(1) (2009), 112–122.

[40] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, *Fast winning strategies in Maker-Breaker games*, Journal of Combinatorial Theory Series B 99 (2009), 39–47.

[41] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, *Fast Winning Strategies in Avoider-Enforcer Games*, Graphs and Combinatorics 25 (2009), 533—544.

[42] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, **Positional Games**, Birkhäuser Basel, 2014.

[43] D. Hefetz, M. Krivelevich, M. Stojaković, and T. Szabó, *Continuous Box game*, "http://www.math.tau.ac.il/~krivelev/BoxGame.pdf" (2011)

[44] D. Hefetz, M. Mikalački, and M. Stojaković, *Doubly Biased Maker-Breaker Connectivity Game*, The Electronic Journal of Combinatorics 19 (2012), P61.

[45] D. Hefetz and S. Stich, *On Two Problems Regarding the Hamiltonian Cycle Game*, Electronic Journal of Combinatorics 16(1) (2009), R28.

[46] S. Janson, T. Łuczak, and A. Ruciński, **Random Graphs**, Wiley, New York, 2000.

[47] F. Knox, *Two constructions relating to conjectures of Beck on positional games*, arXiv preprint, arXiv:1212.3345 (2012).

[48] M. Krivelevich, *Embedding Spanning Trees in Random Graphs*, SIAM Journal on Discrete Mathematics 24 (2010), 1495–1500.

[49] M. Krivelevich, *The critical bias for the Hamiltonicity game is* $(1 + o(1))n/\ln n$, Journal of the American Mathematical Society 24 (2011), 125–131.

[50] M. Krivelevich, *Positional Games*, arXiv preprint, arXiv:1404.2731 (2014).

[51] M. Krivelevich, C. Lee, and B. Sudakov, *Resilient Pancyclicity of Random and Pseudo-random Graphs*, SIAM Journal on Discrete Mathematics 24(1) (2010), 1–16.

[52] M. Krivelevich and N. Trumer, *Waiter-Client Maximum Degree Game*, arXiv preprint, arXiv:1807.11109 (2018).

[53] C. Lee and B. Sudakov, *Dirac's theorem for random graphs*, Random Structures & Algorithms 41(3) (2012), 293–305.

[54] A. Lehman, *A Solution of the Shannon Switching Game*, Journal of the Society for Industrial and Applied Mathematics 12 (1964), 687–725.

[55] A. Liebenau and R. Nenadov, *The threshold bias of the clique-factor game*, Journal of Combinatorial Theory Series B 152 (2022), 221–247.

[56] A. London and A. Pluhár, *Spanning Tree Game as Prim Would Have Played*, Acta Cybernetica 23(3) (2018), 921–927.

[57] M. Mikalački and M. Stojaković, *Fast strategies in biased Maker-Breaker games*, Discrete Mathematics & Theoretical Computer Science 20(2) (2018).

[58] R. Nenadov, A. Steger, and M. Stojaković, *On the threshold for the Maker-Breaker H-game*, Random Structures & Algorithms 49(3) (2016), 558–578.

[59] A. Noever and A. Steger, *Local Resilience for Squares of Almost Spanning Cycles in Sparse Random Graphs*, The Electronic Journal of Combinatorics 24(4) (2017), P4.8.

[60] L. Pósa, *Hamiltonian circuits in random graphs*, Discrete Mathematics 14(4) (1976), 359–364.

[61] M. Stojakovič and Tibor Szabó, *Positional Games on Random Graphs*, Random Structures & Algorithms 26 (2005), 204–223.

[62] D. B. West, **Introduction to Graph Theory**, Prentice Hall, 2001.