The Power of Matching for Online Fractional Hedonic Games

Martin Bullinger
University of Oxford, UK
martin.bullinger@cs.ox.ac.uk

René Romen and Alexander Schlenga
Technische Universität München, Germany
{rene.romen,alexander.schlenga}@tum.de

Abstract

We study coalition formation in the framework of fractional hedonic games (FHGs). The objective is to maximize social welfare in an online model where agents arrive one by one and must be assigned to coalitions immediately. For the basic setting, computing maximal matchings achieves the optimal competitive ratio, which is, however, unbounded for unbounded agent valuations.

We achieve a constant competitive ratio in two related settings while carving out further connections to matchings. If algorithms can dissolve coalitions, then the optimal competitive ratio of $\frac{1}{6+4\sqrt{2}}$ is achieved by a matching-based algorithm. Moreover, we perform a tight analysis for the online matching setting under random arrival with an unknown number of agents. This entails a randomized $\frac{1}{6}$-competitive algorithm for FHGs, while no algorithm can be better than $\frac{1}{3}$-competitive.

1 Introduction

The formation of coalitions is a widely studied problem at the intersection of artificial intelligence, game theory, and the social sciences (Ray, 2007; Aziz and Savani, 2016). The goal is to form groups from a set of agents, which could represent members of a society or, more broadly, firms or computer programs. We call the resulting coalition structure a *partition*, and agents have preferences concerning their assigned coalition. This setting has undergone in-depth scrutiny in game theory where a particularly appealing and well-studied class of coalition formation games are *hedonic games* (Drèze and Greenberg, 1980). Their central—hedonic—aspect is that the preferences of an agent only depend on the members of her coalition but not on the structure or members of other coalitions.

However, even under this natural restriction, stating preferences explicitly requires the consideration of an exponentially large set of potential coalitions.

Hence, for the sake of computational tractability, a significant amount of research has been undertaken concerning hedonic games with inherently concise preference representations. One way of achieving this is to derive an agent's preferences over coalitions from her preferences over single agents. For instance, agents might assign a subjective valuation to each other agent, which can then be aggregated to obtain utilities over coalitions. This approach gives rise to the classes of additively separable (ASHG) or fractional (FHG) hedonic games (Bogomolnaia and Jackson, 2002; Aziz et al., 2019). In this work, we focus on FHGs, in which the utility an agent assigns to a coalition is the average utility she assigns to the coalition members (assuming a utility of 0 for herself). This model has been argued to be suitable for the analysis of network clustering and can be used to represent basic economic scenarios such as the bakers-and-millers game (Aziz et al., 2019).

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 (Flammini et al., 2021b). In the basic model, agents arrive one by one and have to be assigned to existing coalitions 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 absolute value of nonzero utilities, the best possible competitive ratio is $\frac{V_{\min}}{4V_{\max}}$.

A crucial role in achieving welfare approximations has been to employ matchings, which can be interpreted as partitions with coalitions of size at most 2.[1] For instance, the aforementioned competitive ratio is attained by forming maximal matchings, which is even the best deterministic approach for unweighted games (Flammini et al., 2021b). 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 (Flammini et al., 2021a). 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 $\epsilon > 0$ (Flammini et al., 2022), even if weights are bounded globally (Bullinger et al., 2025). Our work extends this intuition by considering two more sophisticated models of online FHGs, where we show that online matching algorithms achieve a constant optimal or close to optimal performance.

In the first model, the algorithm gets the additional power to dissolve coalitions. We show that a 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 matching algorithm in the corresponding matching domain. In the second setting, the algorithm cannot revoke matching decisions, but agents arrive in a uniformly random order. This also avoids the worst-case example by Flammini et al. (2021b), which crucially relies on specifying valuations based on the previous decisions of algorithms. We achieve a $\frac{1}{6}$-competitive algorithm, while no algorithm can be better than

[1] A notable exception are online FHGs with nonnegative weights, for which the optimal algorithm forms coalitions of unbounded size (Flammini et al., 2021b).

$\frac{1}{3}$-competitive. The latter result relies on a tight analysis of matching algorithms with an unknown number of agents, for which a competitive ratio of $\frac{1}{3}$ is optimal. Since we prove this result on the tree domain, a specific domain of instances where positive valuations form trees, it directly transfers algorithmic limitations to the coalition formation setting. We thus once again observe the power of matching algorithms when analyzing an online coalition formation model.

2 Related Work

The hedonic formation of coalitions traces back to Drèze and Greenberg (1980), while hedonic games in the form studied today have been conceptualized by Bogomolnaia and Jackson (2002). The latter paper introduces the class of ASHGs, in which utilities for coalitions are obtained through a sum-based aggregation of individual valuations. Fractional hedonic games were introduced later by Aziz et al. (2019). An overview of hedonic games can be found in the book chapters by Aziz and Savani (2016) and Bullinger et al. (2024).

Several authors studied various notions of stability in FHGs (Brandl et al., 2015; Bilò et al., 2015, 2018; Kaklamanis et al., 2016; Aziz et al., 2019; Brandt and Bullinger, 2022), while Aziz et al. (2015) 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. Matching algorithms are shown to yield reasonable approximation ratios. In particular, Aziz et al. (2015) show that a maximum weight matching (MWM) is a $\frac{1}{4}$-approximation of social welfare in general, unconstrained FHGs. This analysis was later improved and made tight by Flammini et al. (2021a) who prove that MWMs yield precisely a $\frac{1}{2}$-approximation.

An online model for hedonic games was first studied by Flammini et al. (2021b), who consider FHGs and ASHGs.[2] They investigate the model where agents arrive in an adversarial order. They 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 (2023) 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 (2024) 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. (2024). Here, we only discuss the works that are closest to our setting. For unweighted graphs, Gamlath et al. (2019) give the online algorithm with the currently best known competitive ratio for maximum cardinality matchings with adversarial vertex arrival. Kesselheim et al. (2013)

[2]Flammini et al. (2021b) refer to their model as a "coalition structure generation problem" and, therefore, adopt a purely graph-theoretic instead of a game-theoretic perspective.

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. (2022) propose 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. Finally, Bullinger and Romen (2023) study online MWM under free dissolution.

3 Preliminaries and Model

We begin by introducing some notation. For $i \in \mathbb{N}$, we denote $[i] := \{1, \ldots, i\}$. For a set S and $i \in \mathbb{N}$, let $\binom{S}{i} := \{T \subseteq S \mid |T| = i\}$, i.e., $\binom{S}{i}$ denotes the set of all subsets of S of size 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 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. A *matching* is a partition in which all coalitions have size at most 2.[3] For agent $i \in N$ and partition π, let $\pi(i)$ denote the unique coalition in π that i belongs to.

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. (2019), 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 $u_i(C) = \sum_{j \in C \setminus \{i\}} \frac{v_i(j)}{|C|}$. 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

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

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.

The desirability of a partition is measured in terms of social welfare. Given an FHG $G = (N, v)$, we define the *social welfare* of a coalition $C \subseteq N$ as $\mathcal{SW}(C) := \sum_{i \in C} u_i(C)$ and of a partition π as $\mathcal{SW}(\pi) := \sum_{i \in N} u_i(\pi) = \sum_{C \in \pi} \mathcal{SW}(C)$. 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 (Bullinger, 2020). 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 game G if $\mathcal{SW}(\pi) \geq c \cdot \mathcal{SW}(\pi^*(G))$.

If π is a matching, then $\mathcal{SW}(\pi)$ also denotes the weight of the matching (since for each matched pair, both agents contribute $\frac{1}{2}$ of the edge weight). Hence, maximizing social welfare among matchings is precisely the *maximum weight matching* (MWM) problem.

3.2 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 in a predefined way. 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 $ALG(G, \sigma)_1, \ldots, ALG(G, \sigma)_{n(G)}$ of partitions such 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 $i, j \in [k]$.[4] 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. Furthermore, decisions must be identical if all valuations are identical up to this agent's arrival. The output of the algorithm is the partition produced when the final agent is added; we denote $ALG(G, \sigma) := ALG(G, \sigma)_{n(G)}$

Moreover, an algorithm's decisions are assumed to be irrevocable, i.e., agents can only be added to an existing or a completely new coalition. Formally, this means that for all instances (G, σ) and $2 \leq k \leq [n(G)]$, we require that

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

$ALG(G,\sigma)_k[\{\sigma(i) \mid 1 \leq i \leq k-1\}] = ALG(G,\sigma)_{k-1}$, 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 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 a good welfare approximation. We say that ALG is *c-competitive*[5] 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.

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

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

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 between Matchings and FHGs

The first significant connection between MWMs and welfare maximization in FHGs is that the former yields a $\frac{1}{2}$-approximation for the latter. In Section A, we show a very instructive alternative proof of this theorem originally shown by Flammini et al. (2021a). Our argument establishes the connection between MWM and FHGs via random matchings. More precisely, it is easy to see that the social welfare of the MWM is at least as much as the sum of the social welfare of random matchings on an arbitrary partition of the agents. Furthermore, we show that a random matching in a coalition is a $\frac{1}{2}$-approximation of the social welfare of the coalition. If we apply these arguments to the optimal partition, the theorem follows directly.

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

Theorem 1 (Flammini et al. (2021a)). *Every MWM is a $\frac{1}{2}$-approximation of social welfare in 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 (Flammini et al., 2021b). However, there exists a *randomized* online matching algorithm for MWM on unweighted graphs (i.e., maximum cardinality matching) that beats a competitive ratio of $\frac{1}{2}$ (Gamlath et al., 2019), i.e., achieves a competitive ratio of $\frac{1}{2} + 2\epsilon^*$ for some constant $\epsilon^* > 0$. We can apply Theorem 1 to conclude that randomization can be utilized to beat the best deterministic algorithm in this case.

Corollary 1. *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 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 tree (while other valuations are sufficiently negative).

Theorem 2. *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. □

Interestingly, negative results for MWM are usually essentially[6] achieved on the tree domain (Badanidiyuru Varadaraja, 2011; Bullinger and Romen, 2023), 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, Theorems 1 and 2 leave a gap of a factor of 2. As we will see, closing this gap can take significant effort.

5 FHGs 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 an online matching algorithm achieving a competitive ratio of $\frac{1}{3+2\sqrt{2}}$ (McGregor, 2005; Bullinger and Romen, 2023).[7] We can apply Theorem 1 to obtain an algorithmic guarantee for FHGs.

Theorem 3. *There exists an 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 (Badanidiyuru Varadaraja, 2011). By Theorem 2, no 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 3.

We illustrate here the main ideas for its proof and defer the full proof to Section B. The proof technique is similar to the proof by Badanidiyuru Varadaraja (2011) 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.

Theorem 4. *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

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

where $\beta > \frac{1}{3+2\sqrt{2}}$. Such a sequence of numbers does not exist (Badanidiyuru Varadaraja, 2011).

[6]These constructions usually contain 0-weights, which can be replaced with large negative weights.

[7]McGregor (2005) achieves this competitive ratio in the much related edge arrival model. In the full version of their paper, Bullinger and Romen (2023) showed that it is preserved in a vertex arrival model.

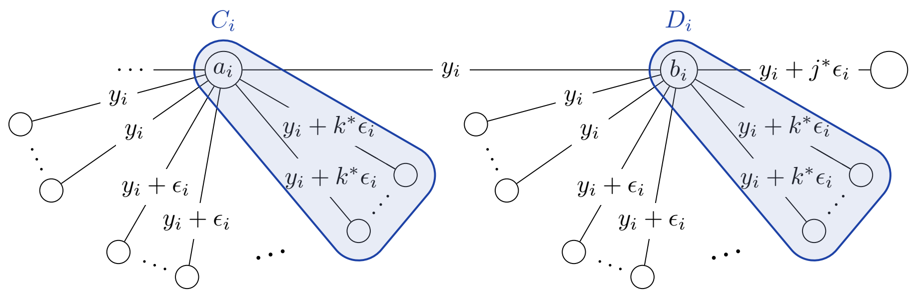

Figure 1: Illustration of Phase i in the construction of the adversarial instance in the proof of Theorem 4. Each star attached to a_i and b_i contains ℓ_i leaves.

The adversarial instance is established in phases, and in each phase, we determine a new element of a sequence $(y_i)_{i\in\mathbb{N}}$ that satisfies an inequality of the type of Inequality 1.[8]

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 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 $y_i + j^*\epsilon_i$. 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 $1 \le j \le i$, 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 $\beta = \gamma + \frac{1}{6+4\sqrt{2}}$. □

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

6 FHGs with Random Arrival

Based on Section 4, a reasonable strategy to obtain good online algorithms for FHGs is to consider good online matching algorithms. For the matching setting under random arrival, Ezra et al. (2022) provide an algorithm that achieves a competitive ratio of $\frac{5}{12} - \mathcal{O}(\frac{1}{n})$ if the algorithm has access to the number of arriving agents n. Knowledge of n is relevant 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 $k = 3$, one obtains an online matching algorithm that is $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$-competitive. We obtain the following theorem.

Theorem 5. *There exists a randomized online matching algorithm with a competitive ratio under random arrival of at least* $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$.

By applying Theorem 1, we can interpret this algorithm as a coalition formation algorithm, which implies the following corollary.

Corollary 2. *There exists a randomized online coalition formation algorithm with a competitive ratio under random arrival of at least* $\frac{1}{6} - \mathcal{O}(\frac{1}{n})$.

Ezra et al. (2022) show that the competitive ratio of their matching algorithm for known n is asymptotically optimal, i.e., no algorithm achieves a competitive ratio of more than $\frac{5}{12}$. However, if n is unknown, a competitive ratio of $\frac{5}{12}$ is off limits. As we show next, a competitive ratio of $\frac{1}{3}$ is asymptotically optimal in the matching domain. Our proof leverages the approach by Ezra et al. (2022) in a much enhanced way: while their worst-case instances is a family of complete graphs in which weights are drawn in a specific way, we need the interplay of two sets of instances whose positive edges form stars and double stars. Moreover, we want our result to hold on the tree domain so that we can apply Theorem 2. However, while it is comparably easy to use large negative weights in a setting with deterministic arrival if specifically tailored large negative edges arrive early in a random setting, we need to be careful that algorithms cannot infer useful information from the edge weights, such as the number of agents or the weight of best matchings. Hence, besides the positive weights, we must draw negative weights from an infinite set. We defer the proofs of intermediary lemmas to Section C.

Theorem 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.

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, and an error threshold ϵ, as specified below. 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 $x > t_B + 2$ and let $\epsilon > 0$ be a constant with $\epsilon \leq \frac{1}{2}$. Let $N = \{a, b\} \cup \{d_i : i \in I\} \cup \{d_j : j \in J\}$ be the set of agents.

First, we define a *star instance* $S^{x,\epsilon}_{I,J}$ by setting the following symmetric valuations:[9] 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* $B^{x,\epsilon}_{I,J}$ 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 $\left(\frac{1}{\epsilon}\right)^{t_B+1}$.

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 $\{d_j : j \in J\}$. 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, 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. Hence, since $\epsilon \leq \frac{1}{2}$, we have that the sum of 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 an algorithm ALG, 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 1. *If there exists no algorithm for matching the maximum weight edge with a competitive ratio of more than* $\frac{1}{3}$*, then there exists no online matching algorithm on the tree domain with a competitive ratio of more than* $\frac{1}{3}$.

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

Hence, in the following, we prove the nonexistence of such algorithms for matching the maximum weight edge. We can assume that all instances are defined for the same, i.e., fixed and sufficiently small ϵ. 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. (2022) 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 2. *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 and matches the maximum weight edge with at least 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 3. *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 3. Finally, we show that algorithmic decisions can be made independently of b and agents associated with J.

Lemma 4. *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 3 and 4 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_{ALG}-competitive for matching the maximum weight edge with $c_{ALG} > \frac{1}{3}$. Its matching parameters are given by a family of functions $f_k \colon \binom{\mathbb{N}}{k} \to [0,1]$, where $k \in \mathbb{N}$. For each k, and subset $I^x \subseteq \binom{\mathbb{N}}{k}$, let $x' := \max I^x$. The value $f_k(I^x)$ 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 $I^x \setminus \{x'\}$, and $x = x' + 2$ is the parameter for negative edges. Note that the condition on x ensures that $x > t_S + 2$.

Ideally, we would like a situation where the matching decision only depends on the size of the set of agents that thus far has arrived and who can match with a. While this can differ hugely for different algorithms, we can, however, prove that we can find an infinite set such that the matching probability within sets of the same size is within an arbitrarily small error window. The proof of this statement is given by Ezra et al. (2022, Claim 4.3), based on an application of the infinite version of Ramsey's theorem (Ramsey, 1930).

Lemma 5 (Ezra et al., 2022)**.** *Let $k \in \mathbb{N}$ and $\delta > 0$. Consider a collection of set functions $f_i \colon \binom{\mathbb{N}}{i} \to [0,1]$, $i \in [k]$. Then there exists an infinite set $T \subseteq \mathbb{N}$ and constants $p_1, \dots, p_k \in [0,1]$ such that, for all $i \in [k]$ and $T' \in \binom{T}{i}$, it holds that $f_i(T') = p_i + \mathcal{O}(\delta)$.*

By Lemma 5, we find an infinite set $T \subseteq \mathbb{N}$ and constants $p_1, \dots, p_k \in [0,1]$ such that, for all $i \in [k]$ and $T' \in \binom{T}{i}$, it holds that $p_i = f_i(T') + \mathcal{O}(\delta)$. Now, consider a star instance $S \in \mathcal{S}$ based on parameters I, J, and x (at this point, ϵ is irrelevant) such that $I \cup J \cup \{x-2\} \subseteq T$ and $|I| = k-1$. By assumption, $p_k = f_k(I \cup \{x-2\}) + \mathcal{O}(\delta)$. We define $h_k := \mathbb{P}(\bigcup_{i \in I}(\{a, d_i\} \in ALG(S)))$, i.e., the probability to match a. The key insight is to estimate this quantity.

Lemma 6. *It holds that $h_k > \frac{2}{3} - \frac{2}{3k} \pm \mathcal{O}(\delta)\frac{2-h_{k-1}}{k}$.*

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 $B \in \mathcal{B}$ such that $I \cup J \cup \{x-2\} \subseteq T$, where x defines its negative weights, and assume that $|I| = |J|$. As usual, the number of agents is n, i.e., $n = 2 + |I| + |J|$. Fix a constant $i \in \{2, \dots, |I| + |J| + 2\}$. Assume that the ith agent to arrive is either a or b and that the other agent among $\{a, b\}$ is already present. Without loss of generality, we assume that a arrives first and b arrives at step i. Otherwise, we swap the names of a,b, and I, J. In this case, at time $i-1$ the set of present agents is $\{a\} \cup I' \cup J'$ where $I' \subseteq I$, $J' \subseteq J$. Clearly, $S^{x,\epsilon}_{I',J'}$ is a star instance.

Let $k := |I'|$. The algorithm must match $\{a, b\}$ in step i to match the maximum weight edge, but this is only possible if a remains unmatched in $S^{x,\epsilon}_{I',J'}$. We have shown that the probability that a is matched in $S^{x,\epsilon}_{I',J'}$ is $h_k > \frac{2}{3} - \frac{2}{k} \pm \mathcal{O}(\delta)\frac{2-h_{k-1}}{k}$. Consequently, the probability that a is unmatched in $S^{x,\epsilon}_{I',J'}$ is $1 - h_k < \frac{1}{3} + \frac{2}{3k} \pm \mathcal{O}(\delta)\frac{2-h_{k-1}}{k}$. We now know that there exist $\delta > 0$ and $N \in \mathbb{N}$ such that $\frac{2}{3k} \pm \mathcal{O}(\delta)\frac{2-h_{k-1}}{k} \le \frac{1}{3}\left(c_{ALG} - \frac{1}{3}\right)$ for all $k \ge N$. Note that this works because h_{k-1} is a probability, so we can, for instance, choose $\delta = \frac{1}{k}$. Let X be the random variable that counts the number of agents from I that arrive before b. Note that there are $\binom{k}{i} i!(k-i)!$ orders in which i elements from I arrive before b. Hence, we have that

$$\begin{aligned}
&\mathbb{P}(X < N) = 2 \cdot \mathbb{P}(X < N \mid \sigma^{-1}(a) < \sigma^{-1}(b)) \\
&\quad \cdot \mathbb{P}(\sigma^{-1}(a) < \sigma^{-1}(b)) = \mathbb{P}(X < N \mid \sigma^{-1}(a) < \sigma^{-1}(b)) \\
&= \frac{1}{(k+1)!} \sum_{i=1}^{N-1} \binom{k}{i} i!(k-i)! \\
&= \frac{1}{(k+1)!} \sum_{i=1}^{N-1} k! = \frac{N-1}{k+1}.
\end{aligned}$$

This tends to 0 for n tending to infinity. In the first line where we use symmetry among a and b, we use that $|I| = |J|$. Therefore, there exists $N' \geq N$ such that $\mathbb{P}(X < N) \leq \frac{1}{3}\left(c_{ALG} - \frac{1}{3}\right)$ for all $n \geq N'$. For $n \geq N'$, we obtain that the probability of $\{a, b\}$ being matched in B in a uniformly random arrival order is bounded by

$$\begin{aligned}
&\mathbb{P}(X < N) + \mathbb{P}(X \geq N)\left(\frac{1}{3} + \frac{2}{3N}\right) \\
&\leq \frac{1}{3}\left(c_{ALG} - \frac{1}{3}\right) + \left(\frac{1}{3} + \frac{1}{3}\left(c_{ALG} - \frac{1}{3}\right)\right) \\
&\leq \frac{1}{3} + \frac{2}{3}\left(c_{ALG} - \frac{1}{3}\right) < c_{ALG}.
\end{aligned}$$

This contradicts our assumption that ALG was c_{ALG}-competitive. □

Combining Theorem 6 with Theorem 2, we conclude that no online coalition formation algorithm has a competitive ratio under random arrival of more than $\frac{1}{3}$.

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

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 adversarial agent arrival model. Designing good online coalition formation algorithms is deeply related to designing good online matching algorithms. It is possible to leverage matching algorithms with little welfare loss, while limitations for matching algorithms can be preserved if they hold on the tree domain.

In the coalition dissolution model, we showed that the optimal competitive ratio is $\frac{1}{6+4\sqrt{2}}$. Moreover, under random arrival, without the power to dissolve coalitions, we proved a tight bound of $\frac{1}{3}$ on the competitive ratio of any algorithm in the matching domain for an unknown number of agents. This then directly implies an $\frac{1}{3}$ upper bound for the FHG domain. Furthermore, the obtained matching algorithm is $\frac{1}{6}$-competitive in the FHG domain. Closing the gap for FHGs in the random arrival model remains an open problem.

Another future direction would be to study online modified fractional hedonic games, which differ from FHGs in that the sum of valuation is divided by the coalition size minus one (Olsen, 2012). We remark that our upper bound of $\frac{1}{3}$ on the competitive ratio under random arrival applies to this setting, too. Finally, similar to the work by Bullinger and Romen (2024), it would be interesting to study stability in online FHGs.

References

H. Aziz and R. Savani. Hedonic games. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 15. Cambridge University Press, 2016.

H. Aziz, S. Gaspers, J. Gudmundsson, J. Mestre, and H. Täubig. Welfare maximization in fractional hedonic games. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 461–467, 2015.

H. Aziz, F. Brandl, F. Brandt, P. Harrenstein, M. Olsen, and D. Peters. Fractional hedonic games. *ACM Transactions on Economics and Computation*, 7(2): 1–29, 2019.

A. Badanidiyuru Varadaraja. Buyback problem - approximate matroid intersection with cancellation costs. In *Proceedings of the 38th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 379–390, 2011.

Z. Baranyai. On the factorization of the complete uniform hypergraph. *Infinite and finite sets*, 1973.

V. Bilò, A. Fanelli, M. Flammini, G. Monaco, and L. Moscardelli. On the price of stability of fractional hedonic games. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1239–1247, 2015.

V. Bilò, A. Fanelli, M. Flammini, G. Monaco, and L. Moscardelli. Nash stable outcomes in fractional hedonic games: Existence, efficiency and computation. *Journal of Artificial Intelligence Research*, 62:315–371, 2018.

A. Bogomolnaia and M. O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

F. Brandl, F. Brandt, and M. Strobel. Fractional hedonic games: Individual and group stability. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1219–1227, 2015.

F. Brandt and M. Bullinger. Finding and recognizing popular coalition structures. *Journal of Artificial Intelligence Research*, 74:569–626, 2022.

M. Bullinger. Pareto-optimality in cardinal hedonic games. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 213–221, 2020.

M. Bullinger and R. Romen. Online coalition formation under random arrival or coalition dissolution. In *Proceedings of the 31st European Symposium on Algorithms (ESA)*, pages 27:1–27:18, 2023.

M. Bullinger and R. Romen. Stability in online coalition formation. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9537–9545, 2024.

M. Bullinger, E. Elkind, and J. Rothe. Cooperative game theory. In J. Rothe, editor, *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 3, pages 139–229. Springer, 2024.

M. Bullinger, V. Chatziafratis, and P. Shahkar. Welfare approximation in additively separable hedonic games. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2025. Forthcoming.

J. H. Drèze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4):987–1003, 1980.

T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang. 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)*, pages 1148 – 1177, 2022.

M. Flammini, B. Kodric, G. Monaco, and Q. Zhang. Strategyproof mechanisms for additively separable and fractional hedonic games. *Journal of Artificial Intelligence Research*, 70:1253–1279, 2021a.

M. Flammini, G. Monaco, L. Moscardelli, M. Shalom, and S. Zaks. On the online coalition structure generation problem. *Journal of Artificial Intelligence Research*, 72:1215–1250, 2021b.

M. Flammini, B. Kodric, and G. Varricchio. Strategyproof mechanisms for friends and enemies games. *Artificial Intelligence*, 302:103610, 2022.

B. Gamlath, M. Kapralov, A. Maggiori, O. Svensson, and D. Wajc. Online matching with general arrivals. In *Proceedings of the 60th Symposium on Foundations of Computer Science (FOCS)*, pages 26 – 37, 2019.

Z. Huang, Z. G. Tang, and D. Wajc. Online matching: A brief survey. *ACM SIGecom Exchanges*, 22(1):135–158, 2024.

C. Kaklamanis, P. Kanellopoulos, and K. Papaioannou. The price of stability of simple symmetric fractional hedonic games. In *Proceedings of the 9th International Symposium on Algorithmic Game Theory (SAGT)*, volume 9928 of *Lecture Notes in Computer Science (LNCS)*, pages 220–232. Springer-Verlag, 2016.

T. Kesselheim, K. Radke, A. Tönnis, and B. Vöcking. 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), pages 589–600. Springer-Verlag, 2013.

A. McGregor. 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), pages 170–181, 2005.

M. Olsen. On defining and computing communities. In *Proceedings of the 18th Computing: The Australasian Theory Symposium (CATS)*, volume 128 of *Conferences in Research and Practice in Information Technology (CRPIT)*, pages 97–102, 2012.

F. P. Ramsey. On a problem of formal logic. *Proceedings of the London Mathematical Society*, s2-30(1):264–286, 1930.

D. Ray. *A Game-Theoretic Perspective on Coalition Formation.* Oxford University Press, 2007.

Appendix

In the appendix, we present additional material and missing proofs.

A Simple proof of Theorem 1

In this section, we show an alternative proof for Flammini et al. (2021a, Theorem 1). The reason why we do so is twofold. First, our approach gives a deeper insight into the connection between matching and coalition formation. Second, our proof is built upon simple general ideas that might be of independent interest. Consequently, our proof is more straightforward on both a formal and intuitive level.

We use two ideas from matching. First, the social welfare of the global maximum weight matching is at least as large as the social welfare of the union of maximum weight matchings on an arbitrary partition of the agents. Second, the social welfare of a maximum weight matching is at least as large as that of a random matching. We combine these ideas and additionally show that a random matching is a $\frac{1}{2}$ approximation of the social welfare to conclude the proof. We start with a lemma that proves the last step. To this end, we define a random matching as a matching drawn uniformly from a set of matchings that partitions the set of agent pairs. If the number of agents is even, it follows from a special case of Baranyai (1973) that such a set exists. Otherwise, we first remove one agent uniformly at random.

Lemma 7. *The expected social welfare of a random matching is a $\frac{1}{2}$-approximation of the social welfare of a coalition.*

Proof. The social welfare of a coalition C is defined as

$$\begin{aligned} \mathcal{SW}(C) &= \sum_{i \in C} u_i(C) \\ &= \sum_{i \in C} \sum_{j \in C \setminus \{i\}} \frac{v_i(j)}{|C|} \\ &= (|C|-1) \sum_{i \in C} \frac{1}{|C|} \sum_{j \in C \setminus \{i\}} \frac{1}{|C|-1} v_i(j) \\ &= (|C|-1) \mathbb{E}_{i \sim C} \left[\mathbb{E}_{j \sim C \setminus \{j\}} \left[v_i(j) \right] \right]. \end{aligned} \tag{2}$$

Where the expectation is taken with respect to the uniform distribution. This equation shows that the welfare of a coalition C is $|C|-1$ times the expected valuation of a random agent pair from C, which is also equivalent to the average weight of a valuation from a pair of agents of C. One can easily show that the expected social welfare of a random matching within C is $\lfloor \frac{|C|}{2} \rfloor$ times the expected valuation of a random agent pair from C. Therefore,

$$
\begin{aligned}
&\left\lfloor \frac{|C|}{2} \right\rfloor \mathbb{E}_{i\sim C}\left[\mathbb{E}_{j\sim C\setminus\{j\}}\left[v_i(j)\right]\right] \\
\geq & \frac{|C|-1}{2}\mathbb{E}_{i\sim C}\left[\mathbb{E}_{j\sim C\setminus\{j\}}\left[v_i(j)\right]\right] \\
\overset{\text{Equation (2)}}{=} & \frac{1}{2}\mathcal{SW}(C).
\end{aligned}
$$

We have shown the expected social welfare of a random matching is a $\frac{1}{2}$-approximation of the social welfare of a coalition. □

Next, we present our proof that maximum weight matchings are a $\frac{1}{2}$ approximation of the social welfare of the optimal partition.

Proof of Theorem 1. Let π^* be an optimal partition. Furthermore, for a set of agents $C \subseteq N$, let $\mu^*(C)$ denote an MWM on C and $\mu_R(C)$ denote a random matching on C. We compute,

$$
\begin{aligned}
\mathcal{SW}\left(\mu^*(N)\right) &\geq \sum_{C\in\pi^*} \mathcal{SW}\left(\mu^*(C)\right) \\
&\geq \sum_{C\in\pi^*} \mathbb{E}\left[\mathcal{SW}\left(\mu_R(C)\right)\right] \\
&\overset{\text{Lemma 7}}{=} \sum_{C\in\pi^*} \frac{\mathcal{SW}(C)}{2} \\
&= \frac{\mathcal{SW}(\pi^*)}{2}.
\end{aligned}
$$

The first inequality holds because the social welfare of the global maximum weight matching is at least as much as the sum of the social welfare of the maximum weight matching on every coalition in the optimal partition. The second inequality holds because the social welfare of maximum weight matching is at least as good as that of random matchings. Finally, we apply Lemma 7 to show that maximum weight matchings are a $\frac{1}{2}$ approximation for the social welfare of FHGs. □

B Full proof of Theorem 4

Our proof of Theorem 4 relies on a similar idea as the proof by Badanidiyuru Varadaraja (2011), 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 (2011). 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 8 (Badanidiyuru Varadaraja (2011))**.** *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*

$$x_i \geq \beta \left(x_{i+1} + \sum_{j=1}^{i+1} x_j \right). \tag{3}$$

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

Lemma 9. *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 lemmas 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 4. *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. Let

$$\beta := 2\left(c + \frac{1}{2}(\gamma - c)\right) = \gamma + c, \tag{4}$$

i.e., it holds that $\beta > 2c = \frac{1}{3+2\sqrt{2}}$. We will eventually derive a contraction to Lemma 8 by constructing a sequence for this β.

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

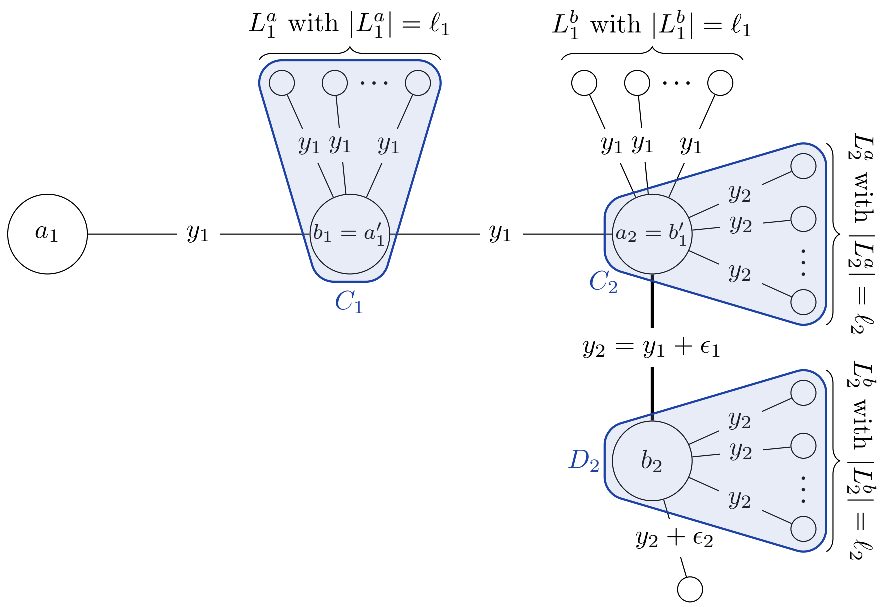

Figure 2: Illustration of the construction in the proof of Theorem 4 for an exemplary algorithm ALG. We display all positive valuations. The remaining valuations within the leaf sets L_1^a, L_1^b, L_2^a, and L_2^b are zero, and all other valuations are large negative numbers. We start with two agents, 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.

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

$$\epsilon_i := \frac{\gamma - c}{2\gamma} 2^{-i} \quad \text{and} \quad \ell_i := \left\lceil \frac{1 - \epsilon_i}{\epsilon_i} \right\rceil. \tag{5}$$

The definition of ℓ_i immediately implies that

$$\frac{\ell_i}{\ell_i + 1} \geq 1 - \epsilon_i. \tag{6}$$

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

$$y_i := v(a_i, b_i). \tag{7}$$

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 9.

We now start the second part of Phase i. New agents for potential star coalitions with slightly higher valuations arrive in this phase. We 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 $j \geq 1$, 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 arrive agents in $L_i^{a,j}$, and a sufficiently large negative valuation for all other agents. 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 $L_i^{a,j}$, 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 9, we obtain

$$\mathcal{SW}(C_i) = \mathcal{SW}(D_i) = 2\frac{\ell_i}{\ell_i + 1}(y_{i+1} - \epsilon_i). \tag{8}$$

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

$$\begin{aligned}
y_i &\geq \gamma \cdot \mathcal{SW}(\pi_i) \\
&= \gamma \left(\mathcal{SW}(D_i) + \sum_{j=1}^{i} \mathcal{SW}(C_j) \right) \\
&\overset{(8)}{=} \gamma \left(2\frac{\ell_i}{\ell_i + 1}(y_{i+1} - \epsilon_i) + \sum_{j=1}^{i} 2\frac{\ell_j}{\ell_j + 1}(y_{j+1} - \epsilon_j) \right) \\
&\overset{(6)}{\geq} \gamma \left(2(1 - \epsilon_i)(y_{i+1} - \epsilon_i) + \sum_{j=1}^{i} 2(1 - \epsilon_j)(y_{j+1} - \epsilon_j) \right) \\
&\geq \gamma \left(2(y_{i+1} - 2y_{i+1}\epsilon_i) + \sum_{j=1}^{i} 2(y_{j+1} - 2y_{j+1}\epsilon_j) \right) \\
&\geq \gamma \left(2(y_{i+1} - 2y_{i+1}\epsilon_i) + \sum_{j=1}^{i} 2(y_{j+1} - 2y_{i+1}\epsilon_j) \right) \\
&= 2\gamma \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right) - 2\gamma y_{i+1} \left(\epsilon_i + \sum_{j=1}^{i} \epsilon_j \right) \\
&\overset{(4),\,(5)}{=} (\beta + (\gamma - c)) \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right) \\
&\quad - 2\gamma y_{i+1} \left(\frac{\gamma - c}{2\gamma} 2^{-i} + \sum_{j=1}^{i} \frac{\gamma - c}{2\gamma} 2^{-j} \right) \\
&\geq \beta \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right) \\
&\quad + (\gamma - c)y_{i+1} - (\gamma - c)y_{i+1} \left(2^{-i} + \sum_{j=1}^{i} 2^{-j} \right)
\end{aligned}$$

$$= \beta \left(y_{i+1} + \sum_{j=1}^{i} y_{j+1} \right).$$

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

$$\begin{aligned} x_i = \frac{y_{i+1}}{y_2} &\geq \frac{1}{y_2} \beta \left(y_{i+2} + \sum_{j=2}^{i+2} y_j \right) \\ &= \beta \left(\frac{y_{i+2}}{y_2} + \sum_{j=2}^{i+2} \frac{y_j}{y_2} \right) = \beta \left(x_{i+1} + \sum_{j=2}^{i+2} x_{j-1} \right) \\ &= \beta \left(x_{i+1} + \sum_{j=1}^{i+1} x_j \right). \end{aligned}$$

Hence, we have constructed the desired sequence and obtained a contradiction by applying Lemma 8. ☐

C Missing proofs in Section 6

In this section, we prove auxiliary lemmas in the proof of Theorem 6. We start with the proof of Lemma 1. 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.

Lemma 10. *For every online matching algorithm ALG, it holds that* $\inf_{S \in \mathcal{S}} \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) \geq c_{ALG} - \epsilon$ *for every* $\epsilon > 0$.

Proof. Consider some star instance $S \in \mathcal{S}$. Then, by definition of the competitive ratio,

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

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

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

$$= \sum_{i \in I \setminus \{t\}} \mathbb{P}(\{a, d_i\} \in ALG(S)) \left(\frac{1}{\epsilon}\right)^i$$
$$+ \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) \left(\frac{1}{\epsilon}\right)^{t_S}$$

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

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

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 one. Since $S \in \mathcal{S}$ was chosen arbitrarily, we obtain $\inf_{S \in \mathcal{S}} \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) \geq c_{ALG} - \epsilon$. □

Next, we show that $c_{ALG} - 2\epsilon$ is a lower bound on the probability with which ALG matches the two centers in bi-star instances. The proof is similar to that of Lemma 10.

Lemma 11. *For every online matching algorithm ALG, it holds that* $\inf_{B \in \mathcal{B}} \mathbb{P}(\{a, b\} \in ALG(B)) \geq c_{ALG} - 2\epsilon$ *for every* $\epsilon > 0$.

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

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

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

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

$$
\begin{aligned}
&+ \sum_{j \in J} \mathbb{P}(\{b, d_j\} \in ALG(B)) v(b, d_j) \\
&+ \mathbb{P}(\{a, b\} \in ALG(B)) v(a, b) \\
= &\sum_{i \in I} \mathbb{P}(\{a, d_i\} \in ALG(B) \left(\frac{1}{\epsilon}\right)^i \\
&+ \sum_{j \in J} \mathbb{P}(\{b, d_j\} \in ALG(B)) \left(\frac{1}{\epsilon}\right)^j \\
&+ \mathbb{P}(\{a, b\} \in ALG(B)) \left(\frac{1}{\epsilon}\right)^{t+1}
\end{aligned}
$$

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 $\left(\frac{1}{\epsilon}\right)^{t+1} > 0$, we get

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

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

We can combine Lemmas 10 and 11 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 10 and 11, 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 1. *If there exists no algorithm for matching the maximum weight edge with a competitive ratio of more than $\frac{1}{3}$, then there exists no online matching algorithm on the tree domain with a competitive ratio of more than $\frac{1}{3}$.*

Proof. Assume that there is a c-competitive online matching algorithm ALG on the tree domain with a competitive ratio of $c > \frac{1}{3}$. Define $\epsilon := \frac{1}{3}\left(c - \frac{1}{3}\right)$ and consider $c' = c - 2\epsilon > \frac{1}{3}$. By Lemmas 10 and 11, ALG is c'-competitive for matching the maximum weight edge. □

Next, we prove our lemma concerning history independence.

Lemma 3. *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 2. 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. 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 $H(A)$ 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 4. *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.*

Proof. Consider an algorithm ALG restricted as per Lemma 2. 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. □

Next, we prove the lemma about h_k.

Lemma 6. *It holds that* $h_k > \frac{2}{3} - \frac{2}{3k} \pm \mathcal{O}(\delta)\frac{2-h_{k-1}}{k}$.

Proof. To prove the lemma, we additionally define $r_k := \mathbb{P}(\{a, d_{t_S}\} \in ALG(S))$, i.e., the probability to match a with d_{t_S}. We now show recursive formulas. To this end, we partition all arrival orders of $\{a\} \cup \{d_i \colon i \in I\}$, i.e., of the agents relevant to matching, into three sets based on the last arriving agent. The first two are the arrival orders $\sigma \in \Sigma(\{a\} \cup \{d_i \colon i \in I\})$ 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 $\mathbb{P}(\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S}) = \frac{k-2}{k}$.

$$\begin{aligned}
h_k &= \mathbb{P}(\bigcup_{i\in I}(\{a, d_i\} \in ALG(S))) \\
&= \mathbb{P}(\bigcup_{i\in I}(\{a, d_i\} \in ALG(S))|\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S}) \\
&\quad \cdot \mathbb{P}(\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S}) \\
&+ \mathbb{P}(\bigcup_{i\in I}(\{a, d_i\} \in ALG(S))|\sigma(k) = d_{t_S})\mathbb{P}(\sigma(k) = d_{t_S}) \\
&+ \mathbb{P}(\bigcup_{i\in I}(\{a, d_i\} \in ALG(S))|\sigma(k) = a)\mathbb{P}(\sigma(k) = a) \\
&= \frac{k-2}{k}h_{k-1} + \frac{1}{k}(h_{k-1} + (1 - h_{k-1})(p_k - \mathcal{O}(\delta))) \\
&\quad + \frac{1}{k}(p_k - \mathcal{O}(\delta)) \\
&= \frac{k-1}{k}h_{k-1} - \frac{p_k}{k}h_{k-1} + \frac{2p_k}{k} - \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k}
\end{aligned}$$

Whenever we apply the recursion, we use that for all $i \in I$, it holds that $(I \setminus \{i\}) \cup \{x - 2\} \subseteq T$. Furthermore, we have $h_2 = p_2 - \mathcal{O}(\delta)$.

Moreover, we have

$$\begin{aligned}
r_k &= \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)) \\
&= \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)|\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S}) \\
&\quad \cdot \mathbb{P}(\sigma(k) \neq a \wedge \sigma(k) \neq d_{t_S}) \\
&\quad + \mathbb{P}(\{a, d_{t_S}\} \in ALG(S)|\sigma(k) = d_{t_S})\mathbb{P}(\sigma(k) = d_{t_S})
\end{aligned}$$

$$
\begin{aligned}
&+ \mathbb{P}(\{a, d_{t_S}\} \in ALG(S) | \sigma(k) = a)\mathbb{P}(\sigma(k) = a) \\
&= \frac{k-2}{k} r_{k-1} + \frac{1}{k}(p_k - \mathcal{O}(\delta))(1 - h_{k-1}) \\
&+ \frac{1}{k}(p_k - \mathcal{O}(\delta)) \\
&= \frac{k-2}{k} r_{k-1} - \frac{p_k}{k} h_{k-1} + \frac{2p_k}{k} - \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k}
\end{aligned}
$$

In addition, it holds that $r_2 = p_2 - \mathcal{O}(\delta)$ since if it matches, in this case, it matches the optimal edge.

Next, we compute $h_k - r_k$, i.e., the probability of matching a suboptimal valuation in a star. The terms $\mathcal{O}(\delta)\frac{2-h_{k-1}}{k}$ do not cancel out because, in general, they can be different.

$$
\begin{aligned}
h_k - r_k &= \frac{k-1}{k} h_{k-1} - \frac{p_k}{k} h_{k-1} + \frac{2p_k}{k} - \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k} \\
&\quad - \frac{k-2}{k} r_{k-1} + \frac{p_k}{k} h_{k-1} - \frac{2p_k}{k} + \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k} \\
&= \frac{k-1}{k} h_{k-1} - \frac{k-2}{k} r_{k-1} \pm \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k}
\end{aligned} \tag{9}
$$

Thus,

$$
\begin{aligned}
h_k &= \frac{k-1}{k} h_{k-1} + r_k - \frac{k-2}{k} r_{k-1} \pm \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k} \\
&= \left(\sum_{i=2}^{k} \frac{i}{k} r_i + \frac{i-2}{k} r_{i-1}\right) \pm \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k} \\
&= r_k + \left(\sum_{i=2}^{k-1} \frac{i}{k} r_i + \frac{i-1}{k} r_i\right) \pm \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k} \\
&= r_k + \left(\sum_{i=2}^{k-1} \frac{1}{k} r_i\right) \pm \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k} \\
&\geq c_{ALG} + \frac{k-2}{k} c_{ALG} \pm \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k} \\
&= 2c_{ALG} - \frac{2}{k} c_{ALG} > \frac{2}{3} - \frac{2}{3k} \pm \mathcal{O}(\delta)\frac{2 - h_{k-1}}{k}.
\end{aligned}
$$

□