The Power of Matching for Online Fractional Hedonic Games

Martin Bullinger
martin.bullinger@bristol.ac.uk
University of Bristol
Bristol, UK

René Romen
Alexander Schlenga
rene.romen@tum.de
alexander.schlenga@tum.de
Technical University of Munich
Munich, Germany

ABSTRACT

Background: Fractional Hedonic Games (FHGs) are a prominent model for coalition formation. Independently, weighted matching on graphs is of interest. Moreover, it is known to achieve good approximation ratios for many cardinal hedonic games. We study two online models for both settings.

Objectives and Research Questions: We investigate the competitive ratio when maximizing social welfare (resp., maximum weight) for online FHGs (resp., online matching). It is known that for adversarial agent arrival, this ratio is bounded only if agents' valuations (resp., edge weights) are themselves bounded. Hence, we ask whether this changes in two natural related settings. The first is random agent arrival. The second allows for adversarial arrival but algorithms are allowed to irrevocably and entirely dissolve coalitions.

Methods: We construct worst-case instances for upper bounds and describe algorithms for lower bounds. The algorithms are matching algorithms which we show to be asymptotically optimal for the FHG setting. All results are theoretical.

Results: We find that for both our models, a constant competitive ratio is possible that is independent of the range of valuations, and we present the corresponding (asymptotically) optimal algorithms. Under random arrival, the optimal competitive ratio is $\frac{1}{3}$ for both matching and FHGs. For coalition dissolution in FHGs, the optimal competitive ratio is $\frac{1}{6+4\sqrt{2}}$.

Conclusions: Matching algorithms are shown to be powerful for maximizing social welfare in FHGs. Moreover, for the random arrival model, we demonstrate that, since it is not possible to construct an adaptive adversary, an interplay of different families of instances is a good way to prove upper bounds on the competitive ratio.

KEYWORDS

Online Matching, Coalition Formation, Fractional Hedonic Games

ACM Reference Format:
Martin Bullinger, René Romen, and Alexander Schlenga. 2026. The Power of Matching for Online Fractional Hedonic Games. In *Appears at the 8th Games, Agents, and Incentives Workshop (GAIW-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., Paphos, Cyprus, May 2026*, IFAAMAS, 21 pages.

1 INTRODUCTION

We study an online partitioning problem in which a set of agents arrives one by one and must be assigned to unique coalitions. This setting naturally generalizes online matching by allowing coalitions of arbitrary size, and, therefore, poses fundamental combinatorial challenges for algorithm designers. At the same time, it has many appealing applications at the intersection of economics, artificial intelligence, and the social sciences, ranging from assigning employees to projects or offices, to grouping students into project teams, patients into support groups, or organizing collaboration among robotic swarms [for further applications, see 3, 30]. Dependent on the context, agents may represent individuals in a society or, more broadly, firms or computer programs. A widely studied framework for modeling coalition formation is that of *hedonic games* [7, 17], in which each agent has preferences over the coalitions they belong to. The defining feature of hedonic games is that preferences depend solely on the members of an agent's own coalition, and not on how other agents are grouped.

However, even under this natural restriction, explicitly specifying preferences requires considering an exponentially large set of potential coalitions. Hence, for the sake of computational tractability, much of the literature has focused on hedonic games with inherently concise preference representations. One common approach derives an agent's preferences over coalitions from their opinion on single agents. For example, agents may assign a subjective cardinal valuation to each other agent, which is then aggregated to determine utilities over coalitions. This idea gives rise to, among others, the well-studied classes of additively separable (ASHGs) and fractional (FHGs) hedonic games [1, 7]. In ASHGs, an agent's utility for a coalition is the sum of the valuations they assign to its members. In FHGs, this sum is divided by the coalition size, i.e., utilities can be seen as an average valuation, assuming a self-valuation of 0. In this work, we focus on FHGs. Aziz et al. [1] argue that this model is well-suited for analyzing network clustering and use it to represent basic economic scenarios such as the bakers-and-millers game. Moreover, Malik et al. [28] apply FHG utilities in a peer-to-peer energy market, showing that the model leads to decent profits.

An important aspect of real-world coalition formation processes is that agents arrive over time. This has motivated the study of an online model of hedonic games by Flammini et al. [23]. In their basic model, agents arrive one by one and have to be assigned to existing coalitions of any size immediately and irrevocably. The objective is to achieve high social welfare, defined as the sum of agents' utilities. Unfortunately, this is a demanding objective in FHGs: if $V_{\min}$ and $V_{\max}$ are the minimum and maximum permitted

---

Appears at the 8th Games, Agents, and Incentives Workshop (GAIW-26). Held as part of the Workshops at the 25th International Conference on Autonomous Agents and Multiagent Systems., Armstrong, Curry, Hosseini, Mattei, Tsang, Wąs (Chairs), May 2026, Paphos, Cyprus.

absolute value of nonzero valuations, the best possible competitive ratio is $\frac{V_{\min}}{4V_{\max}}$.

A crucial role in achieving welfare approximations, whether in an offline or online setting of coalition formation, has been to employ matchings.[1] For instance, the aforementioned competitive ratio is attained by forming maximal matchings, which is even the best deterministic approach for unweighted games [23]. Moreover, the best known polynomial-time approximation algorithm for social welfare in offline FHGs, achieving a 2-approximation, is to form a maximum weight matching [21]. Similarly, in the related model of ASHGs, maximum weight matchings achieve an n-approximation of social welfare, where n is the number of agents. At the same time, an $n^{1-\epsilon}$-approximation is NP-hard to compute for any $\epsilon > 0$ [22], even if $V_{\min} = V_{\max} = 1$ [12]. Our work extends this intuition by considering two more sophisticated models of online FHGs, where we show that online matching algorithms achieve a constant asymptotically optimal competitive ratio.

In the first model, the random arrival setting, agents arrive in a uniformly random instead of an adversarial order, as in the secretary problem [20]. Hence, an algorithm has to compete well against an adversary who only fixes the game but not the precise arrival order. Notably, this avoids the worst-case example by Flammini et al. [23], which crucially relies on specifying valuations based on the previous decisions of algorithms. We achieve a $(\frac{1}{3} - \frac{1}{n})$-competitive algorithm, while no algorithm can be better than $\frac{1}{3}$-competitive. The algorithm we present is based on an online matching algorithm for a general setting of weighted matching where all agents arrive online but the number of agents is unknown. In the matching domain, we obtain the same asymptotically optimal competitive ratio. The tightness of both competitive ratios follows a unified approach: We establish an upper bound for the competitive ratio of any online matching algorithms on the tree domain, a specific domain of instances where positive valuations form trees. We then show that upper bounds on this domain directly transfer to the coalition formation setting.

In the second model, the free dissolution setting, the algorithm has to perform well under any arrival order, but it gets the additional power to dissolve coalitions. This setting is inspired by free edge dissolution in the matching domain [19] and has been introduced to coalition formation by Bullinger and Romen [14]. We show that another matching algorithm achieves the optimal competitive ratio of $\frac{1}{6+4\sqrt{2}}$, which is a factor $\frac{1}{2}$ worse than the best online algorithm in the corresponding matching domain.

2 RELATED WORK

The hedonic formation of coalitions traces back to Drèze and Greenberg [17], while hedonic games in the form studied today have been conceptualized by Bogomolnaia and Jackson [7]. The latter paper introduces the class of ASHGs while fractional hedonic games were only introduced later by Aziz et al. [1]. An overview of hedonic games can be found in the book chapters by Aziz and Savani [3] and Bullinger et al. [13].

Several authors studied various notions of stability in FHGs [1, 5, 6, 8, 9, 26], where the goal is outcomes in which agents cannot perform beneficial deviations to a more preferred coalition. Aziz et al. [2] consider welfare maximization. In addition to examining algorithms for (utilitarian) social welfare, they consider the maximization of egalitarian and Nash welfare. They prove NP-hardness of finding optimal partitions for the different objectives and give polynomial-time approximation algorithms for utilitarian and egalitarian welfare. Matching algorithms are shown to yield good approximation ratios. In particular, Aziz et al. [2] show that a maximum weight matching (MWM) is a $\frac{1}{4}$-approximation of social welfare in FHGs. This analysis was later improved and made tight by Flammini et al. [21] who prove that MWMs yield precisely a $\frac{1}{2}$-approximation.

An online model for hedonic games was first studied by Flammini et al. [23], who consider FHGs and ASHGs.[2] They investigate the model where agents arrive in an adversarial order and give lower and upper bounds for deterministic algorithms on the achievable competitive ratio for maximizing social welfare. Except for simple FHGs, their results are rather discouraging because the competitiveness crucially depends on the range of valuations. For ASHGs, Bullinger and Romen [14] consider the random arrival and the free dissolution models and show that these dependencies vanish. We achieve similar results for FHGs. Furthermore, going beyond welfare maximization, Bullinger and Romen [15] study stability and Pareto optimality for online ASHGs with adversarial agent arrival.

There is a vast body of literature on online matching. A recent survey is given by Huang et al. [25]. Here, we only discuss the works that are closest to our setting. For unweighted graphs, Gamlath et al. [24] give the online algorithm with the currently best known competitive ratio for maximum cardinality matchings with adversarial vertex arrival. Kesselheim et al. [27] study MWMs with random vertex arrival on one side of bipartite graphs and show that the upper bound of $\frac{1}{e}$, which stems from the fact that the scenario generalizes the secretary problem, can be matched by an algorithm. Ezra et al. [18] present an algorithm for approximating an MWM in general weighted graphs with random vertex arrival where the total number of vertices to arrive is known in advance. They also show the asymptotic tightness of that algorithm's competitive ratio by considering a family of graphs where all edge weights differ by a large factor, so there is only one valuable edge for a matching. Our optimal matching algorithm considers their setting with an unknown number of agents. Finally, Feldman et al. [19] introduce free disposal of edges in online matching. They determine the optimal competitive ratio of $1 - \frac{1}{e}$ for the bipartite setting when one side of the agents is present offline. Bullinger and Romen [14] extend free disposal to the notion of free dissolution in coalition formation that we study.

3 PRELIMINARIES AND MODEL

We begin by introducing some notation. For $i \in \mathbb{N}$, we denote $[i] := \{1, \ldots, i\}$. Next, for a graph $G = (V, E)$ and a set of vertices $S \subseteq V$, let $G[S]$ denote the subgraph of G induced by S. Finally, we

---

[1] A notable exception are online FHGs with nonnegative weights, for which the optimal algorithm forms coalitions of unbounded size [23].

[2] While being inspired by models of hedonic games, Flammini et al. [23] develop their model as a "coalition structure generation problem" and, therefore, adopt a purely graph-theoretic instead of a game-theoretic perspective.

denote the indicator function by $\chi(\cdot)$. It takes a Boolean argument as an input and returns 1 if it is true and 0, otherwise.

3.1 Hedonic Games

Let N be a finite set of *agents*. A nonempty subset $C \subseteq N$ is called a *coalition*. The set of coalitions containing agent $i \in N$ is denoted by $\mathcal{N}_i := \{C \subseteq N \mid i \in C\}$. A set π of disjoint coalitions containing all members of N is a *partition* of N. For agent $i \in N$ and partition π, let $\pi(i)$ denote the unique coalition in π that i belongs to. Moreover, for a subset of agents $N' \subseteq N$, we define $\pi[N']$ as the *partition restricted to* N' as $\pi[N'] := \{C \cap N' \mid C \in \pi, C \cap N' \neq \emptyset\}$.

A (cardinal) *hedonic game* is a pair $G = (N, u)$ where N is the set of agents and $u = (u_i)_{i \in N}$ is a tuple of *utility functions* $u_i \colon \mathcal{N}_i \to \mathbb{Q}$. Agents seek to maximize utility and prefer partitions in which their coalition achieves a higher utility. Hence, we define the utility of a partition π for agent i as $u_i(\pi) := u_i(\pi(i))$. We denote by $n(G) := |N|$ the number of agents and write n if G is clear from the context.

Following Aziz et al. [1], a *fractional hedonic game* (FHG) is a hedonic game (N, u), where for each agent $i \in N$ there exists a *valuation function* $v_i \colon N \setminus \{i\} \to \mathbb{Q}$ such that for all $C \in \mathcal{N}_i$ it holds that $u_i(C) = \sum_{j \in C \setminus \{i\}} \frac{v_i(j)}{|C|}$. Note that this implies that the utility for a singleton coalition is 0. Since the valuation functions contain all information for computing utilities, we also represent an FHG as the pair (N, v), where $v = (v_i)_{i \in N}$ is the tuple of valuation functions. Additionally, an FHG can be succinctly represented as a complete directed weighted graph where the weights of directed edges induce the valuation functions.

An FHG (N, v) is said to be *symmetric* if for every pair of distinct agents $i, j \in N$, it holds that $v_i(j) = v_j(i)$. We write $v(i, j)$ for the symmetric valuation between i and j. A complete undirected weighted graph can represent a symmetric FHG. For simplicity, we also denote this graph by (N, v). Moreover, an FHG is said to be *simple* if for every pair of distinct agents $i, j \in N$, it holds that $v_i(j) \in \{0, 1\}$. Simple FHGs can be represented by directed unweighted graphs (where edges represent valuations of 1). Finally, a symmetric FHG is said to belong to the *tree domain* if every connected component of the edges with positive weight in the associated undirected graph forms a tree, and every other edge has a negative weight smaller than the negative sum of all positive edge weights.

We measure the desirability of a partition in terms of social welfare. Given an FHG $G = (N, v)$, we define the *social welfare* of a coalition $C \subseteq N$ in G as $\mathcal{SW}_G(C) := \sum_{i \in C} u_i(C)$ and of a partition π in G as $\mathcal{SW}_G(\pi) := \sum_{i \in N} u_i(\pi) = \sum_{C \in \pi} \mathcal{SW}_G(C)$. We omit the subscript G if the game is clear from the context. Also, we denote by $\pi^*(G)$ a partition that maximizes social welfare in G. Note that we can replace both $v_i(j)$ and $v_j(i)$ by $\frac{1}{2}(v_i(j) + v_j(i))$ for all $i, j \in N$, which results in a symmetric FHG in which the social welfare of every partition remains the same [11]. Hence, it suffices to consider symmetric FHGs instead of the full domain of FHGs. However, note that this technique cannot be applied to simple FHGs (or other restricted classes of FHGs) as the symmetrization may result in nonsimple FHGs. Given $c \leq 1$, a partition π is called a *c-approximation* to social welfare in G if $\mathcal{SW}(\pi) \geq c \cdot \mathcal{SW}(\pi^*(G))$.

3.2 Matching

A *matching* is a partition in which all coalitions have size at most 2.[3] For a matching π, we denote by $v(\pi) = \sum_{e=\{i,j\} \in \pi, |e|=2} v(e)$ its *weight*. We have that $\mathcal{SW}(\pi) = v(\pi)$ because, for each matched pair, both agents contribute $\frac{1}{2}$ of the edge weight to the social welfare. Hence, maximizing social welfare among matchings is precisely the *maximum weight matching* (MWM) problem.

Assume that we are given a complete unweighted graph $G = (N, v)$.[4] A *fractional matching* is a function $\zeta \colon V \times V \to [0, 1]$ such that, for all $i, j \in N$, we have that $\zeta(i, j) = \zeta(j, i)$ and $\sum_{j \in N} \zeta(i, j) = 1$. The value $\zeta(i, j)$ is interpreted as the probability of matching i with j, where we think of $\zeta(i, i)$ as the probability that i remains unmatched. This interpretation makes sense because of the second constraint and we can extract the probability distribution of matching to neighbors as follows. For $i \in N$, we define $\zeta(i) \colon V \to [0, 1]$ by $\zeta(i)(j) = \zeta(i, j)$.

Given a fractional matching ζ, its weight is defined by $v(\zeta) := \sum_{i,j \in N, i \neq j} \zeta(i, j) v(i, j)$. We use fractional matchings as a computational tool for one of our algorithms. For this, note that the problem of computing a *maximum weight fractional matching* (MWFM) is well known to be solvable in polynomial time [see, e.g., 31].

3.3 Online Models and Competitive Analysis

We assume an online model of FHGs where agents arrive one by one and have to be assigned to new or existing coalitions. For an agent set N, define $\Sigma(N) := \{\sigma \colon [|N|] \to N \text{ bijective}\}$. This is interpreted as the set of all *arrival orders*.

An instance (G, σ) of an *online FHG* consists of an FHG $G = (N, v)$ and an arrival order $\sigma \in \Sigma(N)$. An online coalition formation algorithm ALG produces on input (G, σ) a sequence $ALG(G, \sigma)_1, \dots, ALG(G, \sigma)_{n(G)}$ of partitions, where for each $i \in [n(G)]$, $ALG(G, \sigma)_i$ is a partition of $\{\sigma(1), \dots, \sigma(i)\}$. Hence, the partial partitions have to contain precisely the agents that have arrived so far. Moreover, we require that for all input tuples (G, σ) and (H, τ) and $k \in \mathbb{N}$ with $k \leq \min\{n(G), n(H)\}$ it holds that $ALG(G, \sigma)_k = ALG(H, \tau)_k$ whenever $v_{\sigma(i)}(\sigma(j)) = v_{\tau(i)}(\tau(j))$ for all $i, j \in [k]$.[5] This condition says that the algorithmic decision to form the kth partition can only depend on the information the algorithm has obtained until the kth agent arrives. In particular, it cannot depend on the knowledge about agents arriving in the future. Furthermore, this condition implies that decisions must be identical if all valuations are identical up to a certain agent's arrival. The output of the algorithm is the partition produced when the final agent is added; we denote $ALG(G, \sigma) := ALG(G, \sigma)_{n(G)}$.

In addition, an algorithm's decisions are assumed to be irrevocable, i.e., agents can only be added to an existing or a completely new coalition, while not changing the existing coalitions. Formally, this means that for all instances (G, σ) and $2 \leq k \leq n(G)$, we require that $ALG(G, \sigma)_k[\{\sigma(i) \mid 1 \leq i \leq k-1\}] = ALG(G, \sigma)_{k-1}$, i.e., the $(k-1)$st partition is the kth partition restricted to the first $k-1$ agents. An algorithm may, however, have the additional power to

---

[3] In contrast to the standard definition of a matching, we assume that unmatched agents are part of a matching in the form of singleton coalitions.

[4] The below definitions also work for incomplete graphs which we complete by adding edges with weight 0.

[5] We later consider randomized algorithms, for which the produced random partition has to be identical.

dissolve a partition before adding a new agent. In this case, we say that the algorithm operates under *free dissolution* and additionally allow that $ALG(G,\sigma)_k[\{\sigma(i) \mid 1 \leq i \leq k-1\}]$ is of the form $(ALG(G,\sigma)_{k-1} \setminus C) \cup \{\{i\} \mid i \in C\}$ for some $C \in ALG(G,\sigma)_{k-1}$.

The objective is to achieve high welfare. We say that ALG is *c-competitive*[6] if

$$\inf_G \min_{\sigma \in \Sigma(N)} \frac{\mathcal{SW}(ALG(G,\sigma))}{\mathcal{SW}(\pi^*(G))} \geq c.$$

Equivalently, this means that for all instances (G,σ), ALG produces a c-approximation of social welfare. Hence, we benchmark algorithms against a worst-case adversary that can both fix an instance, i.e., the number of agents and their mutual valuations, as well as an exact arrival order.

In addition, we consider a model where the agents arrive in a *uniformly random* arrival order. The objective is then to achieve high welfare in expectation. We denote by $ALG(G)$ the random partition produced with respect to a uniformly random arrival order. An algorithm ALG is said to be *c-competitive under random arrival* if

$$\inf_G \frac{\mathbb{E}_{\sigma \sim \Sigma(N)}\left[\mathcal{SW}(ALG(G))\right]}{\mathcal{SW}(\pi^*(G))} \geq c.$$

Hence, in this model, an algorithm is benchmarked against an adversary that can design a worst-case instance, but has no control over the exact arrival order of the agents. We omit the subscript from the notation of expected values when it is clear from the context. In both models, the *competitive ratio* c_{ALG} of ALG is the supremum c such that ALG is c-competitive. Note that the competitive ratio is always at most 1.

We also consider randomized algorithms, which can use randomization to decide which coalition an agent should be added to. In this case, the competitive ratio is measured with respect to the expected social welfare of the random partition constructed by the randomized algorithm.

The competitive ratio is also defined for subclasses of FHGs, such as simple and symmetric FHGs, where the infimum is only taken over games from that subclass. Finally, the competitive ratio is also defined for online matching algorithms, for which the weight of the matching produced by an algorithm is compared with the weight of an MWM.

4 CONNECTIONS OF MATCHING AND FHGS

A first significant connection between MWMs and welfare maximization in FHGs is that the former yields a $\frac{1}{2}$-approximation for the latter. In Appendix A, we provide an instructive alternative proof of this theorem, which was first shown by Flammini et al. [21]. Our argument establishes a connection between a uniform fractional matching, where each edge is included with probability $\frac{1}{n}$ and the social welfare of a coalition. The result then follows as the weight of a uniform fractional matching is a lower bound on the weight of an MWM.

Theorem 4.1. *[Flammini et al. [21]] Every MWM is a $\frac{1}{2}$-approximation of social welfare in symmetric FHGs.*

This implies the same guarantee for online algorithms: c-competitive online matching algorithms are $\frac{c}{2}$-competitive for online FHGs. We can use this insight to make an interesting observation: it is known that no deterministic online algorithm can achieve a competitive ratio of better than $\frac{1}{4}$ for simple symmetric FHGs [23]. However, there exists a *randomized* online matching algorithm for MWM on unweighted graphs (i.e., maximum cardinality matching) that beats a competitive ratio of $\frac{1}{2}$ [24], i.e., achieves a competitive ratio of $\frac{1}{2} + 2\epsilon^*$ for some constant $\epsilon^* > 0$. We can apply Theorem 4.1 to conclude that randomization can be utilized to beat the best deterministic algorithm in this case.

Corollary 4.2. *There exists $\epsilon^* > 0$ and a randomized online coalition formation algorithm for simple and symmetric FHGs with competitive ratio $\frac{1}{4} + \epsilon^*$.*

In contrast to Theorem 4.1, negative results for MWM, i.e., impossibilities of achieving a certain competitive ratio, do not transfer to FHGs. They only imply that it is impossible to create a matching of a certain quality. This does not rule out that an online algorithm can create a partition with larger coalitions that achieve more social welfare. However, we now show that negative results are inherited on domains where positive valuations form a forest (while other valuations are sufficiently negative). We provide the proof in Appendix A.

Proposition 4.3. *Let $c \leq 1$ and assume that no c-competitive (randomized) algorithm exists for online MWM on the tree domain. Then, no c-competitive (randomized) online coalition formation algorithm exists for symmetric FHGs.*

Interestingly, negative results for MWM are usually essentially achieved on the tree domain [4, 10, 14, 27],[7] which makes the previous theorem very powerful. However, even if we have a tight result for MWM where the lower bound is achieved on the tree domain, Theorem 4.1 and Proposition 4.3 leave a gap of a factor of 2. As we will see, closing this gap can take significant effort and the exact optimal competitive ratio can be at either extreme.

5 FRACTIONAL HEDONIC GAMES UNDER RANDOM ARRIVAL

While under adversarial arrival, forming maximal matchings constitutes the best deterministic online algorithm [23], this and other greedy approaches prove suboptimal under random arrival. Bullinger and Romen [14] describe a family of worst-case instances for greedy algorithms in the random arrival ASHG model, and show that they impose an upper bound of $O\left(\frac{1}{n^2}\right)$ on the competitive ratio of such algorithms. To be more precise, the bound holds for every (randomized) algorithm which always assigns a newly arrived agent to an existing coalition whenever this increases the current social welfare. Since those instances are in the tree domain, algorithms will, without loss of generality, only form matchings. Matchings yield essentially the same welfare in the ASHG and FHG model, only differing by a constant factor of 2 (in ASHGs we do not

---

[6] We use the convention that $\frac{0}{0} = 1$ and $\frac{x}{0} = 0$ for any $x \in \mathbb{Q}$ with $x < 0$.

[7] These constructions usually contain 0-weights, which can be replaced with large negative weights. In the setting of Kesselheim et al. [27], the secretary problem marks the hardest instances, so they are in the tree domain.

divide by the coalition size of 2). Hence, the above bound on the competitive ratio of greedy algorithms holds for our setting, too.

Still, based on Section 4, a reasonable strategy to obtain good online algorithms for FHGs is to consider better online matching algorithms. For the matching setting under random arrival, Ezra et al. [18] provide an algorithm that achieves a competitive ratio of $\frac{5}{12} - O(\frac{1}{n})$ if the algorithm has access to the number of agents n. Importantly, knowledge of n is crucial for achieving this competitive ratio. In the first phase of the algorithm, a subset of k agents is not matched at all, and the optimal competitive ratio is achieved for $k := \lfloor \frac{n}{2} \rfloor$. However, one can also apply their algorithm by setting k to a fixed constant. By setting $k = 3$, for instance, one obtains an online matching algorithm that is $\frac{1}{3} - O(\frac{1}{n})$-competitive. We get the following theorem.

Theorem 5.1. *There exists a randomized online matching algorithm with a competitive ratio under random arrival of at least* $\frac{1}{3} - O(\frac{1}{n})$.

By applying Theorem 4.1, we can interpret this algorithm as a coalition formation algorithm, which implies a competitive ratio of $\frac{1}{6} - O(\frac{1}{n})$ in the coalition formation domain. Its competitive ratio in this domain is upper bounded by $\frac{5}{18}$ (see Appendix B.3). However, we will now prove that we can even achieve a competitive ratio of $\frac{1}{3} - O(\frac{1}{n})$ in the coalition formation domain.

Consider Algorithm 1. This algorithm is once again a matching algorithm but we will show that it achieves the desired guarantee for FHGs. It takes as input an FHG (in its representation as a weighted graph) together with an arrival order and returns a partition that is obtained by assigning agents to coalitions of size at most 2 one by one according to their arrival.

Similar to the algorithm by Ezra et al. [18], Algorithm 1 is parameterized by a natural number $k \geq 3$, which indicates a "waiting phase" of the algorithm, i.e., the first k agents are placed in singleton coalitions. The main step of the algorithm follows a powerful idea applied in various online matching algorithms [18, 27]: Whenever an agent arrives, we compute a maximum weight matching. We then observe the edge to which the newly arrived agent is incident. If the other agent of this edge is still available, i.e., presently in a singleton coalition, we form a new coalition of size 2 and mark both agents as unavailable. However, there is a crucial difference: Unlike Kesselheim et al. [27] and Ezra et al. [18], we do not extract the possible coalition from a maximum weight *integral* matching, but from a maximum weight *fractional* matching. In fact, using a maximum weight integral matching does not yield the desired guarantee for FHGs. We provide an example that highlights the difficulties in Appendix B.3. In contrast, fractional matchings lead to a relaxation of the integral matching polytope, and can, therefore, achieve a higher maximum weight. Now, extracting an edge from the MWFM leads to a higher expected weight. While this does not improve the asymptotic worst-case performance as a matching algorithm, it improves the performance as a coalition formation algorithm.

Another crucial feature of the algorithm is that the computed MWFM in line 6 of Algorithm 1 has to be chosen independent of the set of still available agents and the specific arrival order that has led to the currently present agents. We can do this by running any algorithm for MWFM that we run for a uniformly random renaming of the agents.[8]

Algorithm 1 Online matching and coalition formation under random arrival

```
Input: Complete and weighted undirected graph $G = (N, v)$,
       arrival order $a_1, \dots, a_n$ of the vertices, parameter $k$
Output: Matching $\mu$ of $G$
1:  $A := N$, $\mu := \emptyset$ ▷ Initialize set of available vertices A, returned matching $\mu$
2:  for $t = 1$ to $k$ do
3:      Add $\{a_t\}$ to $\mu$.
4:  for $t = k + 1$ to $n$ do
5:      Let $N_t := \{a_1, \dots, a_t\}$. ▷ $N_t$ is the set of vertices arrived up to time $t$
6:      Let $\zeta_t$ be the MWFM in $G[N_t]$. In the case of multiple MWFMs in $G[N_t]$, we (randomly) choose one of them independent of $A$ and the arrival order up to time $t$.
        ▷ Recall that $\zeta(a_t)$ is the probability distribution of matching $a_t$ to agents in $N_t$, where $a_t$ remains unmatched with probability $\zeta(a_t, a_t)$
7:      Choose a random agent $p_t \in N_t$ according to distribution $\zeta_t(a_t)$.
8:      Let $e_t := \{a_t, p_t\}$.                ▷ I.e., $e_t = \{a_t\}$ if $p_t = a_t$
9:      if $p_t = a_t$ then
10:         With probability $\frac{1}{3} + \frac{2(t-4)!k!}{3(t-1)!(k-3)!}$, remove $a_t$ from $A$.
11:         Add $\{a_t\}$ to $\mu$.
12:     else
13:         if $p_t \in A$ then
14:             Remove $a_t$ and $p_t$ from $A$.
15:             Remove $\{p_t\}$ from $\mu$ and add $e_t$ to $\mu$. ▷ Add the chosen edge to the matching
16:         else
17:             Add $\{a_t\}$ to $\mu$.
18:     for $x \in (N_t \cap A) \setminus e_t$ do
19:         Remove $x$ from $A$ with probability $\frac{\zeta_t(x,x)}{t-2+\zeta_t(x,x)}$.
20: return Matching $\mu$
```

We are ready to present our proof. To make it accessible more quickly, we defer the proofs of intermediary lemmas to Appendix B.2.

Theorem 5.2. *There exists a randomized online coalition formation algorithm with a competitive ratio under random arrival of at least* $\frac{1}{3} - \frac{1}{n}$.

Proof. Throughout the proof we refer to Algorithm 1 as *ALG*. Whenever we run it for a given FHG, we assume a random arrival order.

Let A_t be the set of available agents (i.e., the current set A) after a_t has arrived and been processed. The first key step is to derive a formula for the probability that an agent is unavailable that only depends on the current iteration t as well as the number k of agents

---

[8] Kesselheim et al. [27] and Ezra et al. [18] achieve a similar property for MWMs by randomly perturbing all weights with a small number ϵ, which guarantees uniqueness of the MWM. However, we prefer a method without perturbation because choosing a suitable perturbation parameter without knowledge of the number of arriving agents would cause further complications.

in the waiting phase. In particular, this probability is the same for every agent.

Lemma 5.3. *Assume that we run ALG for any FHG (N, v) and a random arrival order. For every $k \geq 3$, $t \in \{k, \dots, n\}$, every possible realization $\tilde{N}$ of N_t (i.e., $\tilde{N} \subseteq N$, $|\tilde{N}| = t$), and every agent $a \in \tilde{N}$, it holds that*

$$\mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right) = \frac{2}{3}\left(1 - \frac{(t-3)! \cdot k!}{t! \cdot (k-3)!}\right).$$

The next key step is to derive a monotonicity property when running *ALG*. Informally, we show that the performance cannot worsen if we increase any valuation.

Lemma 5.4. *Let $G = (N, v)$, and $G' = (N, v')$ be two symmetric FHGs with $v(i, j) \geq v'(i, j)$ for all $i, j \in N$. Then,*

$$\mathbb{E}\left[\mathcal{SW}_G\left(ALG(G)\right)\right] \geq \mathbb{E}\left[\mathcal{SW}_{G'}\left(ALG(G')\right)\right].$$

The high-level idea for the proof of Lemma 5.4 is as follows. The produced social welfare of *ALG* only depends on the probability that agents are matched and the expected weight of the maximum fractional matching in each step. Now, by Lemma 5.3, increasing a single valuation does not change the probability of agents being matched, while the expected weight of the maximum fractional matchings can only become larger.

The third step is to prove the desired performance of *ALG* on the specific domain of FHGs where the valuations between agents are negative whenever they are between agents of different coalitions of some given optimal partition.

Lemma 5.5. *Assume that we run ALG for $k = 3$. Consider an FHG G together with an optimal (offline) partition $\pi^*(G)$. Assume that for all $C_1, C_2 \in \pi^*(G)$ with $C_1 \neq C_2$, $x \in C_1$, and $y \in C_2$, we have that $v(x, y) < 0$. Then, it holds that*

$$\frac{\mathbb{E}\left[\mathcal{SW}\left(ALG(G)\right)\right]}{\mathcal{SW}\left(\pi^*(G)\right)} \geq \frac{1}{3} - \frac{1}{n}.$$

We can combine the insights of Lemmas 5.4 and 5.5 to prove our theorem. Consider any FHG $G = (N, v)$ together with an optimal partition $\pi^*(G)$. We obtain a related instance $G' = (N, v')$ by making all valuations between different coalitions of $\pi^*(G)$ negative, i.e.,

$$v'(x, y) = \begin{cases} v(x, y) & \text{if } \exists C \in \pi^*(G) \text{ with } x, y \in C \text{ or } v(x, y) < 0, \\ -1 & \text{else.} \end{cases}$$

Note that G' differs from G only by decreasing valuations. Hence, Lemma 5.4 implies that $\mathbb{E}\left[\mathcal{SW}_G\left(ALG(G)\right)\right] \geq \mathbb{E}\left[\mathcal{SW}_{G'}\left(ALG(G')\right)\right]$. Moreover, for any partition π of N, it holds that

$$\mathcal{SW}_{G'}\left(\pi^*(G)\right) = \mathcal{SW}_G\left(\pi^*(G)\right) \geq \mathcal{SW}_G(\pi) \geq \mathcal{SW}_{G'}(\pi).$$

In the equality, we use that valuations in G and G' are identical within coalitions of $\pi^*(G)$. Hence, $\pi^*(G)$ is also maximizes welfare with respect to G'. Now note that G' satisfies the conditions of Lemma 5.5. Thus, we have that

$$\frac{\mathbb{E}\left[\mathcal{SW}_{G'}\left(ALG(G')\right)\right]}{\mathcal{SW}_{G'}\left(\pi^*(G)\right)} \geq \frac{1}{3} - \frac{1}{n}.$$

We conclude that

$$\frac{\mathbb{E}\left[\mathcal{SW}_G(ALG(G))\right]}{\mathcal{SW}_G(\pi^*(G))} \geq \frac{\mathbb{E}\left[\mathcal{SW}_{G'}\left(ALG(G')\right)\right]}{\mathcal{SW}_G\left(\pi^*(G)\right)} = \frac{\mathbb{E}\left[\mathcal{SW}_{G'}\left(ALG(G')\right)\right]}{\mathcal{SW}_{G'}\left(\pi^*(G)\right)} \geq \frac{1}{3} - \frac{1}{n}.$$

Since G was an arbitrary FHG, we conclude that *ALG* has the desired competitive ratio. □

We remark that Algorithm 1 is a matching algorithm because all returned coalitions are of size at most 2. Hence, Theorem 5.2 even implies a competitive ratio of $\frac{1}{3} - \frac{1}{n}$ in the matching domain. Thus, it presents an alternative algorithm for achieving Theorem 5.1.

As we show next, an asymptotic competitive ratio of $\frac{1}{3}$ as achieved in Theorems 5.1 and 5.2 is optimal in both the matching and coalition formation domain. In particular, this also means that the competitive ratio of $\frac{5}{12}$ in the matching domain as achieved by Ezra et al. [18] when the number of agents is known is off limits.

Since our proof is rather long, we discuss the main idea here and defer a formal proof to Appendix B.4. In essence, our construction relies on a careful interplay of two sets of instances whose positive edges form stars and bi-stars, i.e., a union of two stars whose centers are connected by an additional edge. This forces an algorithm to an undesired trade-off: The optimal matching in a star is to match the edge with the largest weight. In our bi-stars, the largest weight is the edge connecting the two centers, so the optimal matching contains exactly this edge. Now, by design of our instances, until both centers have arrived, an algorithm cannot distinguish whether its input is a star or a bi-star. The key step is to show that a competitive ratio of $\frac{1}{3}$ on a star can only be achieved if matching an edge with roughly a probability of at least $\frac{2}{3}$. However, as we will show, this means that when we are in a bi-star then the algorithm can only succeed with a probability of about $\frac{1}{3}$. This leads to a bound of the competitive ratio by $\frac{1}{3}$.

Theorem 5.6. *No randomized online matching algorithm has a competitive ratio under random arrival of more than $\frac{1}{3}$ on the tree domain.*

Combining Theorem 5.6 with Proposition 4.3, we conclude that no online coalition formation algorithm has a competitive ratio under random arrival of more than $\frac{1}{3}$.

Corollary 5.7. *No randomized online coalition formation algorithm has a competitive ratio under random arrival of more than $\frac{1}{3}$.*

6 FRACTIONAL HEDONIC GAMES UNDER COALITION DISSOLUTION

We first consider the setting where algorithms should perform well regardless of a fixed arrival order but where algorithms can dissolve coalitions. In this setting, there exists a deterministic online matching algorithm achieving a competitive ratio of $\frac{1}{3+2\sqrt{2}}$ [14, 29].[9] We can apply Theorem 4.1 to obtain an algorithmic guarantee for FHGs.

---

[9] McGregor [29] achieves this competitive ratio in the much related edge arrival model. Bullinger and Romen [14] showed that it is preserved in the vertex arrival model.

THEOREM 6.1. *There exists a deterministic online coalition formation algorithm operating under free dissolution with a competitive ratio of at least* $\frac{1}{6+4\sqrt{2}}$.

The algorithm mentioned above is optimal for the matching domain in the tree domain [4]. By Proposition 4.3, no deterministic online algorithm is better than $\frac{1}{3+2\sqrt{2}}$-competitive. We can, however, improve upon this result by proving a bound matching Theorem 6.1.

We illustrate here the main ideas for its proof and defer the full proof to Appendix C. The proof technique is similar to the proof by Badanidiyuru Varadaraja [4] in the matching domain. However, we construct an enhanced version of the adversarial instance, where the partitions produced by an algorithm continue to be matchings, but the partition with the highest welfare is better than the best matching by a factor of about 2. We remark that our construction only uses instances with rational valuations, even though we also exclude irrational competitive ratios higher than $\frac{1}{6+4\sqrt{2}}$.

THEOREM 6.2. *No deterministic online coalition formation algorithm operating under free dissolution has a competitive ratio of more than* $\frac{1}{6+4\sqrt{2}}$ *for symmetric FHGs.*

PROOF SKETCH. The crucial idea is to use an algorithm that allegedly beats a competitive ratio of $\frac{1}{6+4\sqrt{2}}$ to construct a sequence of real numbers $(x_i)_{i\in\mathbb{N}}$ with $x_1 = 1$, $x_i \geq 0$ for $i \geq 2$, and such that for all $i \in \mathbb{N}$, it holds that

$$x_i \geq \beta\left(x_{i+1} + \sum_{j=1}^{i+1} x_j\right) \quad (1)$$

where $\beta > \frac{1}{3+2\sqrt{2}}$. The proof of Theorem 2 by Badanidiyuru Varadaraja [4] for the case of $k = 2$ and $f = 0$ shows that such a sequence of numbers does not exist.

The adversarial instance is established in phases, and in each phase, we determine a new element of a sequence $(y_i)_{i\in\mathbb{N}}$ of rational numbers that satisfies an inequality of the type of Inequality (1).[10]

Throughout the execution of the instance, the algorithm can only maintain a single coalition with positive welfare of y_i containing exactly two agents, say $\{a_i, b_i\}$. We now illustrate a Phase i for some fixed $i \in \mathbb{N}$. A visualization is provided in Figure 1. All agents that newly appear have a mutual positive valuation with exactly one of a_i and b_i, a valuation of 0 for some other agents, and a high negative valuation for most agents, in particular for the other agent in $\{a_i, b_i\}$. The new agents form "star" coalitions with a_i and b_i. In the first part of a stage, we achieve a situation where stars with ℓ_i leaves have arrived for both endpoints, where all of their positive valuations are y_i. These are the leftmost stars attached to a_i and b_i in Figure 1.

Then, we let new star coalitions arrive while incrementing their positive valuations by a specifically tailored rational value ϵ_i in each step. Eventually, the algorithm has to dissolve $\{a_i, b_i\}$ and form a new coalition with one of these agents and a new agent of valuation $y_i + j^*\epsilon_i$ for some positive integer j^*. This has to happen as otherwise, edges of unbounded weight arrive, which would lead to an unbounded competitive ratio.

In the previous step, i.e., when agents with valuations of $y_i + k^*\epsilon_i$, where $k^* = j^* - 1$ were arriving, we had two "star" coalitions with a_i and b_i, which we now call C_i and D_i, respectively. Then, a version of Inequality (1) can be established with two differences: (1) instead of β, we have 2γ, where γ is the competitive ratio of our algorithm, and (2) there is an error term dependent on ϵ_i. For this, we compare y_i, i.e., the social welfare of $\{a_i, b_i\}$, with the social welfare of the partition containing D_i and C_j for $1 \leq j \leq i$, where the C_j evolve from earlier phases. Note that C_i and D_i have a welfare of about $2(y_i + j^*\epsilon_i)$.

A crucial idea is to control the error terms to be very small in sum by having ϵ_i decay exponentially for i tending to infinity, while the number of leaves ℓ_i grows as $\frac{1-\epsilon_i}{\epsilon_i}$. This allows to prove Inequality (1) for $\beta = \gamma + \frac{1}{6+4\sqrt{2}}$. □

7 CONCLUSION

We have studied two different models for online coalition formation in FHGs to maximize social welfare, a goal that does not allow for bounded competitive ratios in the standard model with an adversarial arrival order. Designing good online coalition formation algorithms is deeply related to designing good online matching algorithms. It is possible to leverage matching algorithms with a factor 2 in welfare loss, while limitations for matching algorithms can be preserved exactly if they hold on the tree domain.

Under random arrival, we have seen that a competitive ratio of $\frac{1}{3}$ is asymptotically optimal in both the matching and coalition formation domain. Hence, by providing a better analysis of a matching algorithm, we do not lose the factor of 2 in welfare. Moreover, in the coalition dissolution model, we determined that the optimal competitive ratio is $\frac{1}{6+4\sqrt{2}}$ as compared to the optimal competitive ratio of $\frac{1}{3+2\sqrt{2}}$ in the matching domain. For this, we constructed a new family of worst-case instances showing that the worst-case behavior on the coalition formation domain can be a factor of 2 worse than on the matching domain.

An intriguing question is whether forming coalitions larger than 2 can be beneficial. In fact, our paper reinforces the opposing view that matching algorithms exhibit (asymptotically) optimal performance. Thus, from an algorithmic perspective, larger coalitions are often unnecessary. In contrast, requiring algorithms to form larger coalitions can be problematic as such algorithms may fail to provide guarantees regarding approximate social welfare. For example, partitions that include a coalition of size at least three result in a negative welfare for instances on the tree domain. However, this depends on the presence of large negative valuations. In contrast, Flammini et al. [23, Theorems 4.4 and 4.5] present an algorithm that forms larger coalitions for FHGs with nonnegative valuations. Nevertheless, the competitive ratio they achieve depends on the range of the involved valuations. It would be interesting to explore whether this dependency can be eliminated under coalition dissolution or random arrival.

Another future direction is to consider the combination of random arrival and coalition dissolution. Note that our families of worst-case instances are specifically tailored to their setting and do not challenge algorithms in the respective other setting. For example, the algorithms developed by McGregor [29] and Bullinger and Romen [14] discussed in the beginning of Section 6 find the optimal partition on stars and double stars (i.e., the worst-case instance in the random arrival setting can be solved optimally under free

---

[10] It is easy to eventually transform this sequence to the exact desired form of $(x_i)_{i\in\mathbb{N}}$.

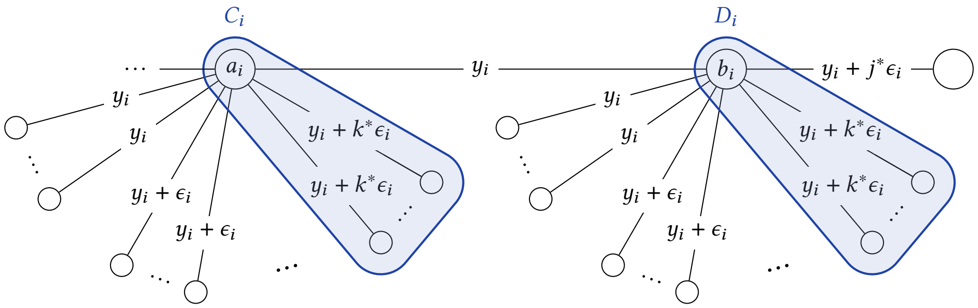

Figure 1: Illustration of Phase i in the construction of the adversarial instance in the proof of Theorem 6.2. Each star attached to a_i and b_i contains ℓ_i leaves.

dissolution, even if agents arrive in an arbitrary order). Hence, it could well be that algorithms can achieve higher competitive ratios in a combined setting.

ACKNOWLEDGMENTS

Most of this work was done when Martin Bullinger was at the University of Oxford. Martin Bullinger was supported by the AI Programme of The Alan Turing Institute. René Romen and Alexander Schlenga were supported by the Deutsche Forschungsgemeinschaft under grant BR 2312/11-2.

REFERENCES

[1] Haris Aziz, Florian Brandl, Felix Brandt, Paul Harrenstein, Martin Olsen, and Dominik Peters. 2019. Fractional Hedonic Games. *ACM Transactions on Economics and Computation* 7, 2 (2019), 1–29.

[2] Haris Aziz, Serge Gaspers, Joachim Gudmundsson, Julián Mestre, and Hanjo Täubig. 2015. Welfare Maximization in Fractional Hedonic Games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*. 461–467.

[3] Haris Aziz and Rahul Savani. 2016. Hedonic Games. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Chapter 15.

[4] Ashwinkumar Badanidiyuru Varadaraja. 2011. Buyback Problem - Approximate Matroid Intersection with Cancellation Costs. In *Proceedings of the 38th International Colloquium on Automata, Languages, and Programming (ICALP)*. 379–390.

[5] Vittorio Bilò, Angelo Fanelli, Michele Flammini, Gianpiero Monaco, and Luca Moscardelli. 2015. On the Price of Stability of Fractional Hedonic Games. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1239–1247.

[6] Vittorio Bilò, Angelo Fanelli, Michele Flammini, Gianpiero Monaco, and Luca Moscardelli. 2018. Nash Stable Outcomes in Fractional Hedonic Games: Existence, Efficiency and Computation. *Journal of Artificial Intelligence Research* 62 (2018), 315–371.

[7] Anna Bogomolnaia and Matthew O. Jackson. 2002. The Stability of Hedonic Coalition Structures. *Games and Economic Behavior* 38, 2 (2002), 201–230.

[8] Florian Brandl, Felix Brandt, and Martin Strobel. 2015. Fractional Hedonic Games: Individual and Group Stability. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1219–1227.

[9] Felix Brandt and Martin Bullinger. 2022. Finding and Recognizing Popular Coalition Structures. *Journal of Artificial Intelligence Research* 74 (2022), 569–626.

[10] Niv Buchbinder, Danny Segev, and Yevgeny Tkach. 2018. Online Algorithms for Maximum Cardinality Matching with Edge Arrivals. *Algorithmica* 81 (2018), 1781–1799.

[11] Martin Bullinger. 2020. Pareto-Optimality in Cardinal Hedonic Games. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 213–221.

[12] Martin Bullinger, Vaggos Chatziafratis, and Parnian Shahkar. 2025. Welfare Approximation in Additively Separable Hedonic Games. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 418–426.

[13] Martin Bullinger, Edith Elkind, and Jörg Rothe. 2024. Cooperative Game Theory. In *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Jörg Rothe (Ed.). Springer, Chapter 3, 139–229.

[14] Martin Bullinger and René Romen. 2025. Online Coalition Formation under Random Arrival or Coalition Dissolution. *ACM Transactions on Algorithms* 22, 1 (2025), 4:1–43.

[15] Martin Bullinger and René Romen. 2025. Stability in Online Coalition Formation. *Journal of Artificial Intelligence Research* 82 (2025), 2423–2452.

[16] José Correa, Paul Dütting, Felix Fischer, and Kevin Schewior. 2019. Prophet Inequalities for I.I.D. Random Variables from an Unknown Distribution. In *Proceedings of the 20th ACM Conference on Economics and Computation (ACM-EC)*. ACM Press, 3–17.

[17] Jacques H. Drèze and Joseph Greenberg. 1980. Hedonic Coalitions: Optimality and Stability. *Econometrica* 48, 4 (1980), 987–1003.

[18] Tomer Ezra, Michal Feldman, Nick Gravin, and Zhihao Gavin Tang. 2022. General Graphs are Easier than Bipartite Graphs: Tight Bounds for Secretary Matching. In *Proceedings of the 22nd ACM Conference on Economics and Computation (ACM-EC)*. 1148–1177.

[19] Jon Feldman, Nitish Korula, Vahab Mirrokni, Shanmugavelayutham Muthukrishnan, and Martin Pál. 2009. Online Ad Assignment with Free Disposal. In *Proceedings of the 5th International Conference on Web and Internet Economics (WINE)*. 374–385.

[20] Thomas S. Ferguson. 1989. Who solved the secretary problem? *Statist. Sci.* 4, 3 (1989), 282–296.

[21] Michele Flammini, Bojana Kodric, Gianpiero Monaco, and Qiang Zhang. 2021. Strategyproof Mechanisms for Additively Separable and Fractional Hedonic Games. *Journal of Artificial Intelligence Research* 70 (2021), 1253–1279.

[22] Michele Flammini, Bojana Kodric, and Giovanna Varricchio. 2022. Strategyproof mechanisms for friends and enemies games. *Artificial Intelligence* 302 (2022), 103610.

[23] Michele Flammini, Gianpiero Monaco, Luca Moscardelli, Mordechai Shalom, and Shmuel Zaks. 2021. On the Online Coalition Structure Generation Problem. *Journal of Artificial Intelligence Research* 72 (2021), 1215–1250.

[24] Buddhima Gamlath, Michael Kapralov, Andreas Maggiori, Ola Svensson, and David Wajc. 2019. Online Matching with General Arrivals. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS)*. 26–37.

[25] Zhiyi Huang, Zhihao Gavin Tang, and David Wajc. 2024. Online Matching: A Brief Survey. *ACM SIGecom Exchanges* 22, 1 (2024), 135–158.

[26] Christos Kaklamanis, Panagiotis Kanellopoulos, and Konstantinos Papaioannou. 2016. The Price of Stability of Simple Symmetric Fractional Hedonic Games. In *Proceedings of the 9th International Symposium on Algorithmic Game Theory (SAGT) (Lecture Notes in Computer Science (LNCS), Vol. 9928)*. Springer-Verlag, 220–232.

[27] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. 2013. An Optimal Online Algorithm for Weighted Bipartite Matching and Extensions to Combinatorial Auctions. In *Proceedings of the 21st European Symposium on Algorithms (ESA) (Lecture Notes in Computer Science (LNCS))*. Springer-Verlag, 589–600.

[28] Sweta Malik, Maeve Duffy, Subhasis Thakur, and John G. Breslin. 2021. Fractional Hedonic Coalition Formation Game for Peer to Peer Energy Trading in a Microgrid. In *Proceedings of the 2021 IEEE Madrid PowerTech*. IEEE Computer Society Press, 1–6.

[29] Andrew McGregor. 2005. Finding Graph Matchings in Data Streams. In *Proceedings of the 8th International Workshop on Approximation Algorithms for Compinatorial Optimization Problems and 9th International Workshop on Randomization and Computation (APPROX & RANDOM) (Lecture Notes in Computer Science (LNCS))*. 170–181.

[30] Debraj Ray. 2007. *A Game-Theoretic Perspective on Coalition Formation.* Oxford University Press.

[31] Alexander Schrijver. 2003. *Combinatorial Optimization - Polyhedra and Efficiency*. Springer.

APPENDIX

In the appendix, we present proofs and further material.

A PROOFS IN SECTION 4

In this section, we present the proofs missing in Section 4.

A.1 Simple proof of Theorem 4.1

In this section, we provide a new proof for Theorem 4.1 which was first proved by Flammini et al. [21, Theorem 1]. The proof by Flammini et al. [21] is based on a comparison of the maximum weight matching and the optimal partition by deriving connections of edges not contained in the matching. By contrast, we present a simple proof based on a folklore result about matchings that states that the maximum weight of a matching exceeds the weight of a "uniform fractional" matching where each edge is fractionally matched with probability $\frac{1}{n}$, if n is the number of vertices. We combine this with the idea that the social welfare of a coalition in an FHG is twice the weight of the "uniform fractional" matching.

Theorem 4.1. *[Flammini et al. [21]] Every MWM is a $\frac{1}{2}$-approximation of social welfare in symmetric FHGs.*

Proof. Assume that we are given an FHG $G = (N, v)$ and let π^* be a partition maximizing social welfare. Let $C \subseteq N$ be a coalition and $\mu^*(C)$ be a maximum weight matching on the subgraph of (N, v) induced by C. Recall that we denote the weight of a matching M by $v(M)$.

A folklore theorem in matching [see, e.g., 14, Lemma 1] says that

$$\frac{1}{|C|}\sum_{\{i,j\}\subseteq C, i\neq j} v(i,j) \leq v(\mu^*(C)). \tag{2}$$

We conclude that

$$\begin{aligned}\mathcal{SW}(\pi^*) &= \sum_{C\in\pi^*}\mathcal{SW}(C) = \sum_{C\in\pi^*}\sum_{i\in C}\sum_{j\in C\setminus\{i\}}\frac{v_i(j)}{|C|}\\ &= \sum_{C\in\pi^*}\frac{1}{|C|}\sum_{\{i,j\}\subseteq C, i\neq j} 2v(i,j)\\ &\overset{\text{Eq. (2)}}{\leq} \sum_{C\in\pi^*} 2v(\mu^*(C)) \leq 2v(\mu^*(N)) = 2\mathcal{SW}(\mu^*(N)).\end{aligned}$$

The second line uses that each valuation occurs twice in a symmetric game, once for each endpoint. The second-to-last line uses that $\bigcup_{C\in\pi^*}\mu^*(C)$ is a matching on N, and, therefore, its weight is bounded by the maximum weight matching of N. The last line uses that each nonsingleton coalition $C = \{i, j\}$ in $\mu^*(N)$ consists of two agents, i.e., $\mathcal{SW}(C) = u_i(C) + u_j(C) = \frac{1}{2}v(i,j) + \frac{1}{2}v(i,j) = v(i,j)$. □

A.2 Matching Limitations on the Tree Domain

In this section, we provide the proof for our result that lets us transfer negative results for online matching algorithms on the tree domain to the coalition formation domain.

Proposition 4.3. *Let $c \leq 1$ and assume that no c-competitive (randomized) algorithm exists for online MWM on the tree domain. Then, no c-competitive (randomized) online coalition formation algorithm exists for symmetric FHGs.*

Proof. We show a proof by contraposition. Assume a c-competitive online coalition formation algorithm *ALG* for symmetric FHGs exists. We construct a c-competitive algorithm *ALG′* on the tree domain that never forms a coalition of size three or more. To this end, let *ALG′* simulate *ALG*, i.e., whenever a new agent and her valuations are revealed to *ALG′*, it feeds the same input to *ALG*. Then, *ALG′* observes the output of *ALG*. If the new agent is in a coalition of size two with positive social welfare, then *ALG′* forms the same coalition. In all other cases, *ALG′* puts the new agent into a singleton coalition. Additionally, if *ALG* dissolves a coalition in the coalition dissolution setting, then *ALG′* also dissolves the matched pair from this coalition if necessary. In particular, *ALG′* only returns (random) matchings and, therefore, is a matching algorithm.

On the tree domain, *ALG′* achieves at least as high (expected) welfare as *ALG* because the large negative valuations make every coalition of size more than two have negative social welfare. Consequently, every coalition of size at least 3 achieves less welfare than when it was dissolved into singleton coalitions (or pairs of positive valuation). Thus, *ALG′* is c-competitive on the tree domain against all possible partitions and, therefore, in particular, against all matchings. □

B FURTHER MATERIAL FOR SECTION 5

In this section, we present further details for the random arrival setting.

B.1 Proof of Theorem 5.1

In this section, we consider the performance of the algorithm by Ezra et al. [18], when the number of agents is unknown and k is set to 3.

Theorem 5.1. *There exists a randomized online matching algorithm with a competitive ratio under random arrival of at least $\frac{1}{3} - O(\frac{1}{n})$.*

Proof. Consider Algorithm 1 as defined by Ezra et al. [18]. We refer to this algorithm as *ALG*. Note that the algorithm is parameterized by a positive integer k.

Consider an arbitrary FHG $G = (N, v)$. Let μ^* be a maximum weight matching and $ALG(G)$ be the matching computed by *ALG*. In the proof of their Theorem 3.1, Ezra et al. [18] obtain the following inequality for all $k \geq 3$:

$$\frac{\mathbb{E}\left[v(ALG(G))\right]}{v(\pi^*(G))} \geq \frac{1}{3} + \frac{k^2}{n^2} - \frac{4k^3}{3n^3} - O\left(\frac{1}{n}\right).$$

Setting $k = 3$, this implies that the competitive ratio of *ALG* is at least

$$\inf_G \frac{\mathbb{E}\left[v(ALG(G))\right]}{v(\pi^*(G))} \geq \frac{1}{3} - O\left(\frac{1}{n}\right).$$

□

B.2 Optimal Coalition Formation Algorithm

In this section, we provide the proofs of auxiliary lemmas in the proof of Theorem 5.2, restated as follows.

Theorem 5.2. *There exists a randomized online coalition formation algorithm with a competitive ratio under random arrival of at least* $\frac{1}{3} - \frac{1}{n}$.

The statement of Lemma 5.3 is similar to the statement of Lemma 3.2 by Ezra et al. [18]. While the proofs are quite different, they both rely on showing that the probability of being unavailable can be captured by the following recursive formula: For all $k, t \in \mathbb{N}$, $t \geq k+1$, define

$$p(k,k) := 0 \text{ and } p(k,t) := \frac{2}{t} + \frac{t-3}{t} \cdot p(k,t-1). \tag{3}$$

Ezra et al. [18] show that this formula resolves as captured in the following lemma, see the end of the proof of their Lemma 3.2.

Lemma B.1 (Ezra et al. [18]). *For* $k, t \in \mathbb{N}$ *with* $k \geq 3$ *and* $t \geq k$, *it holds that*

$$p(k,t) = \frac{2}{3}\left(1 - \frac{(t-3)! \cdot k!}{t! \cdot (k-3)!}\right).$$

Note that this implies that the probability of leaving a_t unmatched in Line 10 of Algorithm 1 is $1 - p(k,t-1)$. This already suggest of a relationship of the resolution of the recursion with the probability of availability. In fact, we now apply it to prove our first key lemma.

Lemma 5.3. *Assume that we run ALG for any FHG* (N,v) *and a random arrival order. For every* $k \geq 3$, $t \in \{k,\dots,n\}$, *every possible realization* $\tilde{N}$ *of* N_t *(i.e.,* $\tilde{N} \subseteq N$, $|\tilde{N}| = t$*), and every agent* $a \in \tilde{N}$, *it holds that*

$$\mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right) = \frac{2}{3}\left(1 - \frac{(t-3)! \cdot k!}{t! \cdot (k-3)!}\right).$$

Proof. Consider any realization $\tilde{N}$ of N_t and any agent $a \in \tilde{N}$. By Lemma B.1, it suffices to prove that

$$\mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right) = p(k,t). \tag{4}$$

We prove this by induction over t. Clearly, since the arrival of the first k agents does not alter the set A_t, we have that $N_k = A_t$ and, therefore, $\mathbb{P}\left[a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right] = 0 = p(k,k)$.

Now assume that $t > k$. We partition the event that a is unavailable (i.e., not in A_t) after the tth agent has been processed and given that $N_t = \tilde{N}$, into the following disjoint events:

(i) $a = a_t$,
(ii) $a = p_t$ and $a \neq a_t$, and
(iii) $a = x$ for some $x \in \tilde{N} \setminus e_t$.

Event (i) occurs with probability $\frac{1}{t}$. For (ii), we compute

$$\begin{aligned}
\mathbb{P}\left(a = p_t, a \neq a_t\right) &= \sum_{x \in \tilde{N} \setminus \{a\}} \mathbb{P}\left(a = p_t \mid x = a_t\right) \cdot \mathbb{P}\left(x = a_t\right) \\
&= \sum_{x \in \tilde{N} \setminus \{a\}} \zeta_t(x,a) \frac{1}{t} \\
&= \sum_{x \in \tilde{N} \setminus \{a\}} \zeta_t(a,x) \frac{1}{t} = \left(1 - \zeta_t(a,a)\right) \frac{1}{t}.
\end{aligned}$$

Hence, the probability for event (iii) is $1 - \frac{1}{t} - (1 - \zeta_t(a,a))\,\frac{1}{t} = \frac{t-2+\zeta_t(a,a)}{t}$. We get

$$\begin{aligned}
&\mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right) \\
&= \mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N} \wedge a = a_t\right) \frac{1}{t} \\
&+ \mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N} \wedge a \notin e_t\right) \frac{t-2+\zeta_t(a,a)}{t} \\
&+ \mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N} \wedge a = p_t \wedge a \neq a_t\right) \left(1 - \zeta_t(a,a)\right) \frac{1}{t}.
\end{aligned}$$

Assume that $a = a_t$. If $p_t \neq a_t$, then a_t is unavailable after being processed if and only if p_t was available at the arrival of a_t, i.e., after the $(t-1)$st agent was processed. By the induction hypothesis for $N_{t-1} = \tilde{N} \setminus \{a\}$, agent p_t was made unavailable with probability $p(k,t-1)$ and is, therefore, available with probability $1 - p(k,t-1)$. Moreover, if $p_t = a_t$, then a_t becomes unavailable if they are removed in line 10. By definition this happens with probability $\frac{1}{3} + \frac{2(t-4)!k!}{3(t-1)!(k-3)!} = 1 - p(k,t-1)$ as well. Hence, in any case, a becomes unavailable with probability $1 - p(k,t-1)$ after being processed.

If a is neither a_t nor p_t, then a is unavailable at the end of iteration t if they are unavailable at the end of iteration $t-1$. Furthermore, if they are available at the end of iteration $t-1$, they can be made unavailable in line 19 of the algorithm. This happens with probability $\frac{\zeta_t(a,a)}{t-2+\zeta_t(a,a)}$.Using the induction hypothesis, we compute that a is unavailable by time t with probability $p(k,t-1) + (1 - p(k,t-1))\frac{\zeta_t(a,a)}{t-2+\zeta_t(a,a)}$.

Finally, if $a = p_t$, then a is always unavailable at the end of iteration t. Indeed, if a was available during the iteration, it will be matched to a_t upon their arrival.

Putting it all together, we get

$$\begin{aligned}
&\mathbb{P}\left(a \in N_t \setminus A_t \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right) = \left(1 - p(k,t-1)\right)\frac{1}{t} + \\
&\left(p(k,t-1) + \left(1 - p(k,t-1)\right) \frac{\zeta_t(a,a)}{t-2+\zeta_t(a,a)}\right) \frac{t-2+\zeta_t(a,a)}{t} \\
&+ \left(1 - \zeta_t(a,a)\right) \frac{1}{t} \\
&= \left(1 - p(k,t-1)\right)\frac{1}{t} + \frac{1}{t}\Big(\left(t-2+\zeta_t(a,a)\right) p(k,t-1) \\
&+ \left(1 - p(k,t-1)\right) \zeta_t(a,a)\Big) + \left(1 - \zeta_t(a,a)\right) \frac{1}{t} \\
&= \frac{1}{t}\Big(1 - p(k,t-1) + \left(t-2+\zeta_t(a,a)\right) p(k,t-1) \\
&+ \left(1 - p(k,t-1)\right) \zeta_t(a,a) + 1 - \zeta_t(a,a)\Big) \\
&= \frac{1}{t}\Big(2 - p(k,t-1) + (t-2)\, p(k,t-1) + \zeta_t(a,a) p(k,t-1) \\
&+ \left(1 - p(k,t-1)\right) \zeta_t(a,a) - \zeta_t(a,a)\Big) \\
&= \frac{1}{t}\left(2 - p(k,t-1) + (t-2)\, p(k,t-1)\right) \\
&= \frac{2}{t} + \frac{t-3}{t} \cdot p(k,t-1) \\
&\overset{\text{Eq. (3)}}{=} p(k,t).
\end{aligned}$$

This concludes the proof of Equation (4) and thus the entire proof.

□

Next, we prove the lemma concerning the monotonicity property when running Algorithm 1.

Lemma 5.4. *Let $G = (N, v)$, and $G' = (N, v')$ be two symmetric FHGs with $v(i, j) \geq v'(i, j)$ for all $i, j \in N$. Then,*

$$\mathbb{E}\left[\mathcal{SW}_G\left(ALG(G)\right)\right] \geq \mathbb{E}\left[\mathcal{SW}_{G'}\left(ALG(G')\right)\right].$$

Proof. Let $t \geq k + 1$. Consider any subset $\tilde{N} \subseteq N$ that realizes as N_t and an agent $a \in \tilde{N}$. We set $v(e_t) = 0$ if $e_t = \{a_t\}$, i.e., e_t consists only of a singleton. We then have

$$\begin{aligned}
&\mathbb{E}\left[v(e_t) \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right] \\
&= \sum_{x \in \tilde{N}} \mathbb{E}\left[v(e_t) \mid N_t = \tilde{N} \wedge a \in \tilde{N} \wedge a_t = x\right] \\
&\cdot \mathbb{P}\left(a_t = x \mid N_t = \tilde{N} \wedge a \in \tilde{N}\right) \\
&= \sum_{x \in \tilde{N}} \mathbb{E}\left[v(e_t) \mid N_t = \tilde{N} \wedge a \in \tilde{N} \wedge x = a_t\right] \cdot \frac{1}{t} \\
&= \frac{1}{t} \sum_{x \in \tilde{N}} \sum_{x' \in \tilde{N}} \mathbb{E}\left[v(e_t) \mid N_t = \tilde{N} \wedge a \in \tilde{N} \wedge x = a_t \wedge x' = p_t\right] \\
&\cdot \mathbb{P}\left(x' = p_t \mid N_t = \tilde{N} \wedge a \in \tilde{N} \wedge x = a_t\right) \\
&= \frac{1}{t} \sum_{x \in \tilde{N}} \sum_{x' \in \tilde{N}} v(x, x') \cdot \zeta_t(x, x') \\
&= \frac{2}{t} \mathbb{E}\left[v(\zeta_t)\right]
\end{aligned}$$

In the first and third equation, we use the law of total probability. The last equality follows because with the two summations we count every pair of vertices twice.

Next, we consider the expected competitive ratio. It holds

$$\mathbb{E}\left[\mathcal{SW}_G\left(ALG(G)\right)\right] = \sum_{t=k+1}^{n} \mathbb{E}\Big[\chi\left[p_t \in A_{t-1}\right] \cdot v(e_t)\Big]$$

$$= \sum_{t=k+1}^{n} \mathbb{E}\Big[\chi\left[p_t \in A_{t-1}\right]\Big] \cdot \mathbb{E}\left[v(e_t)\right] = \sum_{t=k+1}^{n} \mathbb{P}\left(p_t \in A_{t-1}\right) \cdot \mathbb{E}\left[v(e_t)\right]$$

We use that the random variables $\chi\left[p_t \in N_t \cap A_t\right]$ and $v(e_t)$ are independent of each other. Indeed, the MWFM computed in line 6 of Algorithm 1 was chosen independent of A (which at this points was equal to A_{t-1}).

Moreover, by Lemma 5.3, we know that

$$\mathbb{P}\left(p_t \in A_{t-1}\right) = 1 - p(k, t-1).$$

Hence,

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{SW}_G\left(ALG(G)\right)\right] \\
&= \sum_{t=k+1}^{n} \mathbb{P}\left(p_t \in A_{t-1}\right) \cdot \mathbb{E}\left[v(e_t)\right] \\
&= \sum_{t=k+1}^{n} \left(1 - p(k, t-1)\right) \frac{2}{t} \mathbb{E}\left[v(\zeta_t)\right] \\
&\geq \sum_{t=k+1}^{n} \left(1 - p(k, t-1)\right) \frac{2}{t} \mathbb{E}\left[v(\zeta'_t)\right] \\
&\geq \sum_{t=k+1}^{n} \left(1 - p(k, t-1)\right) \frac{2}{t} \mathbb{E}\left[v'(\zeta'_t)\right] \\
&= \sum_{t=k+1}^{n} \mathbb{P}\left(p_t \in A_{t-1}\right) \cdot \mathbb{E}\left[v'(e'_t)\right] = \mathbb{E}\left[\mathcal{SW}_{G'}\left(ALG(G')\right)\right].
\end{aligned}$$

There, ζ'_t and e'_t denote the constructed MWFM and edges when executing Algorithm 1 for G'. In the first inequality, we use that ζ_t is an MWFM for $G[N_t]$ and hence achieve a higher weight in G than ζ'_t. In the second inequality, we use that $v(x, y) \geq v'(x, y)$ for all agents $x, y \in N$. This completes the proof of the lemma. □

Our next lemma shows that the desired performance is achieved on a special set of instances where valuations between coalitions of some optimal partition are negative.

Lemma 5.5. *Assume that we run ALG for $k = 3$. Consider an FHG G together with an optimal (offline) partition $\pi^*(G)$. Assume that for all $C_1, C_2 \in \pi^*(G)$ with $C_1 \neq C_2$, $x \in C_1$, and $y \in C_2$, we have that $v(x, y) < 0$. Then, it holds that*

$$\frac{\mathbb{E}\left[\mathcal{SW}\left(ALG(G)\right)\right]}{\mathcal{SW}\left(\pi^*(G)\right)} \geq \frac{1}{3} - \frac{1}{n}.$$

Proof. Let $G = (N, v)$ be an FHG that satisfies the properties of the statement of the lemma together with a uniformly random arrival order σ.

For $t \leq 3$, we set $e_t = \{a_t\}$. Moreover, as in the proof of Lemma 5.4, we set $v(e_t) = 0$ if $e_t = \{a_t\}$. Now, we know from the proof of Lemma 5.4 that

$$\begin{aligned}
\mathbb{E}\left[\mathcal{SW}\left(ALG(G)\right)\right] &= \sum_{t=4}^{n} \mathbb{P}\left(p_t \in A_{t-1}\right) \cdot \mathbb{E}\left[v(e_t)\right] \\
&= \sum_{t=4}^{n} \left(1 - p(3, t-1)\right) \cdot \mathbb{E}\left[v(e_t)\right].
\end{aligned}$$

We observe that for all $t \geq 4$, it holds that

$$1 - p(3, t-1) = 1 - \frac{2}{3}\left(1 - \frac{(t-4)! \cdot 6}{(t-1)!}\right) > \frac{1}{3}.$$

Consequently,

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{SW}\left(ALG(G)\right)\right] > \frac{1}{3} \sum_{t=4}^{n} \mathbb{E}\left[v(e_t)\right] \\
&= \frac{1}{3} \sum_{\substack{x \in N \\ \sigma^{-1}(x) \geq 4}} \mathbb{E}\left[v(e_{\sigma^{-1}(x)})\right] \\
&= \frac{1}{3} \sum_{\substack{x \in N \\ \sigma^{-1}(x) \geq 4}} \mathbb{E}\left[v(e_{\sigma^{-1}(x)}) \mid \sigma^{-1}(x) \geq 4\right] \\
&= \frac{1}{3} \sum_{x \in N} \mathbb{E}\left[v(e_{\sigma^{-1}(x)}) \mid \sigma^{-1}(x) \geq 4\right] \cdot \mathbb{P}\left(\sigma^{-1}(x) \geq 4\right) \\
&= \frac{1}{3} \sum_{x \in N} \mathbb{E}\left[v(e_{\sigma^{-1}(x)}) \mid \sigma^{-1}(x) \geq 4\right] \cdot \frac{n-3}{n} \\
&= \frac{n-3}{3n} \sum_{C \in \pi^*} \sum_{x \in C} \mathbb{E}\left[v(e_{\sigma^{-1}(x)}) \mid \sigma^{-1}(x) \geq 4\right] \\
&= \frac{n-3}{3n} \sum_{C \in \pi^*} \sum_{x \in C} \mathbb{E}\left[v(e_{\sigma^{-1}(x)}) \mid N_{\sigma^{-1}(x)-1} \cap C \neq \emptyset \wedge \sigma^{-1}(x) \geq 4\right] \\
&\cdot \mathbb{P}\left(N_{\sigma^{-1}(x)-1} \cap C \neq \emptyset \mid \sigma^{-1}(x) \geq 4\right)
\end{aligned}$$

$$\geq \frac{n-3}{3n} \sum_{C\in\pi^*}\sum_{x\in C} \mathbb{E}\left[v(e_{\sigma^{-1}(x)}) \mid N_{\sigma^{-1}(x)-1}\cap C \neq \emptyset \wedge \sigma^{-1}(x) \geq 4\right]$$

$$\cdot \frac{|C|-1}{|C|}$$

$$= \frac{n-3}{3n}\sum_{C\in\pi^*}\sum_{t\in[n]:\sigma(t)\in C} \mathbb{E}\left[v(e_t) \mid N_{t-1}\cap C\neq\emptyset \wedge t\geq 4\right]\cdot\frac{|C|-1}{|C|}.$$

For the inequality at second-to-last line, observe that $\mathbb{P}\left(N_{\sigma^{-1}(x)-1}\cap C\neq\emptyset\right) = \frac{|C|-1}{|C|}$ because every agent in C does not appear as the first agent among the agents in C with that probability. Moreover, as having agents present already can only increase the probability, this quantity can only become larger when conditioning on $\sigma^{-1}(x)\geq 4$.

Let $t\in[n]$ with $\sigma(t)\in C$. We now take a closer look at $\mathbb{E}\left[v(e_t)\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\right]$. Essentially, this is the expected weight of the candidate edge if the arriving agent $\sigma(t)=a_t$ is not the first of $\pi^*(a_t)$ to arrive. We define the fractional matching $\zeta_{C\cap N_t}$ on $G[C\cap N_t]$ by setting

$$\zeta_{C\cap N_t}(x,x') = \zeta_t(x,x') \tag{5}$$

for all $x,x'\in N_t$ with $x,y\in C\in\pi^*$.

By our assumption on the structure of the instance, ζ_t cannot assign positive weight on edges between agents from different coalitions in π^*. Hence, for all $t\geq 4$, $\zeta_{C\cap N_t}$ is a well-defined fractional matching on $G[C\cap N]$ and we have

$$v(\zeta_t) = \sum_{C\in\pi^*} v(\zeta_{C\cap N_t}).$$

Recall that $t\in[n]$ with $\sigma(t)\in C$. Now, similar as in the proof of Lemma 5.4, only restricted to a single coalition, we compute:

$$\mathbb{E}\left[v(e_t)\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\right]$$

$$= \sum_{x\in N_t\cap C}\mathbb{E}\left[v(e_t)\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\wedge a_t = x\right] \tag{6}$$

$$\cdot\,\mathbb{P}\left(a_t = x\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\right)$$

$$= \sum_{x\in N_t\cap C}\mathbb{E}\left[v(e_t)\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\wedge x = a_t\right]\cdot\frac{1}{|N_t\cap C|}$$

$$= \frac{1}{|N_t\cap C|}\sum_{x\in N_t\cap C}\sum_{x'\in N_t\cap C\setminus\{x\}} \tag{7}$$

$$\mathbb{E}\left[v(e_t)\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\wedge x = a_t\wedge x' = p_t\right]$$

$$\cdot\,\mathbb{P}\left(x' = p_t\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\wedge x = a_t\right)$$

$$= \frac{1}{|N_t\cap C|}\sum_{x\in N_t\cap C}\sum_{x'\in N_t\cap C\setminus\{x\}} v(x,x')\cdot\zeta_t(x,x')$$

$$\overset{\text{Eq. (5)}}{=} \frac{1}{|N_t\cap C|}\sum_{x\in N_t\cap C}\sum_{x'\in N_t\cap C\setminus\{x\}} v(x,x')\cdot\zeta_{C\cap N_t}(x,x')$$

$$= \frac{2}{|C\cap N_t|}\mathbb{E}\left[v(\zeta_{C\cap N_t})\right]. \tag{8}$$

Moreover, we know that in general, for any subgraph of G induced by agent set $W\subseteq N$, an MWFM ζ_W on $G[W]$ is at least as good as matching all edges with equal fractions, i.e.,

$$2\cdot v(\zeta_W)\geq\sum_{x\in W}\sum_{x'\in W\setminus\{x\}}\frac{1}{|W|-1}v(x,x') \tag{9}$$

For a subset $W\subseteq N$, let $U(W)$ denote the uniform distribution over ordered pairs of distinct agents in W, i.e., edges in the subgraph induced by W. Combining the previous insights, we get

$$\mathbb{E}\left[v(e_t)\mid N_{t-1}\cap C\neq\emptyset\wedge t\geq 4\right]\overset{\text{Eq. (6)}}{=}\frac{2}{|C\cap N_t|}\mathbb{E}\left[v(\zeta_{C\cap N_t})\right]$$

$$\overset{\text{Eq. (9)}}{\geq}\frac{1}{|C\cap N_t|}\mathbb{E}\left[\sum_{x\in C\cap N_t}\sum_{x'\in C\cap N_t\setminus\{x\}}\frac{1}{|C\cap N_t|-1}v(x,x')\right]$$

$$= \frac{1}{|C\cap N_t|}\frac{(|C\cap N_t|-1)|C\cap N_t|}{|C\cap N_t|-1}\mathbb{E}_{e\sim U(C)}\left[v(e)\right]$$

$$= \mathbb{E}_{e\sim U(C)}\left[v(e)\right].$$

There, the second-to-last equality holds because $C\cap N_t$ is a subset of C drawn uniformly at random, and hence drawing a uniformly random valuation from an ordered pair in $C\cap N_t$ is equivalent to drawing a random valuation from an ordered pair in C directly. All in all,

$$\mathbb{E}\left[\mathcal{SW}\left(ALG(G)\right)\right]$$

$$\geq\frac{n-3}{3n}\sum_{C\in\pi^*}\sum_{t\in[n]:\sigma(t)\in C}\mathbb{E}\left[v(e_t)\mid N_{t-1}\cap C\neq\emptyset\wedge t>3\right]$$

$$\cdot\frac{|C|-1}{|C|}$$

$$\geq\frac{n-3}{3n}\sum_{C\in\pi^*}\sum_{t\in[n]:\sigma(t)\in C}\mathbb{E}_{e\sim U(C)}\left[v(e)\right]\cdot\frac{|C|-1}{|C|}$$

$$= \frac{n-3}{3n}\sum_{C\in\pi^*}|C|\cdot\mathbb{E}_{e\sim U(C)}\left[v(e)\right]\cdot\frac{|C|-1}{|C|}$$

$$= \frac{n-3}{3n}\sum_{C\in\pi^*}\mathbb{E}_{e\sim U(C)}\left[v(e)\right]\cdot(|C|-1)$$

$$= \frac{n-3}{3n}\sum_{C\in\pi^*}\mathcal{SW}\left(C\right)$$

$$= \frac{n-3}{3n}\mathcal{SW}\left(\pi^*\right).$$

This proves the claimed competitive ratio:

$$\frac{\mathbb{E}\left[\mathcal{SW}\left(ALG(G)\right)\right]}{\mathcal{SW}\left(\pi^*\right)}\geq\frac{n-3}{3n}=\frac{1}{3}-\frac{1}{n}$$

□

B.3 Limitations of Algorithm Based on Integral Matchings

In this section, we consider Algorithm 1 for the case where we modify line 6 to select an MWM instead of an MWFM. We select an MWM that only depends on the set of present agents by applying minuscule perturbations of the valuations that act as a tie-breaking mechanism among integral matchings. This is essentially analogous to the online matching algorithm by Ezra et al. [18, Algorithm 1] for the matching setting when the number of agents is known.[11] However, another difference is that they want the maximum weight matchings to be perfect and, therefore, need an even number of vertices. They achieve this by deleting a uniformly random vertex whenever the number of vertices is odd. We do not face this issue

---

[11] Our next result also holds for the version of their algorithm for the case when $k=3$, which is optimal for online matching with an unknown number of agents, cf. Theorem 5.1.

as we deal with the case of being unmatched by making vertices unavailable based on the probabilities $\zeta(x, x)$ for an agent x and a fractional matching ζ.

Proposition B.2. *Let ALG be Algorithm 1 where we modify line 6 to select an MWM as described above. ALG has a competitive ratio under random arrival of at most $\frac{5}{18} < \frac{1}{3}$, even for simple symmetric FHGs.*

Proof. Let ALG be the algorithm from the statement of the proposition. Let $k \in \mathbb{N}$. We define an FHG $G_k = (N, v)$ with $n = 3k$ agents. Let $N = \{a_i, b_i, c_i \colon i \in [k]\}$. The valuations are given as $v(a_i, b_i) = v(a_i, c_i) = v(b_i, c_i) = 1$ for all $i \in [k]$ and all other valuations, i.e., the valuations across triplets of agents, are set to 0. Hence, the underlying graph is a disjoint union of triangle graphs. Clearly, this constitutes a simple symmetric FHG.

Let now $i \in [k]$ be arbitrarily fixed, i.e., we look at a single triangle. Assume that agent a_i arrives at time t_a, agent b_i arrives at time t_b, and c_i arrives at time t_c. Without loss of generality, assume that $t_a < t_b < t_c$. In the algorithm, the edge weights are randomly perturbed to always guarantee a unique MWM. When a_i arrives, the candidate edge e_{t_a} which may be added by the algorithm will have weight 0 (or a_i is left unmatched). When b_i arrives, the candidate edge e_{t_b} will be the edge $\{a_i, b_i\}$ and hence have weight 1. When c_i arrives, the random perturbation leads to a $\frac{1}{3}$ chance the candidate edge e_{t_c} will be $\{a_i, c_i\}$, a $\frac{1}{3}$ chance it will be $\{b_i, c_i\}$, and a $\frac{1}{3}$ chance it will have weight 0 (or that the algorithm lets c_i unmatched in an MWM) because $\{a_i, b_i\}$ is the heaviest of the three edges in the triangle.

Moreover, whether an edge is included also depends on whether the other agent in this edge is available. This probability is given as in the formula of Lemma 5.3.[12] Hence, in the limit when k (and, therefore, n) tends to infinity, the partner of the candidate edge is available with probability $\frac{1}{3}$. Thus, in the limit, the expected social welfare of the algorithm's output obtained by a_i, b_i, and c_i is

$$\frac{1}{3}\left(0 + 1 + \frac{2}{3} \cdot 1\right) = \frac{5}{9}.$$

Hence, we have that $\lim_{k\to\infty} \mathbb{E}(ALG(G_k)) = \frac{5}{9}k$.

However, we have that $\pi^*(G_k) = \{\{a_i, b_i, c_i\} \colon i \in [k]\}$ with $\mathcal{SW}(\pi^*(G_k)) = 2k$. Hence,

$$c_{ALG} \leq \lim_{k\to\infty} \frac{\mathbb{E}(ALG(G_k))}{\mathcal{SW}(\pi^*(G_k))} = \frac{\frac{5}{9}k}{2k} = \frac{5}{18}.$$

This completes the proof. □

The reason why Algorithm 1 does not run into the same issues is that selecting an MWFM allows to match each of the three edges within a triangle with probability $\frac{1}{2}$. Hence, the social welfare achieved by the third agent in a triangle is 1 and not $\frac{2}{3}$, leading to a higher welfare. Moreover, note that the instances constructed in Proposition B.2 also show that Algorithm 1 can be at most $\frac{1}{3}$-competitive, i.e., they determine its asymptotic competitive ratio. This already hints at the much stronger result of Corollary 5.7.

B.4 Proof of Theorem 5.6

In this section, we provide the proof of Theorem 5.6. To make it accessible more quickly, we defer the proofs of intermediary lemmas to Appendix B.5.

The first part of our proof concerns showing that a good competitive ratio on a star is essentially equivalent to matching the maximum weight edge with a high probability. This is similar to the conversion of the problem from a cardinal to an ordinal setting as performed by Ezra et al. [18]. We then want bounds for the probability of matching any edge in a star instance, only dependent on the already arrived vertices. While Ezra et al. [18], inspired by Correa et al. [16], carry out such a step by applying an infinite version of Ramsey's theorem, we perform a direct computation of the probabilities using induction. Still, our proof is quite different from both of these as we have the interplay of two qualitatively different sets of adversarial instances.

Theorem 5.6. *No randomized online matching algorithm has a competitive ratio under random arrival of more than $\frac{1}{3}$ on the tree domain.*

Proof. In the following proof, we assume that all algorithms are randomized and operate under random arrival. Assume for contradiction that there is an online matching algorithm on the tree domain with a competitive ratio of $\hat{c} > \frac{1}{3}$. Without loss of generality, we can assume that $\hat{c}$ is rational. Otherwise, we replace it with any rational number in the open interval $(\frac{1}{3}, \hat{c})$. Define $\epsilon := \frac{1}{3}\left(\hat{c} - \frac{1}{3}\right)$. Note that $\epsilon > 0$ and ϵ is rational.

Let $I, J \subseteq \mathbb{N}$ with $|I|, |J| < \infty$, $I \cap J = \emptyset$ and $I \neq \emptyset$, i.e., they are finite and disjoint, and I is nonempty. We design a family of instances with $n = 2 + |I| + |J|$ agents based on two symmetric valuation functions, one for stars and one for bi-stars, dependent on I, J. Additionally, the instance depends on a value for weights of negative edges, parameterized by x. Given such I and J, we define $t_B := \max(I \cup J)$, i.e., t_B is the largest number in $I \cup J$. We arbitrarily select an integer $x > t_B + 2$. Let $N = \{a, b\} \cup \{d_i \colon i \in I\} \cup \{d_j \colon j \in J\}$ be the set of agents.

First, we define a *star instance* $S^x_{I,J}$ by setting the following symmetric valuations:[13] For all $i \in I$, we set $v(a, d_i) = \left(\frac{1}{\epsilon}\right)^i$. All remaining valuations are set to $-\left(\frac{1}{\epsilon}\right)^x$. We set $t_S := \max I$, i.e., the edge of maximum weight is $\{a, d_{t_S}\}$ with a weight of $\left(\frac{1}{\epsilon}\right)^{t_S}$. Note that $t_S > 0$ as $I \neq \emptyset$.

Moreover, we define a *bi-star instance* $B^x_{I,J}$ with the following symmetric valuations: Recall that $t_B = \max(I \cup J)$. For all $i \in I$ and $j \in J$, we set $v(a, d_i) = \left(\frac{1}{\epsilon}\right)^i$ and $v(b, d_j) = \left(\frac{1}{\epsilon}\right)^j$. We set $v(a, b) = \left(\frac{1}{\epsilon}\right)^{t_B+1}$. Finally, all remaining valuations are set to $-\left(\frac{1}{\epsilon}\right)^x$. Note that the pair $\{a, b\}$ has the highest valuation of $\left(\frac{1}{\epsilon}\right)^{t_B+1}$. Note that, since ϵ is rational, all valuations in star and bi-star instances are rational.

Hence, given the same set of parameters, a star and bi-star instance only differ with respect to the valuations of b with a and agents in $\{d_j \colon j \in J\}$. We denote the set of all star instances with any permissible parameter combination of I, J, x, and ϵ as $\mathcal{S}$. Similarly, we denote the set of all bi-star instances as $\mathcal{B}$.

Note that the algorithm can only distinguish star and bi-star instances once a and b have arrived in a bi-star instance. In fact,

---

[12] The proof of the lemma works when replacing the MWFM by an MWM in every occurrence.

[13] We omit references to parameters from the names of the valuation functions to avoid overloading notation.

once a has arrived in a star instance, or one of a and b has arrived in a bi-star instance, an algorithm sees the star with one of these agents. However, all other agents, and in particular b if we are in a star instance, are only connected by large constant negative valuations and are indistinguishable. Furthermore, the optimal matching for star instances matches $\{a, d_{t_S}\}$ and leaves all other agents alone with a social welfare of $\left(\frac{1}{\epsilon}\right)^{t_S}$. Similarly, in bi-stars, the optimal matching matches $\{a, b\}$ and leaves all other agents as singletons with a social welfare of $\left(\frac{1}{\epsilon}\right)^{t_B+1}$.

Additionally, by the choice of x, both types of instances belong to the tree domain. Indeed, positive valuations are $\left(\frac{1}{\epsilon}\right)^{i}$ for some $i \leq x - 2$ and occur at most once each. Moreover, as $\hat{c} \in (\frac{1}{3}, 1)$, it holds that $\epsilon = \frac{1}{3}(\hat{c} - \frac{1}{3}) \leq \frac{1}{2}$. Hence, we have that the sum of positive valuations is at most $\sum_{i=1}^{x-2} \left(\frac{1}{\epsilon}\right)^{i} \leq \left(\frac{1}{\epsilon}\right)^{x-1} < \left(\frac{1}{\epsilon}\right)^{x}$.

Given a fixed algorithm, we want to find a relationship between its competitive ratio and the probability of matching the highest edge in star and bi-star instances. We say that an algorithm is *c-competitive for matching the maximum weight edge* if it matches the maximum weight edge with probability at least c in star and bi-star instances. We obtain the following relationship. Its proof relies on a separate analysis of stars and bi-stars.

Lemma B.3. *If there exists a $\hat{c}$-competitive online matching algorithm on the tree domain, then there exists an algorithm for matching the maximum weight edge with a competitive ratio of more than $\frac{1}{3}$.*

Hence, to derive a contradiction to our initial assumption of a $\hat{c}$-competitive online matching algorithm, we prove in the following the nonexistence of algorithms for matching the maximum weight edge with probability more than $\frac{1}{3}$. In the following steps, we want to achieve certain conditions under which our algorithms operate without loss of generality. This is similar to the reduction by Ezra et al. [18] to an "ordinal" setting. As a first step, we observe that we can restrict attention to algorithms that, if at all, match the current maximum weight edge in each step.

Lemma B.4. *For every star instance, we may assume without loss of generality that only the current maximum weight edge and no negative weight edges are matched.*

Proof. Consider an algorithm *ALG* for matching the maximum weight edge. We modify this algorithm such that whenever it performs a randomized decision to match an edge, it sets probabilities to 0 for matching edges that are not currently the maximum weight edge or have negative weight. It then continues executing *ALG* as if the decision of *ALG* had been performed. This algorithm has the desired form, i.e., it only matches the current maximum weight edge and no negative weight edges. Moreover, since negative weight edges and edges that are not currently the maximum weight are never the maximum weight edge in star and bi-star instances, the modified algorithm matches the maximum weight edge with the same probability. □

Consequently, we can restrict attention to algorithms that, at each step, face the decision to match the current maximum weight edge, if possible, or do nothing. From now on, we will only consider such algorithms.

We go one step further and show that when a matching decision is performed (to match a current maximum weight edge), this can be assumed to be independent of how the current state is achieved.

Lemma B.5. *For every star instance, we may assume without loss of generality that our algorithm's decisions only depend on*

- *which agents have arrived,*
- *whether a has arrived and is matched, and*
- *whether the last arrived agent is part of the current maximum weight edge.*

From now on, we consider algorithms as per Lemma B.5. Finally, we show that algorithmic decisions can be made independently of b and agents associated with J.

Lemma B.6. *For every star instance, we may assume without loss of generality that our algorithms decisions are independent of agents b and agents associated with J.*

From now on, we consider algorithms that, additionally, fulfill the independence of decisions of b and agents associated with J.

The combination of Lemmas B.5 and B.6 implies that an algorithm is fully specified by the matching probabilities dependent on the observed weights but not the arrival orders. From now on, we consider a fixed algorithm ALG^* and assume for contradiction that it is c^*-competitive for matching the maximum weight edge with $c^* > \frac{1}{3}$. It is fully specified by a function $f\colon 2^{\mathbb{N}} \times \mathbb{N} \longrightarrow [0, 1]$, where f takes as input a subset $I \subseteq \mathbb{N}$ (specifying the leaf weights in a star instance) and a positive integer x (specifying the parameter for negative edges). The value $f(I, x)$ equals the probability of matching the current maximum weight edge provided that a has arrived, is unmatched, the last arrived agent is part of the maximum edge, a has revealed edges precisely to agents corresponding to the set I, and x is the parameter for negative edges.

Now, consider a star instance $S \in \mathcal{S}$ based on parameters I, J, and x (at this point, ϵ is irrelevant). We define

$$h(S) := \mathbb{P}\left(\{a, d_i\} \in ALG^*(S) \text{ for some } i \in I\right),$$

i.e., the probability to match a. The key step is to estimate this quantity.

Lemma B.7. *Let $S \in \mathcal{S}$ with $|I| = k - 1$. Then it holds that $h(S) > \frac{2}{3} - \frac{2}{3k}$ for all $S \in \mathcal{S}$.*

Finally, we want to use the performance on stars to bound the performance on bi-stars. We essentially use that the prefix of every arrival order in every bi-star is indistinguishable from a star instance until both a and b arrive.

Consider a bi-star instance $B \in \mathcal{B}$ defined by I, J, and x (x defines its negative weights), and assume that $|I| = |J|$. As usual, the number of agents is n, i.e., $n = 2 + |I| + |J|$. Let Y be the random variable that counts the number of agents from I that arrive before b if b arrives after a and the number of agents from J that arrive before a if a arrives after b. Moreover, let Y_I be the random variable that counts the number of agents from I that arrive before b and Y_J be the random variable that counts the number of agents from J that arrive before a.

We compute

$$\begin{aligned}
&\mathbb{P}(\{a, b\} \in ALG^*(B) \mid Y \geq y) \\
&= \mathbb{P}(\{a, b\} \in ALG^*(B) \mid Y \geq y, \sigma^{-1}(a) < \sigma^{-1}(b)) \\
&\cdot \mathbb{P}(\sigma^{-1}(a) < \sigma^{-1}(b) \mid Y \geq y)
\end{aligned}$$

$$\begin{aligned}
&+\mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y \geq y, \sigma^{-1}(b) < \sigma^{-1}(a))\\
&\cdot \mathbb{P}(\sigma^{-1}(b) < \sigma^{-1}(a) \mid Y \geq y)\\
&= \frac{1}{2} \cdot \mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y_I \geq y, \sigma^{-1}(a) < \sigma^{-1}(b))\\
&\quad + \frac{1}{2} \cdot \mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y_J \geq y, \sigma^{-1}(b) < \sigma^{-1}(a))\\
&= \mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y_I \geq y, \sigma^{-1}(a) < \sigma^{-1}(b))
\end{aligned}$$

In the last step, we use symmetry between a and b together with I and J, which works because $|I| = |J|$. We thus want to estimate the latter probability.

Note that if a arrives before b, then the agents arriving before b form a star instance where the subset of agents of I that has arrived is a uniformly random subset of size Y_I. Hence, by Lemma B.7, we have that

$$\mathbb{P}(a \text{ matched when } b \text{ arrives} \mid Y_I \geq y, \sigma^{-1}(a) < \sigma^{-1}(b)) > \frac{2}{3} - \frac{2}{3(y+1)}.$$

There, we bound with the worst case where $Y_I = y$, i.e., $k = y + 1$ in Lemma B.7. It follows that

$$\begin{aligned}
&\mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y_I \geq y, \sigma^{-1}(a) < \sigma^{-1}(b))\\
&\leq 1 - \mathbb{P}(a \text{ matched when } b \text{ arrives} \mid Y_I \geq y, \sigma^{-1}(a)\\
&< \sigma^{-1}(b)) < \frac{1}{3} + \frac{2}{3(y+1)}.
\end{aligned}$$

Clearly, there exists $N \in \mathbb{N}$ such that for all $y \geq N$, it holds that $\frac{2}{3(y+1)} \leq \frac{1}{3}\left(c^* - \frac{1}{3}\right)$. Together, for all $y \geq N$, we obtain that

$$\mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y \geq y) < \frac{1}{3} + \frac{1}{3}\left(c^* - \frac{1}{3}\right). \tag{10}$$

Second, we want to estimate $\mathbb{P}(Y < N)$. Clearly, whenever $Y < N$, then we have that $Y_I < N$ or $Y_J < N$. Hence, by a union bound,

$$\begin{aligned}
\mathbb{P}(Y < N) &\leq \mathbb{P}(Y_I < N \text{ or } Y_J < N) \qquad (11)\\
&\leq \mathbb{P}(Y_I < N) + \mathbb{P}(Y_J < N) = 2\mathbb{P}(Y_I < N) \qquad (12)
\end{aligned}$$

We now want to bound $\mathbb{P}(Y_I < N)$. Note that b arrives with equal probability in every fixed position among the agents in $I \cup \{b\}$. Hence, $Y_I < N$ to happen is equal to b arriving in a position in $\{1, \ldots, N\}$ among $I \cup \{b\}$. We conclude that $\mathbb{P}(Y_I < N) = \frac{N}{\frac{n}{2}} = \frac{2N}{n}$.

Note that this tends to 0 for n tending to infinity. Therefore, there exists $N' \geq N$ such that $\mathbb{P}(Y_I < N) \leq \frac{1}{6}\left(c^* - \frac{1}{3}\right)$ for all $n \geq N'$. Combining this with Equation (12), for all $n \geq N'$, we obtain

$$\mathbb{P}(Y < N) \leq \frac{1}{3}\left(c^* - \frac{1}{3}\right). \tag{13}$$

For $n \geq N'$, we conclude that

$$\begin{aligned}
&\mathbb{P}(\{a,b\} \in ALG^*(B))\\
&= \mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y < N)\mathbb{P}(Y < N)\\
&+ \mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y \geq N)\mathbb{P}(Y \geq N)\\
&\leq \mathbb{P}(Y < N) + \mathbb{P}(\{a,b\} \in ALG^*(B) \mid Y \geq N)\\
&\overset{\text{Eqs. (10,13)}}{\leq} \frac{1}{3}\left(c^* - \frac{1}{3}\right) + \left(\frac{1}{3} + \frac{1}{3}\left(c^* - \frac{1}{3}\right)\right)\\
&\leq \frac{1}{3} + \frac{2}{3}\left(c^* - \frac{1}{3}\right) = \frac{2}{3}c^* + \frac{1}{9} < c^*.
\end{aligned}$$

This contradicts our assumption that ALG^* was c^*-competitive. □

B.5 Lemmas in the Proof of Theorem 5.6

In this section, we prove auxiliary lemmas in the proof of Theorem 5.6, restated as follows.

Theorem 5.6. *No randomized online matching algorithm has a competitive ratio under random arrival of more than* $\frac{1}{3}$ *on the tree domain.*

We start with the proof of Lemma B.3. Its proof relies on two auxiliary statements concerning stars and bi-stars.

We first consider stars and want to estimate $\inf_{S \in \mathcal{S}} \mathbb{P}(\{a, d_{t_S}\} \in ALG(S))$, i.e., the infimum of the probability with which the maximum weight edge is matched in stars, where ALG is a $\hat{c}$-competitive online matching algorithm.

Lemma B.8. *Let* ALG *be a* $\hat{c}$*-competitive online matching algorithm. Then it holds that*

$$\inf_{S \in \mathcal{S}} \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) \geq \hat{c} - \epsilon.$$

Proof. Let ALG be a $\hat{c}$-competitive online matching algorithm. Consider some star instance $S \in \mathcal{S}$. By definition of the competitive ratio, it holds that

$$\frac{\mathbb{E}[\mathcal{SW}(ALG(S))]}{\mathcal{SW}(\pi^*(S))} = \frac{\mathbb{E}[\mathcal{SW}(ALG(S))]}{\left(\frac{1}{\epsilon}\right)^{t_S}} \geq \hat{c},$$

where $\pi^*(S)$ denotes the maximum weight matching. We compute

$$\begin{aligned}
\hat{c} \cdot \left(\frac{1}{\epsilon}\right)^{t_S} &\leq \mathbb{E}[\mathcal{SW}(ALG(S))] = \sum_{x,y \in N} \mathbb{P}(\{x,y\} \in ALG(S))v(x,y)\\
&\leq \sum_{i \in I} \mathbb{P}(\{a, d_i\} \in ALG(S))v(a, d_i)\\
&= \sum_{i \in I \setminus \{t_S\}} \mathbb{P}(\{a, d_i\} \in ALG(S)) \left(\frac{1}{\epsilon}\right)^i\\
&+ \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) \left(\frac{1}{\epsilon}\right)^{t_S}
\end{aligned}$$

In the second line, we express the expectation over matchings in terms of single edges. The third line follows from the fact that only the valuations between a and the agents associated with I are positive. Dividing both sides by $\left(\frac{1}{\epsilon}\right)^{t_S} > 0$, we get

$$\begin{aligned}
\hat{c} &\leq \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) + \sum_{i \in I \setminus \{t_S\}} \mathbb{P}(\{a, d_i\} \in ALG(S)) \frac{\left(\frac{1}{\epsilon}\right)^i}{\left(\frac{1}{\epsilon}\right)^{t_S}}\\
&\leq \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) + \sum_{i \in I \setminus \{t_S\}} \mathbb{P}(\{a, d_i\} \in ALG(S))\epsilon\\
&\leq \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) + \epsilon.
\end{aligned}$$

The last inequality follows since $\mathbb{P}(\{a, x\} \in ALG(S))$ for $x \in N$ forms a probability distribution since a cannot be matched with probability more than 1. Since $S \in \mathcal{S}$ was chosen arbitrarily, we obtain $\inf_{S \in \mathcal{S}} \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) \geq \hat{c} - \epsilon$. □

Next, we show that $\hat{c} - 2\epsilon$ is a lower bound on the probability with which a $\hat{c}$-competitive online matching algorithm matches the two centers in bi-star instances. The proof is similar to that of Lemma B.8.

Lemma B.9. *Let ALG be a $\hat{c}$-competitive online matching algorithm. Then it holds that*

$$\inf_{B \in \mathcal{B}} \mathbb{P}(\{a, b\} \in ALG(B)) \geq \hat{c} - 2\epsilon.$$

Proof. Consider a bi-star instance $B \in \mathcal{B}$. Then, by definition of the competitive ratio, it holds that

$$\frac{\mathbb{E}[\mathcal{SW}(ALG(B))]}{\mathcal{SW}(\pi^*(B))} = \frac{\mathbb{E}[\mathcal{SW}(ALG(B))]}{\left(\frac{1}{\epsilon}\right)^{t_B}} \geq \hat{c},$$

where $\pi^*(B)$ denotes the maximum weight matching. We compute

$$\begin{aligned}
\hat{c} \cdot \left(\frac{1}{\epsilon}\right)^{t_B+1} &\leq \mathbb{E}[\mathcal{SW}(ALG(B))] = \sum_{x,y \in N} \mathbb{P}(\{x, y\} \in ALG(B)) v(x, y) \\
&\leq \sum_{i \in I} \mathbb{P}(\{a, d_i\} \in ALG(B)) v(a, d_i) \\
&+ \sum_{j \in J} \mathbb{P}(\{b, d_j\} \in ALG(B)) v(b, d_j) \\
&+ \mathbb{P}(\{a, b\} \in ALG(B)) v(a, b) \\
&= \sum_{i \in I} \mathbb{P}(\{a, d_i\} \in ALG(B) \left(\frac{1}{\epsilon}\right)^{i} \\
&+ \sum_{j \in J} \mathbb{P}(\{b, d_j\} \in ALG(B)) \left(\frac{1}{\epsilon}\right)^{j} \\
&+ \mathbb{P}(\{a, b\} \in ALG(B)) \left(\frac{1}{\epsilon}\right)^{t_B+1}
\end{aligned}$$

In the second line, we express the expectation over matchings in terms of single edges. In the subsequent step, we omit edges with large negative weight. Dividing both sides by $\left(\frac{1}{\epsilon}\right)^{t_B+1} > 0$, we get

$$\begin{aligned}
\hat{c} &\leq \mathbb{P}(\{a, b\} \in ALG(B)) + \sum_{i \in I} \mathbb{P}(\{a, d_i\} \in ALG(B)) \frac{\left(\frac{1}{\epsilon}\right)^{i}}{\left(\frac{1}{\epsilon}\right)^{t_B+1}} \\
&+ \sum_{j \in J} \mathbb{P}(\{b, d_j\} \in ALG(B)) \frac{\left(\frac{1}{\epsilon}\right)^{j}}{\left(\frac{1}{\epsilon}\right)^{t_B+1}} \\
&\leq \mathbb{P}(\{a, b\} \in ALG(B)) + \sum_{i \in I} \mathbb{P}(\{a, d_i\} \in ALG(B)) \epsilon \\
&+ \sum_{j \in J} \mathbb{P}(\{b, d_j\} \in ALG(B)) \epsilon \\
&\leq \mathbb{P}(\{a, b\} \in ALG(B)) + 2\epsilon.
\end{aligned}$$

The third inequality follows since $\mathbb{P}(\{a, x\} \in ALG(B))$ and $\mathbb{P}(\{b, x\} \in ALG(B))$ for $x \in N$ form probability distributions since a and b cannot be matched with probability more than one. Since $B \in \mathcal{B}$ was chosen arbitrarily, we obtain $\inf_{B \in \mathcal{B}} \mathbb{P}(\{a, b\} \in ALG(B)) \geq \hat{c} - 2\epsilon$. ◻

We can combine Lemmas B.8 and B.9 to transition to the goal of proving that there is no algorithm matching the maximum weight edge that is better than $\frac{1}{3}$-competitive.

Lemmas B.8 and B.9, we can transition to the goal to prove that there is no algorithm matching the maximum weight edge that is better than $\frac{1}{3}$-competitive.

Lemma B.3. *If there exists a $\hat{c}$-competitive online matching algorithm on the tree domain, then there exists an algorithm for matching the maximum weight edge with a competitive ratio of more than $\frac{1}{3}$.*

Proof. Assume that there is a $\hat{c}$-competitive online matching algorithm on the tree domain. Consider $c' = \hat{c} - 2\epsilon$. Since $\epsilon = \frac{1}{3}(\hat{c} - \frac{1}{3})$, it holds that $c' = \frac{1}{3}\hat{c} - \frac{2}{9} > \frac{1}{3}$. By Lemmas B.8 and B.9, *ALG* is c'-competitive for matching the maximum weight edge. ◻

Next, we prove our lemma concerning history independence.

Lemma B.5. *For every star instance, we may assume without loss of generality that our algorithm's decisions only depend on*

- *which agents have arrived,*
- *whether a has arrived and is matched, and*
- *whether the last arrived agent is part of the current maximum weight edge.*

Proof. Consider an algorithm *ALG* restricted as per Lemma B.4. We transform this algorithm as follows: Consider the arrival of an agent and assume that the algorithm wants to match with positive probability. This means that the currently arrived agent is a or the agent of the maximum weight edge. Assume that, so far, agents in the set A have arrived. Let $H(A)$ be the history of the algorithm so far, which captures the arrival order of agents in A as well as all previous algorithmic decisions. Let $\mathcal{H}(A)$ be the set of all histories where the agents in A arrive such that the last arrived agent is part of the current maximum weight edge, and a is unmatched at the arrival of the last agent.

We obtain a new algorithm *ALG′* as follows. Upon the arrival of an agent that leads to a matching decision in *ALG* involving agents A, the algorithm *ALG′* ignores the history $H(A)$. Instead, it samples a history $H'(A) \sim \mathcal{H}(A)$ according to the probabilities of this history occurring in *ALG*.[14] Note that this is well-defined as we are operating on a finite game, for which there is only a finite set of histories, and the probabilities of each of the histories occurring only depends on algorithmic (randomized) decisions on all possible histories. Then, it matches the current maximum weight edge if and only if *ALG* would do so given the history $H'(A)$.

By design, we have that *ALG′* performs for $H(A)$ like *ALG* performs for $H'(A)$. Moreover, the distribution of the sampled histories is identical to the distribution of the real histories. Hence, the performance of *ALG′* in terms of matching the maximum weight edge is identical to the performance of *ALG*. However, the decisions of *ALG′* only depend on the set of agents that has arrived, whether a has arrived and is matched, and whether the last agent is part of the current maximum weight edge. ◻

Now, we prove that decisions can be assumed to be independent of b and J.

Lemma B.6. *For every star instance, we may assume without loss of generality that our algorithms decisions are independent of agents b and agents associated with J.*

---

[14] Note that we are not concerned about the computational complexity for designing this algorithm. Instead, we simply define an algorithm based on the potential randomizations of *ALG*. Note that this technique even applies when *ALG* is an "inefficient" algorithm, i.e., performs computations of any length.

Proof. Consider an algorithm *ALG* restricted as per Lemma B.4. Then, *ALG* never matches a negative weight edge. Hence, the first matching decision can happen when a arrives, and subsequently, *ALG* can only match the current maximum weight edge. Moreover, once a has arrived, it is revealed which present agents belong to I. We transform *ALG* so that every matching decision if it is still possible to match, is made as if b and agents associated with J have not yet arrived. In other words, *ALG* behaves on a star instance with respect to parameters I, J, x, and ϵ, as if J was the empty set. Note that the case of the same instance where J really is the empty set is another star instance, and it achieves the same performance as *ALG* achieved on this instance. Hence, its competitive ratio can only improve as it now only depends on a smaller set of star instances. □

Finally, we provide the bound on $h(S)$.

Lemma B.7. *Let $S \in \mathcal{S}$ with $|I| = k-1$. Then it holds that $h(S) > \frac{2}{3} - \frac{2}{3k}$ for all $S \in \mathcal{S}$.*

Proof. Given a star $S \in \mathcal{S}$, we additionally define

$$r(S) := \mathbb{P}(\{a, d_{t_S}\} \in ALG^*(S))$$

for all $S \in \mathcal{S}$, i.e., the probability to match a with d_{t_S}.

We now show recursive formulas for $h(S)$ and $r(S)$ assuming that we are given a star instance $S \in \mathcal{S}$ with $|I| = k-1$. To this end, we partition all arrival orders in $\Sigma(\{a\} \cup \{d_i \colon i \in I\})$, i.e., of the agents relevant to matching, into three sets based on the last arriving agent. The first two are the arrival orders σ in which a or d_{t_S} arrive last, i.e., $\sigma(k) = a$ or $\sigma(k) = d_{t_S}$, respectively. They each make up a $\frac{1}{k}$ fraction of all arrival orders, i.e., $\mathbb{P}(\sigma(k) = a) = \frac{1}{k}$ and $\mathbb{P}(\sigma(k) = d_{t_S}) = \frac{1}{k}$. In the remaining orders, one of the other alternatives arrives last. We have $\mathbb{P}(\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S}) = \frac{k-2}{k}$. Note that for $i \neq t_S$, if d_i arrives last, then the algorithm cannot match, so it is matched only if it has matched already. Furthermore, if d_{t_S} arrives last, then we need to consider two cases. Either a could already be matched or if it is unmatched then we match with probability $f(I, x)$. Finally, if a arrives last, then we match with probability $f(I, x)$.

$$\begin{aligned}
h(S) &= \mathbb{P}(\{a, d_i\} \in ALG^*(S) \text{ for some } i \in I)\\
&= \mathbb{P}(\{a, d_i\} \in ALG^*(S) \text{ for some } i \in I | \sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S})\\
&\quad \cdot \mathbb{P}(\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S})\\
&\quad + \mathbb{P}(\{a, d_i\} \in ALG^*(S) \text{ for some } i \in I | \sigma(k) = d_{t_S}) \mathbb{P}(\sigma(k) = d_{t_S})\\
&\quad + \mathbb{P}(\{a, d_i\} \in ALG^*(S) \text{ for some } i \in I | \sigma(k) = a) \mathbb{P}(\sigma(k) = a)\\
&= \frac{1}{k} \sum_{i \in I \setminus \{d_{t_S}\}} h(S[N \setminus \{d_i\}]) + \frac{1}{k}\big[h(S[N \setminus \{d_{t_S}\}])\\
&\quad + (1 - h(S[N \setminus \{d_{t_S}\}])) f(I, x)\big] + \frac{1}{k} f(I, x)\\
&= \frac{1}{k} \sum_{i \in I} h(S[N \setminus \{d_i\}]) - \frac{f(I, x)}{k} h(S[N \setminus \{d_{t_S}\}]) + \frac{2 f(I, x)}{k}
\end{aligned}$$

Furthermore, we have $h(S^x_{\{d_i\},J}) = f(\{a, d_i\}, x)$ for all $i \in I$ and the star where the only leaf from a is towards d_i and J is arbitrary.

We continue by calculating our second term. We have

$$\begin{aligned}
r(S) &= \mathbb{P}(\{a, d_{t_S}\} \in ALG^*(S))\\
&= \mathbb{P}(\{a, d_{t_S}\} \in ALG^*(S) | \sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S})\\
&\quad \cdot \mathbb{P}(\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S})\\
&\quad + \mathbb{P}(\{a, d_{t_S}\} \in ALG^*(S) | \sigma(k) = d_{t_S}) \mathbb{P}(\sigma(k) = d_{t_S})\\
&\quad + \mathbb{P}(\{a, d_{t_S}\} \in ALG^*(S) | \sigma(k) = a) \mathbb{P}(\sigma(k) = a)\\
&= \frac{1}{k} \sum_{i \in I \setminus \{d_{t_S}\}} r(S[N \setminus \{d_i\}])\\
&\quad + \frac{1}{k} f(I, x)(1 - h(S[N \setminus \{d_{t_S}\}])) + \frac{1}{k} f(I, x)\\
&= \frac{1}{k} \sum_{i \in I \setminus \{d_{t_S}\}} r(S[N \setminus \{d_i\}]) - \frac{f(I, x)}{k} h(S[N \setminus \{d_{t_S}\}]) + \frac{2 f(I, x)}{k}
\end{aligned}$$

In addition, it holds that $r(S^x_{\{d_i\},J}) = f(\{a, d_i\}, x)$ since if ALG^* matches in this case, then it matches the optimal edge.

Next, we compute $h(S) - r(S)$, i.e., the probability of matching a suboptimal valuation in a star. Some terms will cancel out because the probabilities of matching optimally ($r(S)$) and matching at all ($h(S)$) only differ if the last agent to arrive is not a or d_{t_S}.

$$\begin{aligned}
&h(S) - r(S)\\
&= \frac{1}{k} \sum_{i \in I} h(S[N \setminus \{d_i\}]) - \frac{f(I, x)}{k} h(S[N \setminus \{d_{t_S}\}]) + \frac{2 f(I, x)}{k}\\
&\quad - \frac{1}{k} \sum_{i \in I \setminus \{d_{t_S}\}} r(S[N \setminus \{d_i\}]) + \frac{f(I, x)}{k} h(S[N \setminus \{d_{t_S}\}]) - \frac{2 f(I, x)}{k}\\
&= \frac{1}{k} \sum_{i \in I} h(S[N \setminus \{d_i\}]) - \frac{1}{k} \sum_{i \in I \setminus \{d_{t_S}\}} r(S[N \setminus \{d_i\}])\\
&= \frac{1}{k} h(S[N \setminus \{d_{t_S}\}]) + \frac{1}{k} \sum_{i \in I \setminus \{d_{t_S}\}} h(S[N \setminus \{d_i\}]) - r(S[N \setminus \{d_i\}])
\end{aligned}$$

We can repeatedly apply the recursive equation that we just derived. On the right side, this amounts to summing

$$\frac{1}{k} \frac{(k-1-|\tilde{I}|)!|\tilde{I}|!}{(k-1)!} h(S[N \setminus (\{d_{t_S} \cup \tilde{I})\}])$$

for all $\tilde{I} \subsetneq I \setminus \{d_{t_S}\}$. The factor $\frac{(k-1-|\tilde{I}|)!}{(k-1)!}$ collects the accumulated prefactors of all steps, and the factor $|\tilde{I}|!$ accounts for the fact that we can arrive at the same term by removing the alternatives in $\tilde{I}$ in any order. Finally, the remaining difference after removing all elements in $I \setminus \{d_{t_S}\}$ is $h(S^x_{\{d_{t_S}\},\tilde{I}}) - r(S^x_{\{d_{t_S}\},\tilde{I}}) = 0$, which cancels out. We can rewrite $\frac{(k-1-|\tilde{I}|)!|\tilde{I}|!}{(k-1)!} = \frac{1}{\binom{k-1}{|\tilde{I}|}}$. This yields:

$$h(S) - r(S) = \frac{1}{k} \sum_{\tilde{I} \subsetneq I \setminus \{d_{t_S}\}} \frac{1}{\binom{k-1}{|\tilde{I}|}} h(S[N \setminus (\{d_{t_S} \cup \tilde{I})\}]) \qquad (14)$$

Define $||S|| := |I| + 1$ if S is a star defined by I, i.e., $||S|| = |I \cup \{a\}|$. Hence, we have that $||S|| = k$.

We now show the lemma by strong induction over $||S||$. Note that $I \neq \emptyset$ and, therefore, $||S|| \geq 2$ in all star instances. If $||S|| = 2$, then $h(S) = r(S)$. Thus,

$$h(S) = r(S) \geq c^* > \frac{1}{3} = \frac{2}{3} - \frac{2}{3 \cdot 2}.$$

Now assume for all stars S with $||S|| \leq k-1$, it holds that $h(S) > \frac{2}{3} - \frac{2}{3||S||}$. In the following, we use the binomial identity

$$\binom{n-1}{k} = \frac{n-k}{n}\binom{n}{k}. \tag{15}$$

Recall that $|I| = k-1$ and, therefore, $|I \setminus \{d_{t_S}\}| = k-2$. We compute

$$\begin{aligned}
h(S) &\overset{Eq.~(14)}{=} r(S) + \frac{1}{k} \sum_{\tilde{I} \subsetneq I \setminus \{d_{t_S}\}} \frac{1}{\binom{k-1}{|\tilde{I}|}} h(S[N \setminus (\{d_{t_S} \cup \tilde{I})\}]) \\
&> \frac{1}{3} + \frac{1}{k} \sum_{\tilde{I} \subsetneq I \setminus \{d_{t_S}\}} \frac{1}{\binom{k-1}{|\tilde{I}|}} \left(\frac{2}{3} - \frac{2}{3(k-1-|\tilde{I}|)}\right) \\
&= \frac{1}{3} + \frac{1}{k} \sum_{i=0}^{k-3} \frac{\binom{k-2}{i}}{\binom{k-1}{i}} \left(\frac{2}{3} - \frac{2}{3(k-1-i)}\right) \\
&\overset{Eq.~(15)}{=} \frac{1}{3} + \frac{1}{k} \sum_{i=0}^{k-3} \frac{\binom{k-1}{i}}{\binom{k-1}{i}} \frac{k-1-i}{k-1} \left(\frac{2}{3} - \frac{2}{3(k-1-i)}\right) \\
&= \frac{1}{3} + \frac{1}{k} \sum_{i=0}^{k-3} \left(\frac{k-1}{k-1} - \frac{i}{k-1}\right) \left(\frac{2}{3} - \frac{2}{3(k-1-i)}\right) \\
&= \frac{1}{3} + \frac{1}{k} \sum_{i=0}^{k-3} \left(1 - \frac{i}{k-1}\right) \left(\frac{2}{3} - \frac{2}{3(k-1-i)}\right) \\
&= \frac{1}{3} + \frac{1}{k} \sum_{i=0}^{k-2} \left(1 - \frac{i}{k-1}\right) \left(\frac{2}{3} - \frac{2}{3(k-1-i)}\right).
\end{aligned}$$

In the last step, we inserted the term for $i = k-2$, which evaluates to 0 as $\frac{2}{3(k-1-(k-2))} = \frac{2}{3}$.

We finally simplify the two parts of the equation individually. For the first term, we obtain

$$\begin{aligned}
\frac{1}{k} \sum_{i=0}^{k-2} \left(1 - \frac{i}{k-1}\right) \frac{2}{3} &= \frac{2}{3k} \left(\sum_{i=0}^{k-2} 1 - \sum_{i=0}^{k-2} \frac{i}{k-1}\right) \\
&= \frac{2}{3k} \left(k-1 - \frac{1}{k-1} \frac{(k-1)(k-2)}{2}\right) \\
&= \frac{2}{3k} \frac{2k-2-k+2}{2} = \frac{2}{3k} \frac{k}{2} = \frac{1}{3}
\end{aligned}$$

For the second term, we obtain

$$\begin{aligned}
\frac{1}{k} \sum_{i=0}^{k-2} \left(1 - \frac{i}{k-1}\right) \frac{2}{3(k-1-i)} &= \frac{2}{3k} \sum_{i=0}^{k-2} \frac{1 - \frac{i}{k-1}}{k-1-i} = \frac{2}{3k} \sum_{i=0}^{k-2} \frac{\frac{k-1-i}{k-1}}{k-1-i} \\
&= \frac{2}{3k} \sum_{i=0}^{k-2} \frac{1}{k-1} = \frac{2}{3k}
\end{aligned}$$

Inserting back into our equation we get

$$h(S) > \frac{1}{3} + \frac{1}{k} \sum_{i=0}^{k-2} \left(1 - \frac{i}{k-1}\right) \left(\frac{2}{3} - \frac{2}{3(k-1-i)}\right) = \frac{1}{3} + \frac{1}{3} - \frac{2}{3k} = \frac{2}{3} - \frac{2}{3k}.$$

This completes the proof. □

C FULL PROOF OF THEOREM 6.2

Our proof of Theorem 6.2 relies on a similar idea as the proof by Badanidiyuru Varadaraja [4], showing that there does not exist an online matching algorithm (in an edge arrival setting) operating under free dissolution for which the competitive ratio is better than $\frac{1}{3+2\sqrt{2}}$. His proof relies on two steps. First, he shows that a particular sequence of real numbers cannot exist based on a recursive set of inequalities. Second, he shows that the existence of an algorithm with a competitive ratio of better than $\frac{1}{3+2\sqrt{2}}$ implies the existence of just such a sequence. We will use his first step as a black box and then use an adversarial instance of online FHGs to construct the sequence utilizing an online coalition formation algorithm that achieves a competitive ratio of better than $\frac{1}{6+4\sqrt{2}}$. The construction of our adversarial instance is similar to the one by Badanidiyuru Varadaraja [4]. Still, while his optimal partition is a matching consisting of coalitions of size 2, we construct the instance in a way such that the optimal instance consists of coalitions that form stars (i.e., we have symmetric valuations that are equal to some constant if they involve a special center agent and are 0, otherwise). This accounts for the improvement of about a factor of 2 in the welfare of the optimal partition.

We start by stating the lemma that captures the nonexistence of the sequence.

Lemma C.1 (Badanidiyuru Varadaraja [4]). *Let $\beta > \frac{1}{3+2\sqrt{2}}$. Then there exists no sequence $(x_i)_{i \in \mathbb{N}}$ with $x_1 = 1$ and $x_i \geq 0$ for $i \geq 2$ such that for all $i \in \mathbb{N}$, it holds that*

$$x_i \geq \beta \left(x_{i+1} + \sum_{j=1}^{i+1} x_j\right). \tag{16}$$

Next, we evaluate the social welfare of a "star" coalition.

Lemma C.2. *Let $x \in \mathbb{R}$. Consider a set of agents C such that there exists $a \in C$ with symmetric valuations $v(a,b) = x$ for all $b \in C \setminus \{a\}$ and $v(b,b') = 0$ for all $b, b' \in C \setminus \{a\}$ with $b \neq b'$. Then it holds that $\mathcal{SW}(C) = 2\frac{|C|-1}{|C|}x$.*

Proof. Assume that we are in the lemma's situation. Then, $u_a(C) = \frac{|C|-1}{|C|}x$, and for all $b \in C \setminus \{a\}$, it holds that $u_b(C) = \frac{1}{|C|}x$. The assertion follows by summing up utilities. □

We are ready to prove our theorem.

Theorem 6.2. *No deterministic online coalition formation algorithm operating under free dissolution has a competitive ratio of more than $\frac{1}{6+4\sqrt{2}}$ for symmetric FHGs.*

Proof. Let $c := \frac{1}{6+4\sqrt{2}}$. Assume for contradiction that *ALG* is an online coalition formation algorithm operating under free dissolution that achieves a competitive ratio of $\gamma > c$ for symmetric FHGs. Without loss of generality, we may assume that $\frac{c}{\gamma}$ is rational.[15] We want this property to assure that all instances we construct exclusively use rational numbers.

Let

$$\beta := 2\left(c + \frac{1}{2}(\gamma - c)\right) = \gamma + c, \tag{17}$$

---

[15] Indeed, otherwise, we can just perform the proof for a γ' in the open interval (c, γ) with this property. Such a γ' exists as the function $f : [c, \gamma] \to \mathbb{R}$, $f(x) = \frac{c}{x}$ is continuous and hence, by the density of the rational numbers in the real numbers, attains rational numbers in the open interval (c, γ).

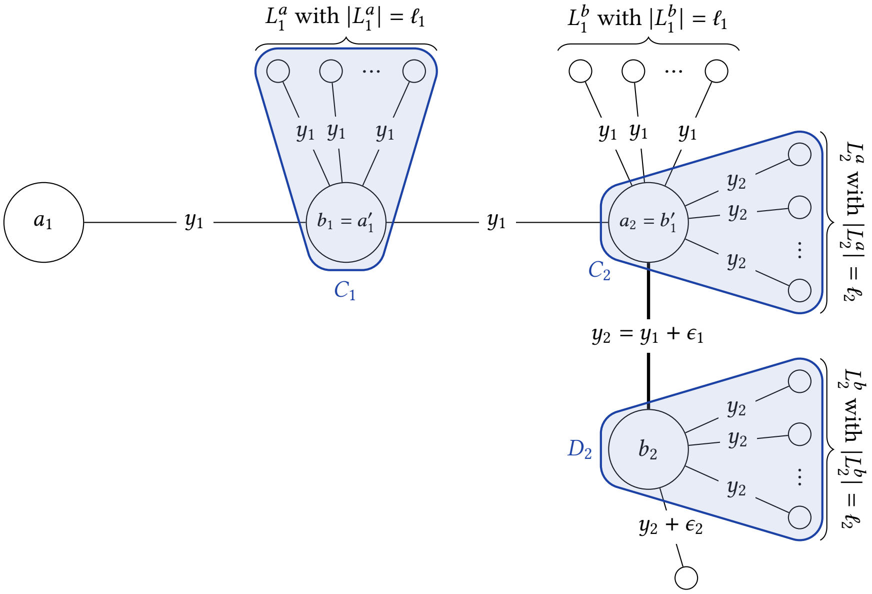

Figure 2: Illustration of the construction in the proof of Theorem 6.2 for an exemplary algorithm *ALG*. We display all positive valuations. The remaining valuations within the leaf sets L_1^a, L_1^b, L_2^a, and L_2^b are zero, and all other valuations are large negative numbers. We start with two agents, a_1 and b_1. We first attempt to dispatch a set L_1^b of leaves towards b_1. However, our algorithm might immediately decide to dissolve $\{a_1, b_1\}$ and create a new coalition $\{a_1', b_1'\}$. We then might be able to have all the leaf agents in L_1^a and L_1^b arrive. This completes the first part of Phase 1. Now, we start the second part, in which we subsequently increment the valuations. *ALG* might decide to immediately dissolve $\{a_1', b_1'\}$ when the next agent arrives. This defines agents a_2, b_2, and coalition C_1. We start with Phase 2. In the first part, the leaf agents L_2^a and L_2^b might arrive without further interruption. Now assume that *ALG* would dissolve $\{a_2, b_2\}$ when the next agent arrives (their edge is indicated in bold). This would give rise to the definition of C_2 and D_2, and we would obtain an inequality for y_2 by comparing with the guarantee for the coalition structure containing the nonempty coalitions C_1, C_2, and D_2.

i.e., it holds that $\beta > 2c = \frac{1}{3+2\sqrt{2}}$. We will eventually derive a contraction to Lemma C.1 by constructing a sequence for this β.

We construct an adversarial instance for this algorithm by constructing a symmetric graph $G = (N, v)$, i.e., we specify the symmetric weights underlying the valuations of an FHG.

The construction maintains the property that the algorithm's current partition can only contain a single coalition with positive welfare and that coalition contains exactly two agents. The adversarial instance is constructed in a sequence of phases, where in every phase, we grow star-like structures around each of the endpoints of the currently maintained nonsingleton coalition. In the first part of Phase i, we achieve a star with ℓ_i leaves, while the algorithm does not change the matched edges. In the second part of Phase i, we iteratively increase the weight on the edges of the stars by ϵ_i until the algorithm changes the matched edge. This has to happen eventually because the algorithm achieves a bounded competitive ratio.

We now specify the two parameters of the construction. For $i \in \mathbb{N}$, define

$$\epsilon_i := \frac{\gamma - c}{2\gamma} 2^{-i} \quad \text{and} \quad \ell_i := \left\lceil \frac{1 - \epsilon_i}{\epsilon_i} \right\rceil. \tag{18}$$

Note that, since $\frac{c}{\gamma}$ is rational, $\epsilon_i = \left(\frac{1}{2} - \frac{c}{\gamma}\right) 2^{-i}$ is also rational. Moreover, the definition of ℓ_i immediately implies that

$$\frac{\ell_i}{\ell_i + 1} \geq 1 - \epsilon_i. \tag{19}$$

We now specify the instance. Our whole construction is illustrated in Figure 2.

The first two agents that arrive are a_1 and b_1 such that $v(a_1, b_1) = 1$. Clearly, *ALG* has to form the coalition $\{a_1, b_1\}$ as otherwise, its competitive ratio would be unbounded. For $i \geq 1$, at the beginning of Phase i, there is a single coalition with nonzero welfare containing precisely agents a_i and b_i.

Moreover, throughout the execution of the instance, all arriving agents will have a positive (mutual) valuation for precisely one agent—one of the agents that presently is in a coalition of positive welfare—, a zero valuation for some agents, and a large negative valuation for all other agents. In particular, the second agent in the coalition of positive welfare yields a large negative valuation, and thus, joining this coalition leads to an overall negative welfare, which cannot be performed by any algorithm with a positive competitive ratio. Hence, the new agent only forms a coalition of positive welfare if the previously existing coalition with positive welfare is dissolved.

Now let $i \geq 1$ and assume that we are at the beginning of Phase i, i.e., so far *ALG* has constructed a partition containing a single coalition with positive welfare containing a_i and b_i. We set

$$y_i := v(a_i, b_i). \tag{20}$$

In the first part of Phase 1, we want to guarantee that at the end of this part, there is a single coalition of positive welfare $C = \{a_i', b_i'\}$ such that for each of a_i' and b_i', ℓ_i agents have arrived such that there are 0-valuations among these agents and a valuation of y_i towards a_i' or b_i'. In other words, the instance contains a bi-star as a substructure where all edges weigh y_i.

We start by setting $a_i' := a_i$ and $b_i' := b_i$. Now, we let arrive a set L_i^b of up to ℓ_i agents that have a valuation of y_i for b_i, 0 for already arrived agents in L_i^b, and a sufficiently large negative valuation for all other agents, e.g., a negative value larger in absolute value than the sum of positive valuations of already existing agents. As we argued before, the only way that *ALG* puts an agent in L_i^b into a coalition of positive welfare is if the coalition of a_i' and b_i' is dissolved and the new agent forms a coalition with b_i'. In this case, we update agent labels: b_i' becomes the new a_i', and the newly arrived agent is the new b_i'.

We repeat this until ℓ_i agents have arrived. Note that this has to happen at some point as we would otherwise have a path of unbounded length with edge weights equal to y_i, which would give rise to a partition of social welfare more than $\frac{1}{\gamma}y_i$, a contradiction.

Now, we repeat the same procedure with a_i': we let arrive a set L_i^a of up to ℓ_i agents that have a valuation of y_i for a_i, 0 for already arrived agents in L_i^a, and a sufficiently large negative valuation for all other agents. If the algorithm decides to dissolve $\{a_i', b_i'\}$ to form a coalition of a_i' with a newly arrived agent, we update agent labels: a_i' stays the new a_i', and the newly arrived agent is the new b_i'. Note that this part must eventually end with all ℓ_i agents having arrived. Otherwise, we have an unbounded number of agents that at some point had the role of b_i', and each of them can form a coalition with an agent in their set L_i^b, which yields unbounded welfare.

We reach the end of the first part of Phase i and have established a pair of agents $\{a_i', b_i'\}$ together with their sets L_i^a and L_i^b. Note that the coalitions $\{a_i'\} \cup L_i^a$ and $\{b_i'\} \cup L_i^b$ are "star" coalitions as in the prerequisites of Lemma C.2.

We now start the second part of Phase i. In this part, further agents arrive that are new leaves to a_i' and b_i'. Compared to y_i, the weight on their connecting edges is increased by ϵ_i. We continue until for each of a_i' and b_i', further sets of ℓ_i agents have arrived. This means that we have now grown starts with a slightly larger value. We repeat the same with increasingly larger valuations in further increments of ϵ_i. The second part of Phase i ends once throughout this procedure, the edge $\{a_i', b_i'\}$ gets dissolved.

We now formalize this idea. Set $L_i^{a,0} := L_i^a$ and $L_i^{b,0} := L_i^b$. We proceed as follows until the algorithm dissolves a coalition and forms a new coalition of positive welfare. For each $j \geq 1$, once all agents in the sets $L_i^{a,j-1}$ and $L_i^{b,j-1}$ have arrived, we proceed as follows. We let a set $L_i^{a,j}$ with ℓ_i agents arrive that have a valuation of $y_i + j\epsilon_i$ for a_i, 0 for already arrived agents in $L_i^{a,j}$, and a sufficiently large negative valuation for all other agents. These agents arrive one by one, so the phase can end before all agents in $L_i^{a,j}$ have arrived. Then we let a set $L_i^{b,j}$ with ℓ_i agents arrive that have a valuation of $y_i + j\epsilon_i$ for b_i, 0 for already arrived agents in $L_i^{b,j}$, and a sufficiently large negative valuation for all other agents.

Note that this part also has to terminate at some point as otherwise agents with an unbounded valuation arrive, leading to a partition of welfare higher than $\frac{1}{\gamma}y_i$.

Once the algorithm forms a new coalition—say this happens when the j^*th sets of agents arrive—we distinguish two cases: If a_i' remains in a nonsingleton coalition with the new agent z, we define $C_i := \{b_i'\} \cup L_i^{b,j^*-1}$ and $D_i := \{a_i'\} \cup L_i^{a,j^*-1}$ and set $a_{i+1} = a_i'$ and $b_{i+1} = z$. Otherwise, if b_i' remains in a nonsingleton coalition with the new agent z, we define $C_i := \{a_i'\} \cup L_i^{a,j^*-1}$ and $D_i := \{b_i'\} \cup L_i^{b,j^*-1}$ and set $a_{i+1} = b_i'$ and $b_{i+1} = z$.

Then, the new agents a_{i+1} and b_{i+1} are the only agents in a coalition of positive welfare $y_{i+1} = v(a_{i+1}, b_{i+1})$. Moreover, C_i and D_i are "star" coalitions that are disjoint from all previous coalitions C_k for $k < i$ and where all nonzero valuations are $y_{i+1} - \epsilon_i$. By Lemma C.2, we obtain

$$\mathcal{SW}(C_i) = \mathcal{SW}(D_i) = 2\frac{\ell_i}{\ell_i + 1}(y_{i+1} - \epsilon_i). \tag{21}$$

Consider the partition π_i containing the coalitions D_i, C_j for $1 \leq j \leq i$, and singleton coalitions for all agents not contained in these. This coalition already exists right before the arrival of the agent such that the coalition $\{a_i', b_i'\}$ is dissolved. Note that at this point, the social welfare of the partition created by *ALG* is y_i, where we add $\frac{y_i}{2}$ for each of a_i' and b_i'. Since *ALG* is γ-competitive, we obtain

$$\begin{aligned}
y_i &\geq \gamma \cdot \mathcal{SW}(\pi_i) = \gamma \left(\mathcal{SW}(D_i) + \sum_{j=1}^{i} \mathcal{SW}(C_j) \right) \\
&\overset{(21)}{=} \gamma \left(2\frac{\ell_i}{\ell_i + 1}(y_{i+1} - \epsilon_i) + \sum_{j=1}^{i} 2\frac{\ell_j}{\ell_j + 1}(y_{j+1} - \epsilon_j) \right) \\
&\overset{(19)}{\geq} \gamma \left(2(1 - \epsilon_i)(y_{i+1} - \epsilon_i) + \sum_{j=1}^{i} 2(1 - \epsilon_j)(y_{j+1} - \epsilon_j) \right) \\
&\geq \gamma \left(2(y_{i+1} - 2y_{i+1}\epsilon_i) + \sum_{j=1}^{i} 2(y_{j+1} - 2y_{j+1}\epsilon_j) \right) \\
&\geq \gamma \left(2(y_{i+1} - 2y_{i+1}\epsilon_i) + \sum_{j=1}^{i} 2(y_{j+1} - 2y_{i+1}\epsilon_j) \right) \\
&= 2\gamma \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right) - 2\gamma y_{i+1} \left(\epsilon_i + \sum_{j=1}^{i} \epsilon_j \right) \\
&\overset{(17),\,(18)}{=} (\beta + (\gamma - c)) \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right) - 2\gamma y_{i+1} \left(\frac{\gamma - c}{2\gamma} 2^{-i} + \sum_{j=1}^{i} \frac{\gamma - c}{2\gamma} 2^{-j} \right) \\
&\geq \beta \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right) + (\gamma - c)y_{i+1} - (\gamma - c)y_{i+1} \left(2^{-i} + \sum_{j=1}^{i} 2^{-j} \right) \\
&= \beta \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right).
\end{aligned}$$

We obtain our desired sequence by scaling the y_i and starting with y_2. Formally, for $i \in \mathbb{N}$, we set $x_i := \frac{y_{i+1}}{y_2}$. Then, $x_1 = \frac{y_2}{y_2} = 1$ and for $i \geq 2$, it holds that $x_i \geq 0$. Moreover, for $i \geq 1$, our previous calculation implies that

$$x_i = \frac{y_{i+1}}{y_2} \geq \frac{1}{y_2}\beta\left(y_{i+2} + \sum_{j=2}^{i+2} y_j\right) = \beta\left(\frac{y_{i+2}}{y_2} + \sum_{j=2}^{i+2} \frac{y_j}{y_2}\right)$$

$$= \beta\left(x_{i+1} + \sum_{j=2}^{i+2} x_{j-1}\right) = \beta\left(x_{i+1} + \sum_{j=1}^{i+1} x_j\right).$$

Hence, we have constructed the desired sequence and obtained a contradiction by applying Lemma C.1. □