SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Online Coalition Formation for Fractional Hedonic Games

Alexander Timothy Schlenga

SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Online Coalition Formation for Fractional Hedonic Games

Online Koalitionsbildung bei Fraktionalen Hedonischen Spielen

Author:	Alexander Timothy Schlenga
Supervisor:	Prof. Dr. Felix Brandt
Advisor:	René Romen, M.Sc.
Submission Date:	22.07.2024

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Garching bei München, 22.07.2024 Alexander Timothy Schlenga

Acknowledgments

I want to thank all members of the computational social choice and algorithmic game theory group. First of all, Felix Brandt taught two excellent courses that awakened my interest in the field and provided me with the necessary foundations to work on this thesis. Felix Brandt, Chris Dong, Fabian Frank, Matthias Greger, Patrick Lederer, and René Romen welcomed me cordially as part of their group. I had a great time working there. I also want to thank Martin Bullinger for fruitful discussions on my work. Special thanks go to René Romen for coming up with various interesting topic suggestions for my thesis, patiently waiting for me to finish another project of my studies that took longer than expected, and, in particular, for being a great advisor for this thesis although he was advising four other students at the same time.

Abstract

Fractional hedonic games (FHGs) are an appealing class of hedonic coalition formation games, in which the utility of an agent equals the average utility she ascribes to the members of her coalition. We study the problem of maximizing the social welfare in FHGs both in the offline and online setting. For the offline setting, where the problem is known to be NP-hard, we show that no FPTAS exists. For the adversarial arrival online model, with unrestricted utilities, it is known that no constant competitive ratio is achievable with a deterministic algorithm. Therefore, we investigate two other interesting models of online FHGs. In the first one, the agents arrive in a uniformly random order. For this setting, we provide an asymptotically $\frac{1}{6}$-competitive algorithm and prove an upper bound of $\frac{1}{3}$ on the possible asymptotic competitive ratio. In the second model, an algorithm may dissolve (at no cost) an existing coalition before assigning an arriving agent to a coalition. Here, we show a $\frac{1}{6+4\sqrt{2}}$-competitive algorithm and prove an upper bound of $\frac{1}{3+2\sqrt{2}}$ on the possible competitive ratio. For the adversarial arrival model, we show that a randomized algorithm achieves a better competitive ratio for symmetric simple FHGs than any deterministic algorithm. We also provide new results on online matching with random vertex arrival on general graphs, with an asymptotically $\frac{1}{3}$-competitive algorithm and an upper bound of $\frac{1}{3}$ on the possible asymptotic competitive ratio.

Contents

1 Introduction

When members of a society form or are assigned to groups, it is interesting to take a mathematical point of view on the matter—may it be in the field of artificial intelligence, social sciences, or other disciplines. The term "society" is to be understood very broadly here, and the respective members can be people as well as, e.g., firms or computer programs. In general, we can speak of the members of the society as *agents* and of the groups as *coalitions*. When a society splits into coalitions, we call the resulting assignment of agents a *partition*. The theory of cooperative games and coalition formation studies precisely this kind of scenarios.

The justifications for this mathematical perspective are manifold. Here, we only want to name a few. A widely accepted (though debatable) premise is that agents exhibit *rational* preferences over the partitions which can form. These preferences can have a comparable intensity. For instance, it is possible that interactions between agents yield an intrinsic amount of utility for them, and agents want to be part of coalitions within which they get the most utility out of interacting with the other coalition members. In other situations, the utilities can be extrinsic in the sense that an external observer assigns them to possible coalitions, fixed groups, or pairs of agents. Utilities are represented as rational numbers (e.g., the quantity of possible economic cooperation of firms or the inverse geographical distance of people's homes to their friends' homes, their work place, or their common sports club) or indicators (e.g., 1 for friendship and 0 for other people). But also models with agents expressing their preferred coalitions not via utilities but via rankings are of great importance. They come into play whenever the intensities of the agents' preferences are unknown, cannot be measured or expressed, or are generally of minor importance. Moreover, such preferences can render the treatment of coalition formation as a voting process more natural.

For many scenarios, a particularly appealing class of coalition formation games are *hedonic games* (Banerjee, Konishi, & Sönmez, 2001; Bogomolnaia & Jackson, 2002; Drèze & Greenberg, 1980). Their central—hedonic—aspect is that the preferences of an agent over the possible partitions depend only on the other members of the agent's own (prospective) coalition but not on the structure or members of other coalitions of the partition. However, even under this natural restriction, writing down the partition preferences of any agent may well take an exponential amount of space. From an algorithmic point of view, this is unfortunate. Therefore, a significant amount of research has been undertaken concerning hedonic games with inherently concise representations (of the agents' preferences). One way of achieving this is to derive an agent's preferences over coalitions from his preferences over other agents. We can, e.g., from each agent's perspective, assign a subjective utility to every other agent of the society. Then, the utility an agent receives from being in a certain coalition is determined by her respective utilities for the other agents in that coalition. This approach gives rise to—among others—the classes of additively separable hedonic games (ASHGs) (Banerjee, Konishi, & Sönmez, 2001; Bogomolnaia & Jackson, 2002), fractional hedonic games (FHGs) (Aziz, Brandl, Brandt, et al., 2019), and modified fractional hedonic games (MFHGs) (Elkind, Fanelli, & Flammini, 2016; Olsen, 2012).

In this work, we focus on FHGs. In an FHG, the utility an agent assigns to a coalition is the average utility she assigns to the members of that coalition (assuming a utility of 0 for herself). From a modeling point of view, they are an apt choice for various scenarios. In (network) clustering (see, e.g., Newman (2004) for a comparison of earlier methods), they have the appeal of rewarding strong connections between agents in the same coalition while preventing the formation of a single coalition containing all agents from being a trivially optimal solution, even under the restriction of allowing only positive utilities. So, FHGs could be used, e.g., to find communities of well connected people on social networks. Aziz, Brandl, Brandt, et al. (2019) demonstrate a model of the well-known game of Bakers and Millers as an FHG. The formation of political parties or coalitions can be nicely modelled with FHGs. If we consider politicians to express like-mindedness with other politicians via corresponding utilities, an FHG will model the situation that they seek to form coalitions of similar opinions. Now, political parties, in order to be successful, should neither be too small nor too large. With ASHGs as the modeling choice, the size of a coalition would not be taken into consideration, possibly leading to very few, very large parties. With MFHGs on the other hand, a solution maximizing the utility of all politicians could always be one in which the largest parties contain only three people (Bullinger, 2020). Besides political parties, also parties in the sense of social events, and other similar scenarios, can be modeled as FHGs. Imagine, people have to decide to which party they should go on a certain evening. At a party, people are influenced by the presence of the others. They like talking to their friends while not like talking to some other people. Then, they would like to attend the parties in such a way that the average enjoyability of talking to people at the same party is high. The fact that in FHGs we divide the sum of the individual utilities by the coalition size cannot only be explained by the taking an average utility. It is also possible that the utility of agents in a coalition scales with their share of a resource common to the coalition. Last but not least, FHGs are already interesting from a graph-theoretic perspective. Their nice aspect in this regard is that they represent a natural extension of the notion of matchings, the only difference being that in matchings only coalitions of size at most 2 are allowed.

1.1 Related Work

Offline FHG Setting Hedonic games have been first proposed by Banerjee, Konishi, and Sönmez (2001), and Bogomolnaia and Jackson (2002), taking up ideas and terminology by Drèze and Greenberg (1980). In their papers, they also introduce the subclass of ASHGs. Based on that idea of computing the individual utility for a coalition from utilities for single agents, Aziz, Brandl, Brandt, et al. (2019) came up with the concept of FHGs. Aziz and Savani (2016) give a survey on hedonic games. Several authors studied various notions of stability in FHGs (Aziz, Brandl, Brandt, et al., 2019; Bilò, Fanelli, Flammini, et al., 2015, 2018; Brandl, Brandt, & Strobel, 2015; Kaklamanis, Kanellopoulos, & Papaioannou, 2016). The study of social welfare maximization for FHGs has been initiated by Aziz, Gaspers, Gudmundsson, et al. (2015). In addition to examining algorithms for (utilitarian) social welfare, they consider maximization of egalitarian and Nash welfare. They prove NP-hardness of finding optimal partitions for the different objectives and give approximation algorithms with polynomial running time. Matching algorithms are shown to yield reasonable approximation ratios, in particular, a maximum weight matching (MWM) is at least a $\frac{1}{4}$-approximation of social welfare in general FHGs. That analysis was later improved and made tight by Flammini, Kodric, Monaco, and Zhang (2021) who prove that a MWM is even a $\frac{1}{2}$-approximation. In their paper, they furthermore propose to consider agents with the option to report their utilities untruthfully and study strategyproofness in FHGs and ASHGs.

Online FHG Setting Online scenarios and algorithms for FHGs and ASHGs were studied first by Flammini, Monaco, Moscardelli, et al. (2021)[1]. They investigate the model where agents arrive in an adversarial order. Besides standard FHGs and ASHGs, they consider settings with additional restrictions on the utility values or coalition size or number. They give lower bounds and upper bounds for deterministic algorithms on the achievable competitive ratio for maximizing the social welfare of a partition. Except for simple FHGs, the results can be seen as rather discouraging because the competitiveness depends on the utility values. More specifically, the best possible competitive ratio is $\frac{U_{min}}{4 \cdot U_{max}}$, where U_{min} and U_{max} are the minimal and maximal absolute value of non-zero utilities, respectively. The situation is similar for ASHGs. In order to find more positive results for ASHGs, Bullinger and Romen (2023) consider two relaxations of the online model (separately). A first relaxation takes power from the adversary by letting agents arrive in a uniformly random order instead of the worst possible. The second one gives more power to the algorithm by allowing it to dissolve an existing coalition (into singletons) before assigning a new agent. In both of the models they show how to get rid of the dependency on utilities in the competitive ratio. Furthermore, in another recent work, Bullinger and Romen (2024) study the stability and Pareto-optimality of solutions concepts for ASHGs with online adversarial agent arrival.

Online Matching The literature on online matching algorithms is vast. A recent survey is given by Huang, Tang, and Wajc (2024). Here, we only list the works which are the closest and most relevant to our setting. For unweighted graphs, Gamlath, Kapralov,

[1] Flammini, Monaco, Moscardelli, et al. (2021) analyze the model as a "coalition structure generation problem", i.e., instead of talking about hedonic games with agents, utilities, and social welfare they adopt a purely graph-theoretic perspective.

Maggiori, et al. (2019) give the currently best known approximative online algorithm for computing a maximum cardinality matching (MCM) with adversarial vertex arrival. Kesselheim, Radke, Tönnis, and Vöcking (2013) study MWM 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, Feldman, Gravin, and Tang (2022) propose an algorithm for approximating a MWM in general weighted graphs with random vertex arrival where the total number of vertices to arrive is known in advance. The algorithm starts with a sampling phase, i.e., the first half of the arriving vertices are simply left unmatched for the moment. In the second half, in each step, it computes what would be a locally (at the current time) optimal MWM. If the partner of the newly arrived vertex in that locally optimal MWM is not matched yet, the algorithm matches the two in its solution. Otherwise, the new vertex is left unmatched. They also show asymptotic tightness of that algorithm's competitive ratio by considering a family of graphs where all edge weights differ by a large factor, so that there is only one really valuable edge for a matching. Bullinger and Romen (2023) study online general vertex arrival for MWM under free dissolution.

1.2 Our Contribution

We study the maximization of social welfare for FHGs in both the offline and online setting. First, we tackle the open question of whether an approximation ratio of $\frac{1}{2}$ for maximizing social welfare offline in FHGs is the best guarantee achievable in polynomial time (Chapter 3). While we ultimately leave the question open, we give some indicators that if a polynomial-time algorithm with a better approximation ratio exists, it might be hard to find. Moreover, we make progress on the side of hardness by showing that (given P$\neq$NP) the problem does not admit a fully polynomial-time approximation scheme (FPTAS).

With our main focus being different online models for FHGs, we there start by showing that for symmetric simple FHGs with adversarial agent arrival, a randomized matching algorithm can achieve a better competitive ratio than the bound for deterministic algorithms given by Flammini, Monaco, Moscardelli, et al. (2021) (Chapter 4). Then, inspired by the results of Bullinger and Romen (2023) for ASHGs, we get rid of the competitive ratio's dependence on the utilities for general FHGs by modifying the setting—we investigate the random arrival (Chapter 5) and free dissolution (Chapter 6) variants of online FHGs and show how to obtain constant competitive ratios by using different online matching algorithms. We also provide upper bounds for the competitive ratio in both models.

Online random vertex arrival for MWM problems with an unknown total number of vertices, to the best of our knowledge, has not been covered by existing research yet. Hence, we point out how our results for FHGs apply to that problem and that the bounds become tight when only matchings are taken into consideration (Chapter 5).

Problem	Class	Lower Bound	Upper Bound
Adversarial FHG	General	$\frac{U_{min}}{4 \cdot U_{max}}$ (a,f)	$\frac{U_{min}}{4 \cdot U_{max}}$* (f)
	Simple	$\frac{1}{8} + \epsilon$ (Cor. 2)	?
	Symm. Simple	$\frac{1}{4} + 2\epsilon$ (Thm. 5)	?
F. Dissolution FHG	General	$\frac{1}{6+4\sqrt{2}}$ (Thm. 14)	$\frac{1}{3+2\sqrt{2}}$* (Thm. 15)
Random FHG	Symm. Simple	$\frac{1}{3} - \mathcal{O}(\frac{1}{n})$ (Thm. 10)	?
	General	$\frac{1}{6} - \mathcal{O}(\frac{1}{n})$ (Thm. 8)	$\frac{1}{3}$ (Thm. 12)
Random MWM	General	$\frac{1}{3} - \mathcal{O}(\frac{1}{n})$ (Cor. 4)	$\frac{1}{3}$ (Cor. 5)
Random MCM	General	$\frac{2}{3} - \mathcal{O}(\frac{1}{n})$ (Thm. 9)	?

Table 1.1: An overview of bounds for the competitive ratios for online FHGs and MWMs. U_{min} and U_{max} are the minimal and maximal absolute value of non-zero utilities, respectively. Entries "?" mean that only trivial upper bounds are known. Upper bounds marked with * only hold for deterministic algorithms. Results marked with (a) are by Aziz, Gaspers, Gudmundsson, et al. (2015) and those marked with (f) are by Flammini, Monaco, Moscardelli, et al. (2021).

Class	Approximation	Hardness
(Symmetric) FHG	$\frac{1}{2}$ (f)	No FPTAS (Thm. 1)
Symmetric Simple FHG	$\frac{1}{2}$ (f)	NP-hard (a)

Table 1.2: An overview of known approximation ratios and hardness results for offline FHGs. The problem for symmetric FHGs is the same as for general FHGs for the reason explained in Remark 1. Results marked with (f) are by Flammini, Kodric, Monaco, and Zhang (2021). Results marked with (a) are by Aziz, Gaspers, Gudmundsson, et al. (2015).

2 Preliminaries

We begin by introducing some mathematical standard notations. After that, we give the formal terms and definitions for FHGs and competitiveness needed for our results. We conclude the chapter by discussing a small example.

Notation For $i \in \mathbb{N}$, let $[i]$ denote the set of all natural numbers smaller or equal to it ($\{1, \ldots, i\}$). For a set S, let $\binom{S}{k}$ denote the set of all subsets of size k of S. We use $\uplus$ to denote the disjoint union of sets, i.e., $A \uplus B$ denotes the union of sets A and B where $A \cap B = \emptyset$. 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. By $\mathrm{I}(\cdot)$, we denote the indicator function. It takes a boolean argument and returns 1 if it is true and 0 otherwise.

Let N be a finite set of *agents* with $n := |N|$. A non-empty set $C \subseteq N$ is called a *coalition*. The set of coalitions containing a certain agent i is denoted by $\mathcal{N}_i := \{C \subseteq N \mid i \in C\}$. A set π of disjoint coalitions containing together exactly the members of N is a *partition* of N (we have $\biguplus_{C \in \pi} C = N$). For an agent $i \in N$ and a partition π, let $\pi(i)$ denote the coalition in π which i belongs to. Given a subset of agents $S \subseteq N$ and a partition π of N, let $\pi[S]$ denote the restriction of π to S, i.e., $\pi[S] := \{C \mid C \neq \emptyset \wedge \exists C' \in \pi : C = C' \cap S\}$.

Definition 1. *A hedonic game is a pair $(N, \precsim)$ where N is the set of agents and $\precsim$ a set of preferences $(\precsim_i)_{i \in N}$ of the agents. For agent i, $\precsim_i$ is a partial order on $\mathcal{N}_i$ stating his preferences over all possible coalitions of which he is a member.*

Let $u_i(j) \in \mathbb{Q}$ denote the utility of an agent i for another agent j. The preferences of i over her possible coalitions will now be defined by the average utility she assigns to the members of her coalition (while treating the utility assigned to herself as 0), i.e., the utility agent i assigns to being in a coalition C is

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

Definition 2. *A hedonic game is called an* FHG *if it can be represented by a complete directed weighted graph $G = (N, w)$ such that*

- *the vertices of G are the agents N,*
- *the edge weights are the utilities, i.e, for an edge $(i, j) \in N \times N$ (where $i \neq j$) we have*

$$w(i, j) = u_i(j),$$

- *and for any $i \in N$ and $C_1, C_2 \in \mathcal{N}_i$ we have*

$$C_1 \precsim_i C_2 \iff u_i(C_1) \leq u_i(C_2).$$

Note that for singleton coalitions the utility of its member is always 0. We call an FHG *symmetric* if for every pair of distinct agents $i, j \in N$, their utilities for each other are equal, formally $w(i, j) = w(j, i)$. Then, the game can be represented by an undirected graph. We call an FHG *simple* if for every pair of distinct agents $i, j \in N$, the utility of i for j is either 0 or 1, formally $w(i, j) \in \{0, 1\}$. Then, the game can be represented by an unweighted graph (by an unweighted graph we mean a complete graph with edge weights 1 and 0). A partition π of an FHG G is said to be *Pareto-improved* by a partition π' of G if all agents receive at least the same utility in π' as in π and at least one agent receives strictly more utility in π' than in π. A partition is called *Pareto-optimal* if no other partition Pareto-improves it.

Definition 3. *The* social welfare *of a partition π of G is the sum of the utilities of all agents for π.*

$$\mathcal{SW}_G(\pi) := \sum_{i \in N} u_i(\pi(i))$$

We can write this in an alternative form which makes calculations easier.

$$\begin{aligned}\sum_{i \in N} u_i(\pi(i)) &= \sum_{C \in \pi} \sum_{i \in C} u_i(C) = \sum_{C \in \pi} \sum_{i \in C} \frac{\sum_{j \in C \setminus \{i\}} w(i,j)}{|C|} \\ &= \sum_{C \in \pi} \frac{1}{|C|} \sum_{i \in C} \sum_{j \in C \setminus \{i\}} w(i,j) = \sum_{C \in \pi} \frac{1}{|C|} \sum_{(i,j) \in C \times C \wedge i \neq j} w(i,j)\end{aligned}$$

Formally, the corresponding problem for which we investigate algorithms is the following.

SocialWelfareFHG	
Input:	An FHG G
Solution:	A partition π of G with maximum social welfare

For SocialWelfareFHG, we can assume instances to be symmetric w.l.o.g. thanks to a simple modification of the edge weights.

Remark 1. *Given an FHG as $G = (N, w)$, we define a symmetrized (undirected) version $G' = (N, w')$ in the following manner where $w'(i, j) = w'(j, i)$ is a shorthand notation for $w'(\{i, j\})$.*

$$\forall \{i, j\} \in \binom{N}{2} : w'(i, j) := \frac{w(i, j) + w(j, i)}{2}$$

Then every (well-defined) partition π yields the same social welfare on G and G'.

Proof. Consider an arbitrary partition π of G.

$$\mathcal{SW}_G(\pi) = \sum_{C\in\pi} \frac{1}{|C|} \sum_{(i,j)\in C\times C\wedge i\neq j} w(i,j) = \sum_{C\in\pi} \frac{1}{|C|} \sum_{\{i,j\}\in\binom{C}{2}} w(i,j) + w(j,i)$$
$$= \sum_{C\in\pi} \frac{1}{|C|} \sum_{\{i,j\}\in\binom{C}{2}} 2w'(i,j) = \sum_{C\in\pi} \frac{1}{|C|} \sum_{i\in C} \sum_{j\in C\setminus\{i\}} w'(i,j) = \mathcal{SW}_{G'}(\pi)$$

□

Notice, though, that an issue arises if there are restrictions imposed on the utilities of the game, e.g., as in simple FHGs. This symmetrization turns simple FHGs into non-simple ones if they were not symmetric before. Algorithms for welfare maximization on simple FHGs cannot, without further ado, assume the input to be a symmetric instance.

SocialWelfareSimpleFHG
Input: A simple FHG G
Solution: A partition π of G with maximum social welfare

Accordingly, we need to define a separate problem for the symmetric case there.

SocialWelfareSymmetricSimpleFHG
Input: A symmetric simple FHG G
Solution: A partition π of G with maximum social welfare

Another restriction we consider is one not on the input but on the output of the algorithm.

Definition 4. *A partition π is called a* matching *if and only if it contains only coalitions of size at most two.*

$$\forall C \in \pi : |C| \leq 2$$

We see that computing a MWM on graphs is a special case of welfare maximization in FHGs.

MWM
Input: An FHG G
Solution: A matching μ of G with maximum social welfare

The same applies to MCMs for simple FHGs.

MCM
Input: A simple FHG G
Solution: A matching μ of G with maximum social welfare

We use the terms MWM and MCM to refer both to the problems and their solutions.

Definition 5. *A matching π^* is a MWM on G if and only if its social welfare is optimal among the matchings on G.*

$$(\forall C \in \pi^* : |C| \leq 2) \land (\forall \pi : (\forall C \in \pi : |C| \leq 2) \implies \mathcal{SW}_G(\pi) \leq \mathcal{SW}_G(\pi^*))$$

This renders the next definition very simple.

Definition 6. *A MCM is a MWM on a simple FHG.*

Note that our definition of matchings differs slightly from the standard one. Since we only consider complete graphs (as the descriptions of FHGs), every pair of vertices shares an edge, may it be of zero or even negative weight. Accordingly, every pair of vertices can be matched. However, due to the fact that we are exclusively interested in social welfare, this has no impact.

Let us conclude the definitions with the competitiveness of online algorithms. Since we take into account randomized algorithms, the competitive ratio is defined via the expected performance of an algorithm. The randomness of the expectation here does not only include inherently random behavior of the algorithm but also possible randomness of the problem instance, e.g., in the random arrival setting (Chapter 5).

Definition 7. *We say that an online algorithm ALG achieves a* competitive ratio *of $c \in [0, 1]$ (it is c-competitive) on a set of FHG instances $\mathcal{G}$ if and only if the ratio of the expected social welfare of the algorithm's output and the social welfare of the optimal partition π^* is at least c for all instances in $\mathcal{G}$.*

$$\forall G \in \mathcal{G} : c \leq \frac{\mathbb{E}\left[\mathcal{SW}_G\left(ALG(G)\right)\right]}{\mathcal{SW}_G(\pi^*)}$$

ALG achieves a competitive ratio of c if and only if it does so on the entire domain. If $\mathcal{SW}_G(\pi^*) = 0$, we define the competitive ratio of any algorithm to be 1 *on* $\{G\}$.

A Word on Terminology In our proofs, it is often simpler to only speak about the graph representation of an FHG rather than about the utilities or preferences of agents, especially because we focus mainly on symmetric FHGs. Therefore, we will regularly refer to agents as vertices, and argue about edges instead of utilities.

An example of a symmetric FHG G is given in Figure 2.1. It consists of 6 agents—represented by labeled vertices—with the respective utilities depicted by the labeled edges among them. Pairs of agents between which no edge is drawn have a mutual utility of 0. The colored areas show a possible partition π on G. The two coalitions in it

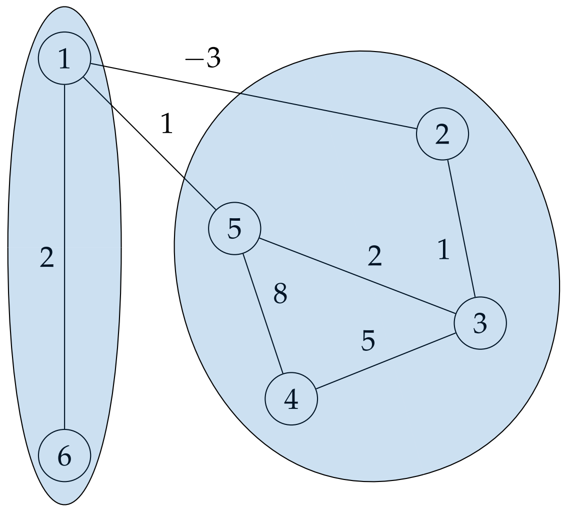

Figure 2.1: A symmetric FHG with 6 agents. Edges not drawn have weight 0. The colored areas represent a partition consisting of two coalitions.

are $C_1 = \{1, 6\}$ and $C_2 = \{2, 3, 4, 5\}$. The social welfare of the partition is

$$\begin{aligned}
\mathcal{SW}_G(\pi) &= \sum_{C \in \pi} \frac{1}{|C|} \sum_{(i,j) \in C \times C \wedge i \neq j} w(i,j) \\
&= \left(\frac{1}{|C_1|} \sum_{(i,j) \in C_1 \times C_1 \wedge i \neq j} w(i,j) \right) + \left(\frac{1}{|C_2|} \sum_{(i,j) \in C_2 \times C_2 \wedge i \neq j} w(i,j) \right) \\
&= \left(\frac{2}{|C_1|} \sum_{\{i,j\} \in \binom{C_1}{2}} w(i,j) \right) + \left(\frac{2}{|C_2|} \sum_{\{i,j\} \in \binom{C_2}{2}} w(i,j) \right) \\
&= \left(\frac{2}{|C_1|} w(1,6) \right) \\
&\quad + \left(\frac{2}{|C_2|} \left(w(2,3) + w(2,4) + w(2,5) + w(3,4) + w(3,5) + w(4,5) \right) \right) \\
&= \left(\frac{2}{2} \cdot 2 \right) + \left(\frac{2}{4} \left(1 + 0 + 0 + 5 + 2 + 8 \right) \right) \\
&= 2 + 8 = 10.
\end{aligned}$$

3 Offline Setting

In this chapter we study the offline setting where all agents and utilities are known to the algorithm. Section 3.1 is about the computational hardness of the problem. In Section 3.2, we are interested in guarantees on the worst-case performance of algorithms with polynomial running time.

3.1 Hardness

It is known that SocialWelfareSymmetricSimpleFHG is NP-hard (Aziz, Gaspers, Gudmundsson, et al., 2015). Of course, this implies NP-hardness of SocialWelfareFHG. But for that problem, we can show a stronger hardness result using a reduction from MinimumCliqueCover, which is well-known to be NP-hard (Karp, 1972).

MinimumCliqueCover	
Input:	An undirected unweighted graph G
Solution:	A partition of G into the minimal possible number of cliques

Here, we mean an unweighted graph in the classical sense, with unweighted edges and missing edges. Notice that it does not matter whether we require a partition of the graph into cliques or only a set of cliques that covers all vertices.

Theorem 1. SocialWelfareFHG *does not admit an FPTAS unless P=NP.*

Proof. Given an instance G of MinimumCliqueCover with n vertices, turn it into a complete weighted graph G' such that all vertex pairs, which share an edge in G, share an edge of weight 1 in G', and all vertex pairs, which do not share an edge in G, share an edge of weight $-n$ in G'.

We interpret G' as an FHG. Note that any reasonable solution to this FHG will only form coalitions which are cliques in the original MinimumCliqueCover instance because it would contain edges of highly negative weight otherwise. This allows for an easy method of computing the social welfare. Let π be a partition of G' which consists of m cliques (in G).

$$\mathcal{SW}_{G'}(\pi) = \sum_{C\in\pi} \frac{2}{|C|} \sum_{\{i,j\}\in\binom{C}{2}} 1 = \sum_{C\in\pi} \frac{2}{|C|} \cdot \frac{|C|(|C|-1)}{2} = \sum_{C\in\pi} (|C|-1) = n - m$$

Assume now, we have access to an oracle which, for any $\epsilon > 0$, returns a $(1-\epsilon)$-approximate solution to SocialWelfareFHG and whose running time is polynomial in the input size and in $\frac{1}{\epsilon}$. We call this oracle with $\epsilon = \frac{1}{n}$. Let π denote the partition returned by the oracle and π^* an optimal partition of G'.

$$\frac{\mathcal{SW}(\pi)}{\mathcal{SW}(\pi^*)} \geq 1 - \frac{1}{n}$$

Let m be the number of coalitions (cliques) in π and m^* the number of cliques in π^*. Note that π^* solves MinimumCliqueCover optimally on G. We get

$$\begin{aligned}
\frac{n-m}{n-m^*} &\geq 1 - \frac{1}{n} \\
n - m &\geq \left(1 - \frac{1}{n}\right)(n - m^*) \\
n - m &\geq n - m^* - 1 + \frac{m^*}{n} \\
-m &\geq -m^* - 1 + \frac{m^*}{n} \\
m &\leq m^* + 1 - \frac{m^*}{n} < m^* + 1.
\end{aligned}$$

And clearly

$$\begin{aligned}
\mathcal{SW}(\pi) &\leq \mathcal{SW}(\pi^*) \\
n - m &\leq n - m^* \\
n + m^* &\leq n + m \\
m^* &\leq m.
\end{aligned}$$

From $m < m^* + 1$ and $m^* \leq m$ we follow $m = m^*$. Thus, π would return the minimum number of cliques on G and solve MinimumCliqueCover in polynomial time. □

On a side note, observe that the constructed FHG instances only contain edges of weight 1 and $-n$ which allows them to be interpreted as a variant of aversion to enemies-games. The hardness result applies already to this restricted class of FHGs.

In the proof, we applied a relatively straightforward reduction from MinimumClique-Cover to SocialWelfareFHG. A slightly adapted version of this reduction can be applied from related problems too, like, e.g., ChromaticNumber or even ColorSaving. ChromaticNumber is essentially the same problem as MinimumCliqueCover but on the complement graph, i.e., the objective is not to cover all graph vertices with cliques but with (a minimum number of) independent sets (called "colors"). ColorSaving has the same input and objective but a different measure of quality of the solution. Instead of counting how many colors (independent sets) we need for the graph, we pretend to have $n = |V|$ colors at hand and count how many of them we do not need to use in our solution. We want to show a close connection of SocialWelfareFHG and ColorSaving approximability. Given an instance of ColorSaving, we first compute the complement graph (which can clearly be done in polynomial time) and then change the objective to using a minimum number of cliques to cover it. Now, we apply the above reduction to SocialWelfareFHG. Let us suppose the SocialWelfareFHG oracle returns a solution with social welfare s. It consists of $n - s$ cliques in the graph. Taking the complement graph again, we return to the original graph and the $n - s$ cliques become $n - s$ independent sets, i.e., colors with which we can cover the graph. We have saved $n - (n - s) = s$ colors. The achieved social welfare equals the number of saved colors. Of course, this also holds if the oracle returns an optimal solution. As a result, we get the following.

Corollary 1. *If there is a polynomial-time algorithm achieving an approximation ratio of c for* SocialWelfareFHG*, there is also a polynomial-time algorithm achieving an approximation ratio of c for* ColorSaving.

The best known approximation ratio for ColorSaving, however, currently is $\frac{193}{240}$ (Athanassopoulos, Caragiannis, Kaklamanis, & Kyropoulou, 2009) which is better than $\frac{1}{2}$, the best known for SocialWelfareFHG.

3.2 Approximation

With the hardness results in mind, we want to see what we can actually achieve in polynomial time. It has been discovered that matchings are a possible way of obtaining worst-case guarantees for SocialWelfareFHG.

Theorem 2 (Flammini, Kodric, Monaco, and Zhang, 2021)**.** *Every MWM is a $\frac{1}{2}$-approximation of* SocialWelfareFHG.[1] *This bound is tight for MWMs.*

Since a MWM can be computed in polynomial time (Gabow & Tarjan, 1991), this is already quite encouraging and will also be helpful for online settings. For the offline setting, the question arises whether there might be a simple way to further improve the social welfare from the partition computed by a MWM. As such "simple improvements" we consider beneficial coalition merges and Pareto-improvements. Note, however, that computing Pareto-improvements for (symmetric) FHGs is in general NP-hard (Bullinger, 2020). We will see that neither of these two approaches results in a better approximation ratio because there exist worst-case instances for which no such improvement is possible.

First, we look at beneficial coalition merges (we also say "extending" the matching). A beneficial coalition merge is one which increases the social welfare of the partition. Algorithm 1 is non-deterministic on purpose to demonstrate that for the approximation ratio it is irrelevant with which (of possibly multiple) MWM it starts and in which order the coalitions are merged. Its running time is clearly polynomial for any myopic choice of the coalitions to be merged, e.g., according to some ranking.

Proposition 1. *Algorithm 1 is only a $\frac{1}{2}$-approximation for* SocialWelfareFHG.

[1] Flammini, Kodric, Monaco, and Zhang showed the theorem for symmetric FHGs and did not mention that, because of the possible symmetrization, it holds for general FHGs.

Algorithm 1 Extended MWM

Input: An FHG G

Output: A partition on G

Let μ be a MWM on G
Set $\pi := \mu$
while Beneficial merge in π is possible **do**
  Let C_1 and C_2 be two coalitions in π which can be beneficially merged
  Set $\pi := \pi \cup \{C_1 \cup C_2\} \setminus \{C_1, C_2\}$ ▷ *Merge the coalitions*
return π

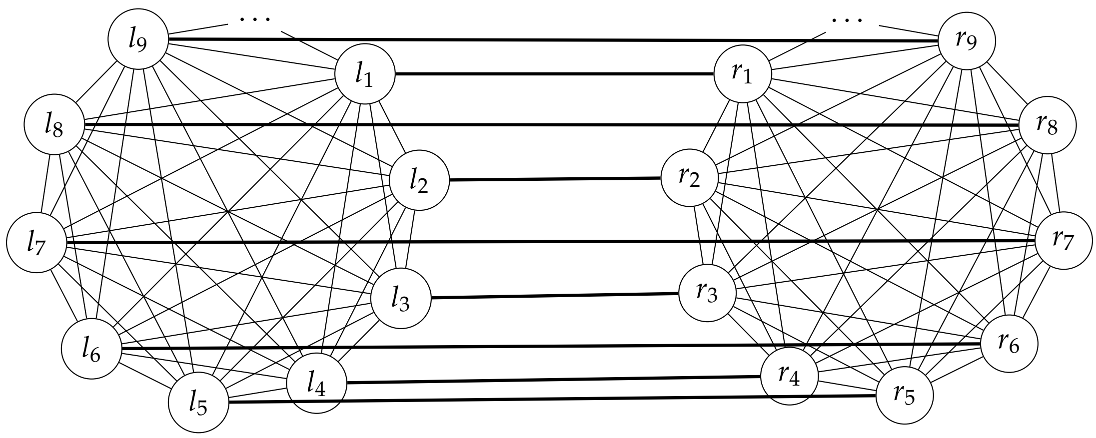

Figure 3.1: Instances where extending a MWM does not improve the approximation ratio. It is $n = 2k$, and the vertices are l_i and r_i, where $i \in [k]$. Normal edges have weight 1, thick edges have weight $1 + \epsilon$ (where $\epsilon > 0$ is very small), and all edges not drawn have weight $-n$. The MWM contains all thick edges as coalitions and cannot be extended. Its social welfare is $k \cdot (\epsilon + 1)$. The optimal partition puts the l_i vertices in one coalition and the r_i vertices in the other. Its social welfare is $2k - 2$.

Proof. Theorem 2 implies that the approximation ratio is at least $\frac{1}{2}$ because Algorithm 1 can never perform worse than a MWM. To see that it is at most $\frac{1}{2}$ too, consider the family of instances depicted in Figure 3.1. There are $|V| = n = 2k$ agents who form two disjoint cliques, each of size k, respectively. We have $V = L \uplus R$ where $|L| = |R| = k$, $L = \{l_1, l_2, \dots, l_k\}$, and $R = \{r_1, r_2, \dots, r_k\}$. For all $i, j \in [k]$ with $i \neq j$, we have $w(l_i, l_j) = w(r_i, r_j) = 1$, $w(l_i, r_j) = -n$, and $w(l_i, r_i) = 1 + \epsilon$. Consider the parameter $\epsilon > 0$ to be very small.

Algorithm 1 starts with a MWM which is in this case $\mu = \{\{l_i, r_i\} \mid i \in [k]\}$. As no beneficial merge is possible, it directly returns that partition and stops. The achieved social welfare is $k\frac{2(1+\epsilon)}{2} = k + k\epsilon$.

The optimal solution, though, is to form two coalitions consisting of the large cliques, i.e., $\pi^* = \{L, R\}$. It has social welfare $2\frac{2\binom{k}{2}}{k} = 2k - 2$. Now, for $\epsilon := \frac{1}{k}$ and $k \to \infty$ the approximation ratio approaches $\frac{1}{2}$.

$$\lim_{k\to\infty} \frac{\mathcal{SW}(\mu)}{\mathcal{SW}(\pi^*)} = \lim_{k\to\infty} \frac{k + k\epsilon}{2k-2} = \lim_{k\to\infty} \frac{k+1}{2k-2} = \frac{1}{2}$$

□

Next, we show an even stronger statement (regarding the upper bound of the approximation ratio). However, the proof of Proposition 1 can still be considered interesting as the used instances are not star-like (as opposed to most other proofs).

Proposition 2. *Algorithm 1 is only a $\frac{1}{2}$-approximation for* SocialWelfareFHG *instances where its returned partition is Pareto-optimal.*

Proof. Consider the family of instances depicted in Figure 3.2 (for $k = 9$). The graphs are spiders with a central vertex a, a set of inner vertices I, and outer vertices O. There are $|V| = n = 2k + 1$ vertices in total. We have $V = \{a\} \uplus I \uplus O$ where $|I| = |O| = k$, $I = \{i_1, i_2, \ldots, i_k\}$, and $O = \{o_1, o_2, \ldots, o_k\}$. For all $j, l \in [k]$ with $j \neq l$, we have $w(a, i_j) = 1$, $w(i_j, i_l) = 0$, $w(i_j, o_j) = \epsilon$, and $w(a, o_j) = w(i_j, o_l) = w(o_i, o_j) = -n$. Again, take the parameter $\epsilon > 0$ to be very small.

Algorithm 1 starts with a MWM which is w.l.o.g. $\mu = \{a, i_1\} \cup \{\{i_j, o_j\} \mid j \in [k] \setminus \{1\}\} \cup \{o_1\}$. As no beneficial merge is possible, it directly returns that partition and stops. The returned partition is Pareto-optimal, meaning there exists no Pareto-improvement. The achieved social welfare is $2 \cdot \frac{1}{2} + (k-1) \cdot 2 \cdot \epsilon = 1 + 2k\epsilon - 2\epsilon$.

The optimal solution is to form one coalition consisting of the central vertex and the inner vertices and put all outer vertices in singleton coalitions, i.e., $\pi^* = \{\{a\} \cup I\} \cup \{\{o_j\} \mid j \in [k]\}\}$. It has social welfare $2\frac{k}{k+1}$. Now, for $\epsilon := \frac{1}{k^2}$ and $k \to \infty$ the approximation ratio approaches $\frac{1}{2}$.

$$\lim_{k\to\infty} \frac{\mathcal{SW}(\mu)}{\mathcal{SW}(\pi^*)} = \lim_{k\to\infty} \frac{1 + 2k\epsilon - 2\epsilon}{2\frac{k}{k+1}} = \lim_{k\to\infty} \frac{1 + 2\frac{1}{k} - 2\frac{1}{k^2}}{2\frac{k}{k+1}} = \frac{1}{2}$$

□

The impression that it might be hard to achieve a competitive ratio better than $\frac{1}{2}$ can be backed by another observation. Even when having a minimum clique cover at hand (which is itself NP-hard to compute), one cannot do better (in simple FHGs), although that can be considered a quite powerful clustering/community-detection solution which is stronger than a MCM.

Proposition 3. *Any minimum clique cover is a $\frac{1}{2}$-approximation for* SocialWelfareSymmetricSimpleFHG*.*

Proof. First, we show that the approximation ratio is at least $\frac{1}{2}$. Assume, the minimum clique cover returns k cliques. Then, the achieved social welfare is $n - k$. Regarding the optimal partition, we can w.l.o.g. assume that it is the grand coalition (a coalition

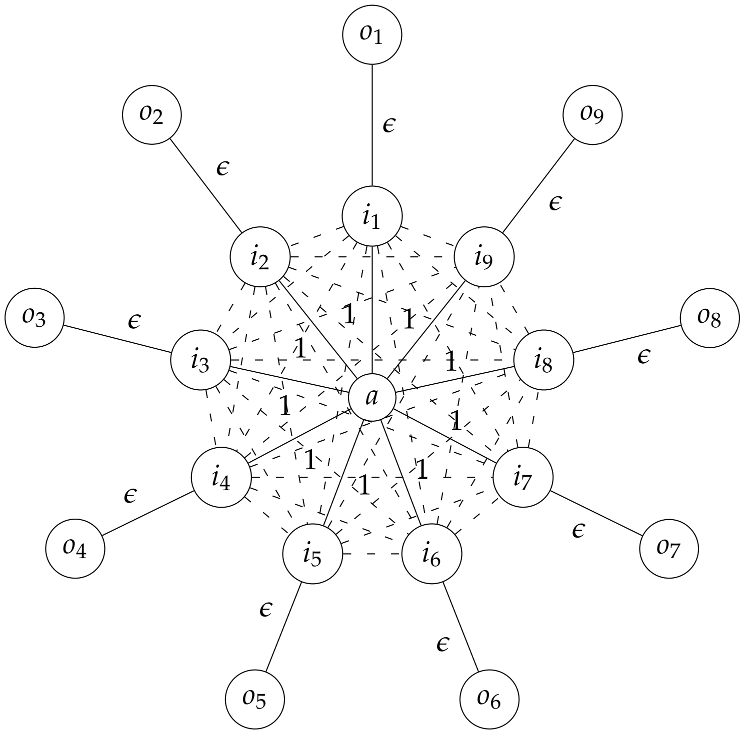

Figure 3.2: Instance (with $k = 9$) of a family where extending a MWM does not improve the approximation ratio while being Pareto-optimal. Dashed edges have weight 0, edges not drawn have weight $-n$, edges from the central vertex a to the inner vertices i_j have weight 1, and edges from inner vertices i_j to outer vertices o_j have weight ϵ. One MWM matches the edge $\{a, i_1\}$ and $\{i_j, o_j\}$ for $j \in [9] \setminus \{1\}$. It has social welfare $1 + 2k\epsilon - 2\epsilon$. The optimal partition is one coalition with a and all i_j while the other vertices are in singleton coalitions. It has social welfare $2\frac{k}{k+1}$.

containing all agents). If the grand coalition is not optimal, consider any optimal partition $\pi^* = \{C_1^*, C_2^*, \dots\}$ and suppose we would not return a global minimum clique cover but the union of minimum clique covers on the subgraphs C_1^*, C_2^*, ..., which obviously yields at most the social welfare of a global minimum clique cover and lets the following analysis work on the subgraphs as if they were FHGs themselves.

The unweighted graph of the game must have at least one edge missing between all pairs of cliques returned by the minimum clique cover. Then, out of $\frac{1}{2} \cdot n \cdot (n-1)$ possible edges, at least $\frac{1}{2} \cdot k \cdot (k-1)$ are missing, making a total of at most $\frac{1}{2} \cdot (n \cdot (n - 1) - k \cdot (k-1))$ edges. We can thus bound the social welfare of the grand coalition by

$$\begin{aligned}
&\frac{2 \cdot \frac{1}{2} \cdot (n \cdot (n-1) - k \cdot (k-1))}{n} \\
=&\frac{n \cdot (n-1) - k \cdot (k-1)}{n} \\
=&\frac{n^2 - n - k^2 + k}{n} \\
\overset{k \leq n}{\leq}&\frac{n^2 - k^2}{n} \\
=&\frac{(n-k)(n+k)}{n} \\
\overset{k \leq n}{\leq}&\frac{(n-k)2n}{n} \\
=&2(n-k).
\end{aligned}$$

Second, we show that the approximation ratio is at most $\frac{1}{2}$. Consider the family of stars, i.e., graphs of the form $|V| := \{a\} \uplus O$ where $O = \{o_1, o_2, \dots, o_{n-1}\}$, with edges $w(a, o_i) = 1$ and $w(o_i, o_j) = 0$ for all $i, j \in [n-1]$ where $i \neq j$. A minimum clique cover coincides with a MCM here. It matches one edge of the star, say $\{a, o_1\}$, and leaves the other vertices as singletons. The social welfare is accordingly 1. The optimal partition is to form the grand coalition, which gives a social welfare of $\frac{2(n-1)}{n}$. For $n \to \infty$, this results in a competitive ratio of $\frac{1}{2}$. □

It is an open problem whether for the upper bound of the proof other families of triangle-free graphs work as well.

4 Adversarial Arrival

Now, after having examined the possibilities and impossibilities of the offline setting, let us turn our attention to the main focus of this thesis, which is on online settings. The first one we consider can be regarded as the classical one in the literature. However, it is a pessimistic model. We study the competitive ratio of algorithms under the respective worst-case arrival order of the agents (which we can imagine to be iteratively designed by an "adversary"). This forbids a constant competitive for SocialWelfareFHG because the utilities are unrestricted (Flammini, Monaco, Moscardelli, et al., 2021). The core of the problem is the following. Suppose, the first two agents arrive, and they have a mutual utility of $\alpha > 0$. If the algorithm does not put them into a common coalition, the adversary can stop the input, and the competitive ratio is unbounded. If the algorithm does put them into one coalition, the adversary can let a third agent arrive which has a utility of β for the first agent and a utility of $-\beta$ for the second, where β is way larger than α. Then, the optimal offline solution would be to put the first and third agent into a coalition and achieve a social welfare of β. The competitive ratio is therefore bounded by $\frac{\alpha}{\beta}$. As a consequence, the best known deterministic algorithm for SocialWelfareFHG under adversarial arrival is to greedily match edges whenever possible (Flammini, Monaco, Moscardelli, et al., 2021). For restricted utilities, e.g., simple FHGs, the situation is different and a constant competitive ratio can be obtained. There is a known upper bound for deterministic algorithms.

Theorem 3 (Flammini, Monaco, Moscardelli, et al., 2021)**.** *For* SocialWelfareSymmetricSimpleFHG *with adversarial arrival, there exists no deterministic algorithm with a competitive ratio of $\frac{1}{4} + \epsilon$ for any constant $\epsilon > 0$.*

We can get a better competitiveness via a randomized matching algorithm.

Theorem 4 (Gamlath, Kapralov, Maggiori, et al., 2019)**.** *For* MCM *with adversarial arrival, there exists a randomized algorithm with a competitive ratio of $\frac{1}{2} + \epsilon$ for some constant $\epsilon > 0$.*

From Theorem 2 and Theorem 4 we get the following result.

Theorem 5. *For* SocialWelfareSymmetricSimpleFHG *with adversarial arrival, there exists a randomized algorithm with a competitive ratio of $\frac{1}{4} + \epsilon$ for some constant $\epsilon > 0$.*

Let us consider simple FHGs which are not (necessarily) symmetric. To the best of our knowledge, these were not studied before with respect to the maximization of social welfare. If we just pretend that the instance we get was symmetric, i.e., every directed edge would have a counterpart in the other direction, and then apply the online MCM algorithm by Gamlath, Kapralov, Maggiori, et al. (2019), our solution of the real instance must be at least half as good as on the assumed symmetric one. This is because the obvious worst case is that every directed edge does not have a counterpart. As a result, we also get a constant competitive ratio there.

Corollary 2. *For* SocialWelfareSimpleFHG *with adversarial arrival, there exists a randomized algorithm with a competitive ratio of* $\frac{1}{8} + \epsilon$ *for some constant* $\epsilon > 0$.

5 Random Arrival

Here, we discuss the random arrival scenario. We assume that, for a fixed FHG, the arrival order of all agents is determined by a permutation which is chosen uniformly at random. This seems like a realistic take in many applications and allows for more positive results. The imaginary adversary can now only determine the utilities but not the arrival order. This makes it harder to trick an algorithm into creating coalitions which turn out as unfortunate in the long run. Again, our approach is to look at matching algorithms and use them for general FHGs. First, we present an existing algorithm for approximating MWMs under online random arrival when the total number of agents to come is known. Then, we study two methods of using it as a basis to create an algorithm for an unknown total number of agents. The first approach, discussed in Section 5.1, is based on repeatedly executing the algorithm for an exponentially growing estimation of n. The second approach, discussed in Section 5.2, leads to better results by directly turning the algorithm into a version that does not depend on knowledge of n. In Section 5.3, we prove an upper bound on the possible competitive ratio.

Algorithm 2 is of crucial importance for our results. It is due to Ezra, Feldman, Gravin, and Tang (2022). The gist of its technique is to compute the new MWM of the known subgraph every time a vertex has arrived. If the partner of the new vertex in that local MWM is yet unmatched in the algorithm's solution, the edge connecting both gets added to the solution. The algorithm has a parameter k, which can be chosen optimally if n is known beforehand, as in the model studied by Ezra, Feldman, Gravin, and Tang (2022). It is important to note that the algorithm assumes to be given a complete graph. There must be no missing edges but only such of 0 weight. By having a complete graph as the input, it follows that in every subgraph with an even number of vertices, there exists a MWM in which the only unmatched vertices share only edges of non-positive weight with other unmatched vertices. Since in the model of Ezra, Feldman, Gravin, and Tang (2022), no edges of negative weight are considered, we have to adapt the algorithm a little bit to make sure that such local MWMs are still complete. However, it is easy to see that their analysis is still correct for our version. For computing the local MWMs, our algorithm treats negative edges as if they had weight 0 (in the pseudocode, this is reflected by the input specification). If such an edge (or one which originally had weight 0) is part of a local MWM (i.e., it is an e_t in line 12), it will not be added to the solution which is output by the algorithm due to the check in line 15. There, we basically check if the edge has positive weight. The reason why the comparison is against an ϵ and not 0 is given by a measure to ensure the satisfaction of another requirement of the algorithm, namely, that the MWM is always unique. We achieve this (with probability 1) by choosing a sufficiently small value $\epsilon > 0$ and perturbing the weight of every edge by a random value in $[0, \epsilon]$. "Sufficiently small" here means two things. First, with our check in line 15, we must still be able to distinguish between edges which originally had a positive weight and such edges which only became positive after the perturbation. Second, the unique MWMs of the subgraphs must always also be a MWM on the respective unperturbed subgraph.

Algorithm 2 Online MWM for random arrival

Input: A tuple (G, ϵ), where

- $G = (V, w)$ is an undirected complete weighted graph with all edge weights non-negative and pairwise different (its vertices arrive in random order)
- ϵ is the perturbation threshold

Output: A matching on G

1: Let $v_1, \dots, v_n$ be the vertices in arrival order
2: $A := V$, $\mu := \emptyset$ ▷ *A is the set of available vertices, μ is the returned matching*
3: **for** $t = k + 1$ to n **do**
4: Let $V_t := \{v_1, \dots, v_t\}$ ▷ *V_t is the set of vertices arrived up to time t*
5: **if** t is odd **then**
6: Select $r_t \in \{1, \dots, t-1\}$ uniformly at random
7: Set $V'_t := V_t \setminus \{v_{r_t}\}$ ▷ *delete a random vertex from $v_1, \dots, v_{t-1}$*
8: **else**
9: Set $V'_t := V_t$
10: Let μ_t be the MWM in $G[V'_t]$
11: Let p_t be the partner of v_t in μ_t
12: Set $e_t := \{v_t, p_t\}$
13: **if** $p_t \in V_t \cap A$ **then**
14: Remove v_t and p_t from A
15: **if** $w(e_t) > \epsilon$ **then**
16: Add e_t to μ ▷ *add the chosen edge to the matching*
17: **return** matching μ

Theorem 6 (Ezra, Feldman, Gravin, and Tang, 2022)**.** *For* MWM *with random arrival, Algorithm 2 achieves a competitive ratio of* $\frac{1}{3} + \frac{k^2}{n^2} - \frac{4k^3}{3n^3} - \mathcal{O}(\frac{1}{n})$.

All vertices arriving up to time k fall into a sampling phase. Here, the algorithm refrains from matching any vertices and keeps them available for later use. If we set $k = \lfloor \frac{n}{2} \rfloor$, we get the best performance for the algorithm. This makes the algorithm depend on knowledge of the total number of vertices.

Corollary 3 (Ezra, Feldman, Gravin, and Tang, 2022)**.** *For* MWM *with random arrival and known n, there exists an algorithm with a competitive ratio of* $\frac{5}{12} - \mathcal{O}(\frac{1}{n})$.

By Theorem 2 and Corollary 3, we obtain the following result.

Lemma 1. *For* SocialWelfareFHG *with random arrival and known n, Algorithm 2 achieves a competitive ratio of* $\frac{5}{24} - \mathcal{O}(\frac{1}{n})$.

5.1 Iterated Doubling Approach

We can get rid of the dependency of a known n while still keeping a constant competitive ratio. Our first approach sticks to the idea of a sampling phase.

Definition 8 (Bullinger and Romen, 2023)**.** *Let ALG be any online coalition formation algorithm for known and even n. The* iterated doubling variant *of ALG (I-ALG) proceeds as follows: It maintains a parameter i that is set to $i = 0$ in the beginning and increased by 1 whenever the next 2^{i+1} agents have arrived. We refer to the time during which the counter is set to a certain value j as the jth* phase*. In the jth phase, I-ALG applies ALG to the agents arriving in the jth phase, assuming that 2^{j+1} agents arrive.*

Originally, the iterated doubling variant of an algorithm has been defined for the use case of ASHGs. We make use of the fact that the technique also works for FHGs. Moreover, we apply a more fine-grained analysis here, which uses Bullinger and Romen's analysis as a basis.

Lemma 2. *Let ALG be an online algorithm for* SocialWelfareFHG *with random arrival and known even n, which has a competitive ratio of c. Then, I-ALG has a competitive ratio of* $\frac{c}{12} - \mathcal{O}(\frac{1}{n})$ *for* SocialWelfareFHG *with random arrival (and unknown n).*

Proof. Assume that *ALG* is a c-competitive algorithm for SocialWelfareFHG under random arrival with known and even n. Let us consider *I-ALG*. In the jth phase, if it gets completed, 2^{j+1} agents arrive. Therefore, at the time it gets completed, exactly $2^{j+2} - 2$ agents have arrived in total, as $\sum_{i=0}^{j} 2^{i+1} = \sum_{j=1}^{j+1} 2^i = (\sum_{j=0}^{j+1} 2^i) - 1 = 2^{j+2} - 2$.

Let i^* denote the largest index such that *I-ALG* completes phase i^*. We know $n \geq 2^{i^*+2} - 2$ and $n < 2^{i^*+3} - 2$.

$$\begin{array}{rcccl} & 2^{i^*+2} - 2 & \leq & n & < & 2^{i^*+3} - 2 \\ \implies & 2^{i^*+2} & \leq & n+2 & < & 2^{i^*+3} \\ \implies & i^* + 2 & \leq & \log_2(n+2) & < & i^* + 3 \\ \implies & i^* & \leq & \log_2(n+2) - 2 & < & i^* + 1 \\ \implies & i^* & = & \lfloor \log_2(n+2) - 2 \rfloor & & \\ & & = & \lfloor \log_2(n+2) \rfloor - 2 & & \end{array} \tag{5.1}$$

Since in phase j, 2^{j+1} agents arrive and the total number of agents n is less than $2^{i^*+3} - 2$, the fraction of agents arriving in that phase must be

$$\frac{2^{j+1}}{n} > \frac{2^{j+1}}{2^{i^*+3} - 2} > 2^{j-i^*-2} = \frac{1}{2^{i^*-j+2}}. \tag{5.2}$$

In the following, we define $x := i^* - j + 2$ for better readability.

Let $J_j \subseteq N$ with $j \leq i^*$ be the random subset of agents in the jth (completed) iteration. Then, it holds that

$$\begin{aligned}\mathbb{E}_{\sigma\sim U(\Sigma(N))}\left[\mathcal{SW}(I\text{-}ALG(G,\sigma))\right] &\geq \sum_{j=0}^{i^*}\mathbb{E}_{J_j}\left[\mathbb{E}_{\sigma\sim U(\Sigma(J_j))}\left[\mathcal{SW}(ALG(G[J_j],\sigma))\right]\right]\\ &\geq \sum_{j=0}^{i^*}\mathbb{E}_{J_j}\left[c\cdot\mathcal{SW}(\pi^*(G[J_j]))\right]\\ &= c\cdot\sum_{j=0}^{i^*}\mathbb{E}_{J_j}\left[\mathcal{SW}(\pi^*(G[J_j]))\right]. \end{aligned}\tag{5.3}$$

Let π^* be a partition for G achieving maximum welfare. Moreover, define $E^* := \left\{\{u,v\} \in \binom{N}{2} \mid u \in \pi^*(v)\right\}$, i.e., the pairs of agents that are in a joint coalition in π^*. Note that for every set $\{u,v\} \in E^*$ it holds that

$$\begin{aligned}\Pr(\{u,v\}\subseteq J_j) &\overset{(5.2)}{>} \frac{n}{2^x\cdot n}\cdot\frac{\frac{n}{2^x}-1}{n-1} = \frac{1}{2^x}\cdot\frac{\frac{n-2^x}{2^x}}{n-1} = \frac{1}{4^x}\cdot\frac{n-2^x}{n-1}\\ &> \frac{1}{4^x}\cdot\frac{n-2^x}{n} = \frac{1}{4^x}\left(1-\frac{2^x}{n}\right) = \frac{1}{4^x}-\frac{1}{2^x\cdot n}.\end{aligned}\tag{5.4}$$

The first inequality holds because we have n different possible positions for u, from which more than $\frac{n}{2^x}$ are in J_j, and after that $n-1$ different possible positions for v, from which more than $\frac{n}{2^x}-1$ are left in J_j. We want to use the sum of those probabilities for all $j \leq i^*$.

$$\sum_{j=0}^{i^*}\Pr(\{u,v\}\subseteq J_j) \overset{(5.4)}{>} \sum_{j=0}^{i^*}\left(\frac{1}{4^x}-\frac{1}{2^x\cdot n}\right) = \underbrace{\sum_{j=0}^{i^*}\frac{1}{4^x}}_{=:A} - \underbrace{\sum_{j=0}^{i^*}\frac{1}{2^x\cdot n}}_{=:B}$$

We analyze the two parts of the sum, A and B, separately. Let $m := 2^{i^*+2}$ and observe that $\frac{n+2}{2} < m \leq n+2$ considering Equation (5.1). For A we get

$$\begin{aligned}A &= \sum_{j=0}^{i^*}\frac{1}{4^{i^*-j+2}} = \sum_{j=0}^{i^*}\left(\frac{1}{2^{i^*-j+2}}\right)^2 = \sum_{j=0}^{i^*}\left(\frac{2^j}{m}\right)^2 = \left(\frac{1}{m}\right)^2\sum_{j=0}^{i^*}4^j\\ &= \left(\frac{1}{m}\right)^2\cdot\frac{1}{3}\left(4^{i^*+1}-1\right) = \frac{1}{3m^2}\left(\frac{1}{4}(2^{i^*+2})^2-1\right)\\ &= \frac{1}{3m^2}\left(\frac{1}{4}m^2-1\right) = \frac{1}{12}-\frac{1}{3m^2}\\ &\geq \frac{1}{12}-\frac{1}{3\left(\frac{n+2}{2}\right)^2} = \frac{1}{12}-\frac{4}{3n^2+12n+12}.\end{aligned}$$

For B we get

$$B = \sum_{j=0}^{i^*} \frac{1}{2^{i^*-j+2} \cdot n} = \sum_{j=0}^{i^*} \frac{2^j}{m \cdot n} = \frac{1}{m \cdot n} \sum_{j=0}^{i^*} 2^j = \frac{1}{m \cdot n} \left(2^{i^*+1} - 1\right)$$
$$= \frac{1}{m \cdot n} \left(\frac{1}{2} \cdot 2^{i^*+2} - 1\right) = \frac{1}{m \cdot n} \left(\frac{1}{2} m - 1\right) < \frac{1}{m \cdot n} \cdot \frac{1}{2} \cdot m = \frac{1}{2n}.$$

So

$$\sum_{j=0}^{i^*} \Pr(\{u, v\} \subseteq J_j) > A - B > \frac{1}{12} - \frac{4}{3n^2 + 12n + 12} - \frac{1}{2n}. \tag{5.5}$$

Recall that, for a given subset $J_j \subseteq N$, $\pi^*[J_j]$ denotes the partition π^* restricted to J_j. It follows

$$\begin{aligned}
\sum_{j=0}^{i^*} \mathbb{E}_{J_j} \left[\mathcal{SW}(\pi^*(G[J_j]))\right] &\geq \sum_{j=0}^{i^*} \mathbb{E}_{J_j} \left[\mathcal{SW}(\pi^*[J_j])\right] \\
&\geq \sum_{j=0}^{i^*} \sum_{\{u,v\} \in E^*} \Pr(\{u, v\} \subseteq J_j) \frac{2w(u, v)}{|\pi^*(u)|} \\
&\overset{(5.5)}{>} \left(\frac{1}{12} - \frac{4}{3n^2 + 12n + 12} - \frac{1}{2n}\right) \mathcal{SW}(\pi^*) \\
&= \left(\frac{1}{12} - \mathcal{O}\left(\frac{1}{n}\right)\right) \mathcal{SW}(\pi^*).
\end{aligned} \tag{5.6}$$

The second inequality here holds because the coalitions of π^* can potentially only become smaller, not larger, when restricted to agents in a J_j. Therefore, the denominator cannot become smaller by leaving out that restriction.

Putting it all together, we get

$$\begin{aligned}
\mathbb{E}_{\sigma \sim U(\Sigma(N))} \left[\mathcal{SW}(I\text{-}ALG(G, \sigma))\right] &\overset{(5.3)}{=} c \cdot \sum_{j=0}^{i^*} \mathbb{E}_{J_j} \left[\mathcal{SW}(\pi^*(G[J_j]))\right] \\
&\overset{(5.6)}{=} c \cdot \left(\frac{1}{12} - \mathcal{O}\left(\frac{1}{n}\right)\right) \mathcal{SW}(\pi^*) \\
&= \left(\frac{c}{12} - \mathcal{O}\left(\frac{1}{n}\right)\right) \mathcal{SW}(\pi^*).
\end{aligned}$$

□

When applying the iterated doubling technique to Algorithm 2, we get the first algorithm with a constant competitiveness.

Theorem 7. *For* SocialWelfareFHG *with random arrival, there exists an algorithm with a competitive ratio of* $\frac{5}{288} - \mathcal{O}(\frac{1}{n})$.

Proof. By Lemma 1, Algorithm 2 with $k = \lfloor \frac{n}{2} \rfloor$ achieves a competitive ratio of $\frac{5}{24} - \mathcal{O}(\frac{1}{n})$ for SocialWelfareFHG with random arrival and known n. Then, its iterated doubling variant, by Lemma 2, has a competitive ratio of $\frac{\frac{5}{24} - \mathcal{O}(\frac{1}{n})}{12} - \mathcal{O}(\frac{1}{n}) = \frac{5}{288} - \mathcal{O}(\frac{1}{n})$. □

5.2 No Sampling Phase

In the previous section we have seen how to achieve a constant competitive ratio via repeated executions of a matching algorithm. It turns out that the approach of making the sampling phases longer by time performs worse than having no sampling phase at all. Moreover, having no sampling phase is a more simple way of getting a constant competitive ratio. Observe that Algorithm 2 only needs to know n in advance if we choose k depending on n. That means if we simply let k be a constant, then, by Theorem 6, we still get a good competitive ratio without having to know the number of agents[1]. Then, for $n \to \infty$, the relative length of the sampling phase (and its impact) vanishes.

Corollary 4. *For* MWM *with random arrival, there exists an algorithm with a competitive ratio of* $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$.

In conjunction with Theorem 2, Corollary 4 directly shows the following.

Theorem 8. *For* SocialWelfareFHG *with random arrival, there exists an algorithm with a competitive ratio of* $\frac{1}{6} - \mathcal{O}(\frac{1}{n})$.

It remains unclear whether this result is tight in the sense that $\frac{1}{6}$ is the best asymptotic competitive ratio one can ascribe to Algorithm 2. Its obvious weakness is that it never forms large coalitions. But in contrast to ASHGs, this is not such a big issue for FHGs. Consider, e.g., an FHG given by a clique of size n with all edges of equal weight. It represents the worst case for the approximation ratio of a MWM with respect to an optimal partition. There are no heavy edges the matching could grab. Instead, the welfare of a partition is solely influenced by the coalition size. Nevertheless, there is a property of such instances that the algorithm can exploit. To see this, first observe that in general graphs one challenge for the algorithm is the fact that, in expectation, it learns about an (random) edge rather late. While the number of known vertices grows linearly (each time a new vertex arrives, we know one more), that of known edges may grow quadratically (when a new vertex arrives at step k, we get to know up to $k-1$ new edges). Hence, when the algorithm matches a relatively heavy edge at an early point in time, there is a high risk of that edge not being relatively heavy globally[2]. For a (complete) graph in which all edges have equal weight this problem does not arise. We will see that, as a result, Algorithm 2 (with constant k) maintains a competitive ratio of $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$ on cliques with uniform edge weights. In fact, this competitive ratio is achieved by the algorithm on the class of symmetric simple FHGs. This is an equivalent statement because if an instance contains edges of weight 0 (instead of the uniform edge weight), that cannot make the algorithm worse off, and scaling the uniform edge weights by any constant factor does not change the algorithm's performance.

Theorem 9. *For* MCM *with random arrival, there exists an algorithm with a competitive ratio of* $\frac{2}{3} - \mathcal{O}(\frac{1}{n})$.

[1]For the analysis of Ezra, Feldman, Gravin, and Tang to work we need $k \geq 3$, though.

[2]Evading this is one of the ideas behind having a sampling phase.

Proof. Ezra, Feldman, Gravin, and Tang (2022) show that the competitive ratio of Algorithm 2 with respect to computing a MWM is given by

$$\sum_{t=k+1}^{n} \left(1 - \frac{2}{3}\left(1 - \frac{(t-4)! \cdot k!}{(t-1)! \cdot (k-3)!}\right)\right) \cdot \mathbb{E}\left[\frac{w(e_t)}{w(\mu^*)}\right],$$

where e_t is the candidate edge to be added in the tth step (see lines 12 and 16 of the algorithm), and μ^* is the (globally optimal) MWM. We apply the algorithm with $k = 3$. As all edges are of the same weight, we know that for every time $t > k$, we have the following inequality, which is asymptotically tight for cliques.

$$\frac{w(e_t)}{w(\mu^*)} \geq \frac{w(e_t)}{\lfloor \frac{n}{2} \rfloor \cdot w(e_t)} \geq \frac{2}{n}$$

We get the following competitive ratio in the MCM domain (recall that a MCM is a MWM on an unweighted graph).

$$\begin{aligned}
&\sum_{t=4}^{n} \left(1 - \frac{2}{3}\left(1 - \frac{(t-4)! \cdot 6}{(t-1)!}\right)\right) \cdot \frac{2}{n} \\
=&\sum_{t=4}^{n} \left(1 - \frac{2}{3}\left(1 - \frac{6}{(t-1)(t-2)(t-3)}\right)\right) \cdot \frac{2}{n} \\
=&\sum_{t=4}^{n} \frac{4}{n \cdot (t-3)} - \frac{8}{n \cdot (t-2)} + \frac{4}{n \cdot (t-1)} + \frac{2}{3n} \\
=&\frac{2}{3} - \frac{2}{n-2} + \frac{4}{n-1} - \frac{2}{n} = \frac{2}{3} - \mathcal{O}\left(\frac{1}{n}\right)
\end{aligned}$$

□

This translates into a competitive ratio of $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$ for SocialWelfareSymmetricSimpleFHG because Theorem 2 is tight here.

Theorem 10. *For* SocialWelfareSymmetricSimpleFHG *with random arrival, there exists an algorithm with a competitive ratio of* $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$.

5.3 Upper Bound

Now that we know a natural way of achieving a decent competitive ratio, let us take a look at which upper bound for the competitive ratio we can determine in this setting. We start by showing the following theorem. After that, we combine the techniques of that proof with an additional idea to prove a better bound.

Theorem 11. *For* SocialWelfareFHG *with random arrival, no algorithm has an asymptotic competitive ratio better than* $\frac{1}{2}$.

To show this, we leverage a family of stars that exhibit only one relatively heavy edge. An algorithm that wants to perform good has to maximize its chances of putting exactly that edge in a coalition (returning a matching with only one matched edge). In many aspects we follow a proof by Ezra, Feldman, Gravin, and Tang (2022, Theorem 4.1).

Proof of Theorem 11. Consider the family $\mathcal{G} := \bigcup_{n\in\mathbb{N}} \mathcal{G}_n$ of FHG instances. For every n, the set $\mathcal{G}_n$ consists of all stars $G_{N'}$ with $|N'| = n-1$. A star $G_{N'} = (N, w)$, where $N := a \uplus N'$, and N' is an arbitrary finite non-empty subset of the natural numbers, has a central vertex a and $n-1$ outer vertices N'. The weights are $w(a,i) := (\alpha n)^i$ for edges between a and any outer vertex $i \in N'$. The factor α is an arbitrary positive rational constant whose sole purpose is to prevent the algorithm from inferring knowledge about n from the observed edge weights. Without it, the algorithm could, for every pair of edge weights it knows, divide one by the other and know that the result must be a power of n. Observe that, since for any such pair of edges the weights differ by a factor of at least αn, there is only one really valuable edge when considering large n. Let $t := \max(N')$. We call the edge between a and t the *top edge*. Let $w(i,j) < -(\alpha n)^{t+1}$ for all edges between vertices $i, j \in N'$ of the outer type[3].

Now, for an online algorithm to have a competitive ratio of c for a fixed instance $G_{N'}$, it must output a partition containing the coalition $\{a, t\}$ (i.e., the top edge) in at least a fraction c of all $n!$ possible arrival orders, in expectation.

Claim 1. *Let ALG be an algorithm for* SocialWelfareFHG *with random arrival. Fix an arbitrary instance $G_{N'} \in \mathcal{G}$. If ALG is c-competitive on $G_{N'}$, then it returns a partition containing only the top edge with probability at least $c - \frac{1}{\alpha n}$. Moreover, in those cases the top edge is the only non-singleton coalition.*

Proof. First, notice that whenever a partition has a coalition containing an outer edge $\{i, j\}$ with $i, j \in N'$, its social welfare will be negative. Therefore, we restrict our attention to algorithms returning at most one coalition of the form $\{a, i\}$, where $i \in N'$, and apart from that only singleton coalitions.

Fix some $G_{N'}$ and let π^* be the corresponding optimal partition. We have

$$\frac{\mathcal{SW}(ALG(G_{N'}))}{\mathcal{SW}(\pi^*)} = \frac{\mathcal{SW}(ALG(G_{N'}))}{(\alpha n)^t} \geq c,$$

[3]The exact weights of those edges are not important as long as they are so highly negative that they shall never be inside coalitions. We avoid defining $w(i,j) := -(\alpha n)^{t+1}$ as that might give an algorithm information about the weight of the top edge.

and so

$$
\begin{aligned}
c \cdot (\alpha n)^t \leq & \mathcal{SW}(ALG(G_{N'})) \\
= & \sum_{i \in N'} \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{i\}\right\} \cup \{a,i\}\right] \cdot w(a,i) \\
= & \sum_{i \in N'} \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{i\}\right\} \cup \{a,i\}\right] \cdot (\alpha n)^i \\
= & \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{t\}\right\} \cup \{a,t\}\right] \cdot (\alpha n)^t \\
& + \sum_{i \in N' \setminus \{t\}} \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{i\}\right\} \cup \{a,i\}\right] \cdot (\alpha n)^i \\
\implies c \leq & \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{t\}\right\} \cup \{a,t\}\right] \\
& + \sum_{i \in N' \setminus \{t\}} \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{i\}\right\} \cup \{a,i\}\right] \cdot \frac{(\alpha n)^i}{(\alpha n)^t} \\
\leq & \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{t\}\right\} \cup \{a,t\}\right] \\
& + \sum_{i \in N' \setminus \{t\}} \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{i\}\right\} \cup \{a,i\}\right] \cdot \frac{(\alpha n)^{t-1}}{(\alpha n)^t} \\
= & \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{t\}\right\} \cup \{a,t\}\right] \\
& + \frac{1}{\alpha n} \sum_{i \in N' \setminus \{t\}} \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{i\}\right\} \cup \{a,i\}\right] \\
\leq & \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{t\}\right\} \cup \{a,t\}\right] + \frac{1}{\alpha n} \\
\implies c - \frac{1}{\alpha n} \leq & \Pr\left[ALG(G_{N'}) = \left\{\{j\} \mid j \in N' \setminus \{t\}\right\} \cup \{a,t\}\right].
\end{aligned}
$$

□

So if an algorithm wants to accomplish a (good) constant competitive ratio, its one and only goal must be to put the top edge in a coalition of size 2 and put all other agents in singleton coalitions. As long as the central agent a has not arrived yet, the algorithm can do nothing else but put agents in singleton coalitions. At the time a arrives, it may form a coalition solely containing the currently heaviest edge—we say it "matches" the "current top edge"—and hope that it is the top edge. If it does so, it has to put all remaining arriving agents in singleton coalitions. If not, it may continue to observe the incoming agents and in each step decide to commit and match the new edge or keep waiting.

Claim 2. *For every algorithm ALG achieving a competitive ratio of c on the instances $\mathcal{G}$, there exists an algorithm ALG' achieving a competitive ratio of at least $c - \frac{1}{\alpha n}$ on $\mathcal{G}$ obeying to the following paradigms.*

At every time step $k \geq 1$, ALG' may only match the current top edge (if possible) or create a new singleton coalition. Its decision among the two alternatives only depends on

- *the time step: k*
- *the agents which have arrived: $\{i \in N \mid \sigma^{-1}(i) \leq k\}$*
- *the order in which the agents have arrived: $\sigma|_{[k]}$*

Proof. Consider an arbitrary algorithm ALG achieving a competitive ratio of c on the instances $\mathcal{G}$. We will modify it to an algorithm ALG' that never matches an edge which is not the current top edge and achieves a competitive ratio of at least $c - \frac{1}{\alpha n}$ on $\mathcal{G}$. If, for any instance $G_{N'} \in \mathcal{G}$ and any arrival order σ, ALG matches an edge that is not the current top edge with probability $p > 0$, ALG' will—instead of matching that not current top edge—not match at all but just keep waiting (by opening up a new singleton coalition). Otherwise, ALG' behaves as ALG. In the case where they behave differently, suppose, ALG' achieves a competitive ratio of c'_σ on $G_{N'}$ with σ given as the arrival order. Now, consider ALG. Since it cannot match the top edge for arrival order σ when matching a *not* current top edge at any time, its competitive ratio c_σ for σ is upper bounded by $(1-p)c'_\sigma + p\frac{1}{\alpha n}$. Then,

$$c'_\sigma \geq (1-p)c'_\sigma \geq c_\sigma - p\frac{1}{\alpha n} \geq c_\sigma - \frac{1}{\alpha n}.$$

As a result, the competitive ratio of ALG' is at least $c - \frac{1}{\alpha n}$ on $\mathcal{G}$. □

Next, we show that we can even restrict our attention to history-independent algorithms, i.e., algorithms not taking into account the order in which the agents have arrived.

Claim 3. *For every algorithm ALG achieving a competitive ratio of c on the instances $\mathcal{G}$, there exists an algorithm ALG'' achieving a competitive ratio of at least $c - \frac{1}{\alpha n}$ on $\mathcal{G}$ obeying to the following paradigms.*

At every time step $k \geq 1$, ALG'' may only match the current top edge (if possible) or create a new singleton coalition. Its decision among the two alternatives only depends on

- *the time step: k*
- *the agents which have arrived: $\{i \in N \mid \sigma^{-1}(i) \leq k\}$*

Proof. We start with an algorithm ALG' given by Claim 2. To such an algorithm ALG' we construct an algorithm ALG'' which is independent of the seen agent arrival order. First, fix an arbitrary time step k and a set N_k of agents having arrived. Then, fix an arrival order $\sigma_k := \sigma|_{[k]}$ up till then, satisfying $\sigma_k([k]) = N_k$. Assume that

- $a \in N_k$,
- ALG' has not matched an edge yet,
- and that either
  - $\sigma_k(k) = a$ or

– the latest arrived edge $\{a, \sigma_k(k)\}$ is the current top edge.

If one of these conditions were not met, ALG' would not make a match in step k. Let ALG''_{σ_k} be an algorithm that ignores the actual arrival order σ_k and instead randomly generates a virtual arrival order σ'_k of N_k, where $\sigma'_k(k) = a$ or the latest edge is the current top edge, conditioned on the probability of ALG' not having matched yet for σ'_k. More specifically, define

$$X(s) := \mathrm{I}\left(s([k]) = N_k \wedge \left(s(k) = a \vee \{a, s(k)\} = \mathrm{argmax}_{e \in \binom{N_k}{2}} w(e)\right)\right)$$
$$S := \sum_{\sigma'_k} \Pr\left[ALG' \text{ would not have matched on } \sigma'_k \text{ yet}\right] \cdot X(\sigma'_k).$$

Then,

$$\Pr\left[ALG''_{\sigma_k} \text{ generates } \sigma'_k\right] := \frac{X(\sigma'_k)}{S} \Pr\left[ALG' \text{ would not have matched on } \sigma'_k \text{ yet}\right].$$

ALG''_{σ_k} continues on the instance, assuming that the arrival order up to time k was σ'_k and therefore is independent of the actually seen arrival order up to that time. Aside from that, ALG''_{σ_k} behaves as ALG'. As σ'_k is drawn from the same distribution as σ_k, ALG''_{σ_k} has, in expectation, the same competitive ratio as ALG'.

We define ALG'' dynamically. It starts like ALG' but at every time k satisfying above assumptions, it starts to imitate the respective ALG''_{σ_k}. An inductive argument shows that the expected competitive ratio of ALG'' is the same as of ALG'. Moreover, ALG'' is always independent of the seen arrival order. □

Claim 3 shows that with respect to constant competitive ratios we can w.l.o.g. consider only algorithms exclusively matching a current top edge and being oblivious to the arrival order. Any such algorithm can be represented by a sequence of functions $f_k : \binom{\mathbb{N}}{k} \to [0,1]$ where $k \in \mathbb{N}$. The function f_k describes the behavior of the algorithm at time $k+1$. It takes as input the set of observed outer vertices (the natural numbers by which they are named) and returns the probability with which the algorithm matches the current top edge if possible.

We go one step further and show that we only get imprecise by an arbitrarily small factor $\epsilon > 0$ if we restrict our attention to algorithms not even taking into account the (absolute) edge weights, meaning every function f_k is a constant p_k with error at most ϵ. In other words, we show that for our instances the described cardinal setting is equivalent to an ordinal setting where an algorithm can compare edge weights only pairwise.

On a side note, the situation for an algorithm here is very similar to a well-known problem where one should pick the larger of two numbers (Cover, 1987). In the problem, there are two slips of paper with numbers written on the back side. After randomly choosing one of them and inspecting the number written on it, one must decide whether to stick with it or discard it and choose the unknown number on the other slip. Naively, one would think that the probability of getting the larger number must be exactly $\frac{1}{2}$.

But one can do better. By randomly generating a "split number" according to any distribution covering all real numbers, the probability of winning can be increased slightly above $\frac{1}{2}$. If the first observed number is lower than the split number, one chooses the other, unknown number. Otherwise, one chooses the number at hand. An algorithm for our star instances faces a decision of the same kind when the central vertex a arrives. Either it matches the heaviest edge now, or it speculates that the (other vertex of the) globally heaviest edge will arrive after a. In that case it must wait and try to get that edge. The important observation is that for the mentioned problem with numbers on slips of paper, an adversary writing the numbers can push ones winning probability arbitrarily close to $\frac{1}{2}$ by reducing the probability that the split number lies between the two written ones. This idea generalizes to our setting.

Lemma 3 is based on the infinite version of Ramsey's theorem (1930). Such applications of Ramsey's theorem have been used in various similar variations of secretary and prophet inequality problems to show worst-case equivalence of the cardinal and ordinal setting (Correa, Dütting, Fischer, & Schewior, 2019; Kaplan, Naori, & Raz, 2020; Moran, Snir, & Manber, 1985).

Lemma 3 (Ezra, Feldman, Gravin, and Tang, 2022)**.** *For any $n \in \mathbb{N}$, any collection of set functions $f_k : \binom{\mathbb{N}}{k} \to [0,1]$, $k \in [n]$, and any $\epsilon > 0$, there exists an infinite set $T \subset \mathbb{N}$ and constants $p_1, \ldots, p_n \in [0,1]$, such that $f_k(N'_k) = p_k + \mathcal{O}(\epsilon)$ for all $N'_k \in \binom{T}{k}$, $k \in [n]$.*

Since the total number of agents that will arrive is finite, for every instance $G_{N'} \in \mathcal{G}_n$, only a prefix of length n of the sequence f_k is relevant. For every $\hat{n} \in \mathbb{N}$, we can apply Lemma 3 with $\epsilon \in \mathcal{O}(\frac{1}{\hat{n}^2})$ to obtain an infinite set T for which all algorithms ignoring the absolute observed edge weights on $G_{N'}$, where $N' \subset T$, could improve their probability of matching the top edge on instances with $n \leq \hat{n}$ by at most $\mathcal{O}(\frac{1}{n})$ if they would incorporate the edge weights into their decisions.

To see this, let $N'_{k-1} := N_k \setminus \{a\}$ and consider any arrival order σ, a time $k \in [n] \setminus \{1\}$, and outer agents $N'_{k-1} \in \binom{T}{k-1}$ where $\sigma^{-1}(a) \leq k \wedge \sigma^{-1}(t) \leq k \wedge \sigma(k) \in \{a, t\}$. Assume, an algorithm ALG achieves a competitive ratio c. Let ALG' be an algorithm described by constant f_ks according to Lemma 3. Then,

$$
\begin{aligned}
c &\leq \Pr\left[\{a,t\} \in ALG(G_{N'},\sigma)\right] \\
&= f_{k-1}(N'_{k-1}) \cdot \prod_{\substack{j \\ \sigma^{-1}(a) \leq j \leq k \\ \forall j' < j : w(a,j') < w(a,j)}} (1 - f_{j-1}(N'_{j-1})) \\
&= \left(p_{k-1} - \mathcal{O}\left(\frac{1}{\hat{n}^2}\right)\right) \cdot \prod_{\substack{j \\ \sigma^{-1}(a) \leq j \leq k \\ \forall j' < j : w(a,j') < w(a,j)}} \left(1 - p_{j-1} + \mathcal{O}\left(\frac{1}{\hat{n}^2}\right)\right) \\
&= \left(p_{k-1} \cdot \prod_{\substack{j \\ \sigma^{-1}(a) \leq j \leq k \\ \forall j' < j : w(a,j') < w(a,j)}} (1 - p_{j-1})\right) \pm \mathcal{O}(n) \cdot \mathcal{O}\left(\frac{1}{\hat{n}^2}\right) \\
&= \left(p_{k-1} \cdot \prod_{\substack{j \\ \sigma^{-1}(a) \leq j \leq k \\ \forall j' < j : w(a,j') < w(a,j)}} (1 - p_{j-1})\right) \pm \mathcal{O}\left(\frac{1}{n}\right) \\
&= \Pr\left[\{a,t\} \in ALG'(G_{N'},\sigma)\right] \pm \mathcal{O}\left(\frac{1}{n}\right).
\end{aligned}
$$

Hence, ALG' has a competitive ratio of at least $c - \mathcal{O}(\frac{1}{n})$. We can strengthen Claim 3 to the following.

Claim 4. *For every algorithm ALG achieving a competitive ratio of c on the instances $\mathcal{G}$, there exists an algorithm ALG′ achieving a competitive ratio of $c - \mathcal{O}(\frac{1}{n})$ on $\mathcal{G}$ such that at every time step $k \geq 1$, ALG′ may only match the current top edge (if possible), and its (possibly randomized) decision whether to do so only depends on the time step k.*

With these simplifying assumptions for algorithms on instances $G_{N'} \in \mathcal{G}$, it is now easy to see that an algorithm has no significant way of telling when it is the right moment to match an edge. The probability that a arrives before t is $\frac{1}{2}$. When an algorithm observes the arrival of a it can either decide to match the current top edge, hoping that it is the top edge, or not match yet, hoping that (t and) the top edge is yet to come. Knowledge of the time step k does not help here because n is unknown to the algorithm. Its decision has to be purely probabilistic. Thus, asymptotically, an algorithm cannot make the right decision more than half of the times in expectation, meaning its competitive ratio is upper bounded by $\frac{1}{2}$. □

Next, we prove a better upper bound for the competitive ratio.

Theorem 12. *For* SocialWelfareFHG *with random arrival, no algorithm has an asymptotic competitive ratio better than $\frac{1}{3}$.*

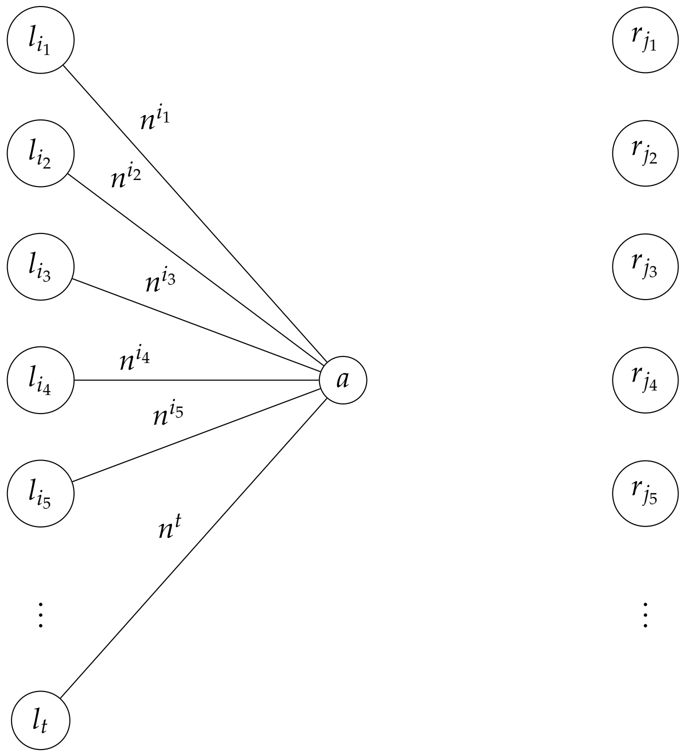

Figure 5.1: The star instances $\mathcal{S}_n$ for which an algorithm has to match the edge $\{a, l_t\}$ to achieve a constant competitive ratio. The ratio between two edge weights is always at least n.

To obtain this bound, we mix the star instances of the proof for Theorem 11 with bistar (double star) instances where the heavy edge connects the two central vertices. The key idea is that we do so in such a way that an algorithm cannot determine quick enough whether the input is a star or a bistar. If it is a star, the algorithm must match the heavy edge of the star, connecting the central vertex with an outer one. If it is a bistar, though, it must wait until both central vertices have arrived and then match the edge between them. Since an algorithm still can only match one edge, it has to guess whether the input is a star or a bistar. And on the stars it still will not be able to perform better than $\frac{1}{2}$.

For a simpler exposition, we will leave out the factor α in the edge weights this time, which would technically also be needed here. It is easy to see that if it were included, all arguments would still work the same.

Proof of Theorem 12. Consider the two families $\mathcal{S} := \bigcup_{n \in \mathbb{N}} \mathcal{S}_n$ and $\mathcal{B} := \bigcup_{n \in \mathbb{N}} \mathcal{B}_n$ of FHG instances. The family $\mathcal{S}_n$ contains stars of size n with one central agent a, a set of outer agents L, and extra agents R which are not connected to any agent (see Figure 5.1). It is $|L| + |R| + 1 = n$. The family $\mathcal{B}_n$ contains bistars of size n with two central agents a and b, and two sets of outer agents L and R of the same size (see Figure 5.2). It is

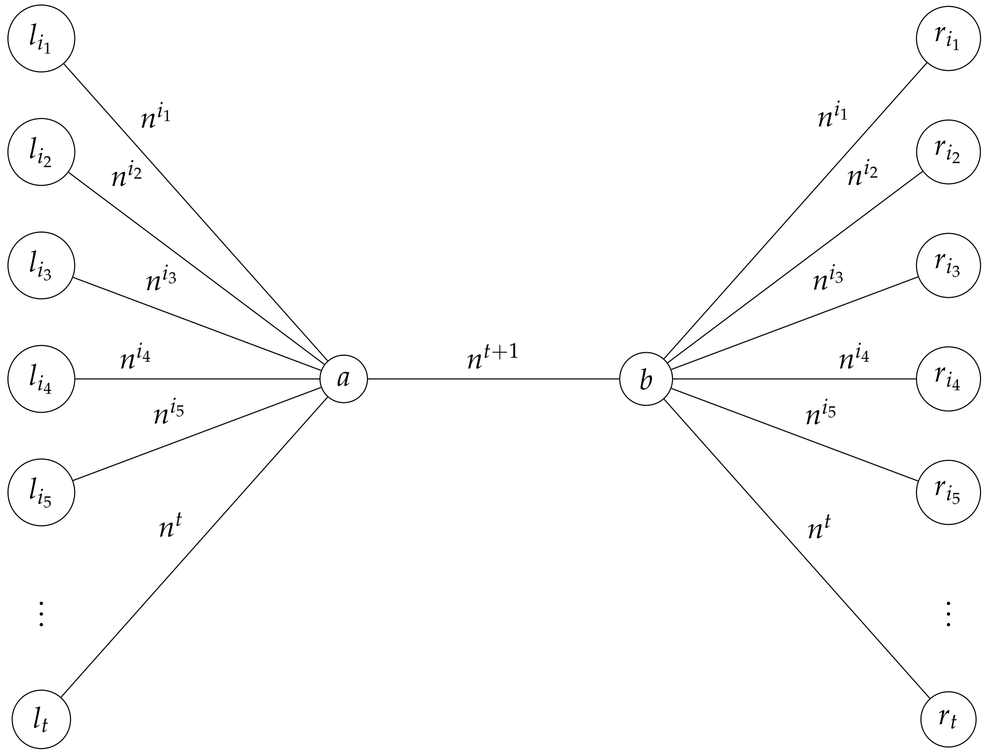

Figure 5.2: The bistar instances $\mathcal{B}_n$ for which an algorithm has to match the edge $\{a, b\}$ to achieve a constant competitive ratio. The ratio between two edge weights is always at least n.

$|L| + |R| + 2 = 2|L| + 2 = n$. By $S_{L,R} \in \mathcal{S}_n$ we denote a star and by $B_{L,R} \in \mathcal{B}_n$ a bistar (with n vertices, respectively). For both stars and bistars, L and R can be represented by finite subsets of the natural numbers. There are $I, J \subset \mathbb{N}$ such that $L := \{l_i \mid i \in I\}$ and $R := \{r_j \mid j \in J\}$. For bistars $\mathcal{B}$, we have $I = J$.

The stars have positive edges only between L and a. The bistars have them between b and R too, and a heavy positive edge between a and b. More specifically, for $S_{L,R} \in \mathcal{S}_n$ and $B_{L,R} \in \mathcal{B}_n$, let $t := \max(I)$. Then, for all $i, i_1, i_2 \in I$ with $i_1 \neq i_2$ and $j, j_1, j_2 \in J$ with $j_1 \neq j_2$, we have $w(l_i, a) = n^i$ and $\max\{w(l_{i_1}, l_{i_2}), w(r_{j_1}, r_{j_2}), w(l_i, r_j), w(a, r_j)\} < -n^{t+2}$, which ensures a highly negative weight of those edges[4]. Moreover, for $B_{L,R} \in \mathcal{B}_n$ we have $w(b, r_j) = n^j$ as well as $w(a, b) = n^{t+1}$. In the instances $\mathcal{B}$ we call the heaviest edge $\{a, b\}$ the "bridge".

As the bistars are symmetric (consider the automorphism mapping a to b and l_i to r_i), we can w.l.o.g. assume that a arrives before b and thus only have to consider half of the arrival orders. An argument very similar to that of Claim 1 shows that an algorithm has to match the bridge with probability at least $c_{\mathcal{B}} - \frac{1}{n}$ in order to achieve a competitive ratio of $c_{\mathcal{B}}$ on $\mathcal{B}$.

[4]See the footnote in the proof of Theorem 11.

We partition the event of matching the bridge according to the different possible arrival times of b. The idea is that, before b arrives, an algorithm cannot distinguish between a star and a bistar. Moreover, the set of all agents arriving before b forms a star instance itself.

Let us now denote by $c_{\mathcal{B}}(n)$ the competitive ratio that an algorithm achieves on $\mathcal{B}_n$.

$$\begin{aligned}
c_{\mathcal{B}}(n) &\leq \frac{1}{n} + \Pr\left[\text{match the bridge on } \mathcal{B}_n\right] \\
&= \frac{1}{n} + \sum_{k=2}^{n} \Pr\left[\text{match the bridge on } \mathcal{B}_n \mid \sigma(k) = b\right] \cdot \Pr\left[\sigma(k) = b\right] \\
&\leq \frac{1}{n} + \sum_{k=2}^{n} \Pr\left[\text{do not match before step } k \text{ on } \mathcal{B}_n \mid \sigma(k) = b\right] \cdot \Pr\left[\sigma(k) = b\right] \\
&= \frac{1}{n} + \sum_{k=2}^{n} \Pr\left[\text{do not match on } \mathcal{S}_{k-1}\right] \cdot \Pr\left[\sigma(k) = b\right] \\
&= \frac{1}{n} + \sum_{k=2}^{n} \Pr\left[\text{do not match on } \mathcal{S}_{k-1}\right] \cdot \frac{2(k-1)}{n(n-1)} \\
&= \frac{1}{n} + \frac{2}{n(n-1)} \sum_{k=1}^{n-1} \Pr\left[\text{do not match on } \mathcal{S}_{k}\right] \cdot k
\end{aligned}$$

Assume for contradiction that an algorithm achieves a competitive ratio of $c \geq \frac{1}{3} + \epsilon$ on $\mathcal{S} \cup \mathcal{B}$, where $\epsilon > 0$ is a constant. Then,

$$\frac{1}{3} + \epsilon \leq \frac{1}{n} + \frac{2}{n(n-1)} \sum_{k=1}^{n-1} \Pr\left[\text{do not match on } \mathcal{S}_{k}\right] \cdot k. \tag{5.7}$$

We want to show that under that restriction, an algorithm cannot maintain a competitive ratio of $\frac{1}{3} + \epsilon$ for stars. Observe that Claims 1, 2, 3, and 4 still hold

- under the restriction of not matching at all with a given probability
- and with "extra" agents sharing no edge of positive weight with any other agent,

so we can apply them to $\mathcal{S}$. Recall that this allows us to describe any algorithm (up to an arbitrarily small error) as a sequence $(p_k)_{k\in\mathbb{N}}$ of probabilities, where p_k represents the probability of matching the current top edge at time step k, given that the algorithm has the option to do so. To find the conflict, we define as $f_{\mathcal{S}}(n)$ the probability of the algorithm not matching at all on instances $\mathcal{S}_n$ and as $c'_{\mathcal{S}}(n)$ its probability of matching the top edge on $\mathcal{S}_n$.

We get

$$\begin{aligned}
f_{\mathcal{S}}(1) =& 1 && (5.8)\\
f_{\mathcal{S}}(n) =& \Pr\left[\text{do not match on } \mathcal{S}_n\right]\\
=& \Pr\left[\text{do not match on } \mathcal{S}_n \mid \sigma(n) = a\right] \cdot \Pr\left[\sigma(n) = a\right]\\
&+ \Pr\left[\text{do not match on } \mathcal{S}_n \mid \sigma(n) \neq a\right] \cdot \Pr\left[\sigma(n) \neq a\right]\\
=&(1 - p_n) \cdot \frac{1}{n} + \Pr\left[\text{do not match on } \mathcal{S}_n \mid \sigma(n) \neq a\right] \cdot \frac{n-1}{n} && (5.9)
\end{aligned}$$

and

$$\begin{aligned}
&\Pr\left[\text{do not match on } \mathcal{S}_n \mid \sigma(n) \neq a\right]\\
=& \Pr\left[\text{do not match on } \mathcal{S}_n \mid \sigma(n) \neq a \wedge \sigma(n) \neq l_t\right] \cdot \Pr\left[\sigma(n) \neq l_t \mid \sigma(n) \neq a\right]\\
&+ \Pr\left[\text{do not match on } \mathcal{S}_n \mid \sigma(n) \neq a \wedge \sigma(n) = l_t\right] \cdot \Pr\left[\sigma(n) = l_t \mid \sigma(n) \neq a\right]\\
=& f_{\mathcal{S}}(n-1) \cdot \frac{n-2}{n-1} + \left(f_{\mathcal{S}}(n-1) \cdot (1 - p_n)\right) \cdot \frac{1}{n-1}, && (5.10)
\end{aligned}$$

so

$$\begin{aligned}
f_{\mathcal{S}}(n) \overset{(5.9)(5.10)}{=}& (1 - p_n) \cdot \frac{1}{n} + \left(f_{\mathcal{S}}(n-1) \cdot \frac{n-2}{n-1} + \left(f_{\mathcal{S}}(n-1) \cdot (1 - p_n)\right) \cdot \frac{1}{n-1}\right) \cdot \frac{n-1}{n}\\
=& \frac{1}{n}\left(1 - p_n + (n - 1 - p_n) \cdot f_{\mathcal{S}}(n-1)\right). && (5.11)
\end{aligned}$$

Furthermore,

$$\begin{aligned}
c'_{\mathcal{S}}(1) =& 1 && (5.12)\\
c'_{\mathcal{S}}(n) =& \Pr\left[\text{match the top edge on } \mathcal{S}_n\right]\\
=& \Pr\left[\text{match the top edge on } \mathcal{S}_n \mid \sigma(n) = a\right] \cdot \Pr\left[\sigma(n) = a\right]\\
&+ \Pr\left[\text{match the top edge on } \mathcal{S}_n \mid \sigma(n) \neq a\right] \cdot \Pr\left[\sigma(n) \neq a\right]\\
=& p_n \cdot \frac{1}{n} + \Pr\left[\text{match the top edge on } \mathcal{S}_n \mid \sigma(n) \neq a\right] \cdot \frac{n-1}{n} && (5.13)
\end{aligned}$$

and

$$\begin{aligned}
&\Pr\left[\text{match the top edge on } \mathcal{S}_n \mid \sigma(n) \neq a\right]\\
=& \Pr\left[\text{match the top edge on } \mathcal{S}_n \mid \sigma(n) \neq a \wedge \sigma(n) \neq l_t\right] \cdot \Pr\left[\sigma(n) \neq l_t \mid \sigma(n) \neq a\right]\\
&+ \Pr\left[\text{match the top edge on } \mathcal{S}_n \mid \sigma(n) \neq a \wedge \sigma(n) = l_t\right] \cdot \Pr\left[\sigma(n) = l_t \mid \sigma(n) \neq a\right]\\
=& c'_{\mathcal{S}}(n-1) \cdot \frac{n-2}{n-1} + \left(f_{\mathcal{S}}(n-1) \cdot p_n\right) \frac{1}{n-1}, && (5.14)
\end{aligned}$$

so

$$\begin{aligned}
c'_{\mathcal{S}}(n) \overset{(5.13)(5.14)}{=}& p_n \cdot \frac{1}{n} + \left(c'_{\mathcal{S}}(n-1) \cdot \frac{n-2}{n-1} + \left(f_{\mathcal{S}}(n-1) \cdot p_n\right) \frac{1}{n-1}\right) \cdot \frac{n-1}{n}\\
=& \frac{1}{n}\left(p_n + p_n \cdot f_{\mathcal{S}}(n-1) + (n-2) \cdot c'_{\mathcal{S}}(n-1)\right). && (5.15)
\end{aligned}$$

Consider $f_{\mathcal{S}}(n) + c'_{\mathcal{S}}(n)$. We will see that p_n does not play a role in this sum.

$$\begin{aligned}
&f_{\mathcal{S}}(n) + c'_{\mathcal{S}}(n) && (5.16)\\
\overset{(5.11)(5.15)}{=}&\frac{1}{n}\left(1 - p_n + (n-1-p_n)\cdot f_{\mathcal{S}}(n-1)\right)\\
&+\frac{1}{n}\left(p_n + p_n \cdot f_{\mathcal{S}}(n-1) + (n-2)\cdot c'_{\mathcal{S}}(n-1)\right)\\
=&\frac{1}{n}\left(1 + (n-1)\cdot f_{\mathcal{S}}(n-1) + (n-2)\cdot c'_{\mathcal{S}}(n-1)\right)\\
=&\frac{1}{n} + f_{\mathcal{S}}(n-1) - \frac{1}{n}\cdot f_{\mathcal{S}}(n-1) + c'_{\mathcal{S}}(n-1) - \frac{2}{n}\cdot c'_{\mathcal{S}}(n-1)\\
=&f_{\mathcal{S}}(n-1) + c'_{\mathcal{S}}(n-1) + \frac{1}{n}\left(1 - f_{\mathcal{S}}(n-1) - 2\cdot c'_{\mathcal{S}}(n-1)\right)\\
\leq&f_{\mathcal{S}}(n-1) + c'_{\mathcal{S}}(n-1) + \frac{1}{n}\left(1 - f_{\mathcal{S}}(n-1) - c'_{\mathcal{S}}(n-1) - \left(\frac{1}{3} + \epsilon - \frac{1}{n}\right)\right) && (5.17)\\
=&f_{\mathcal{S}}(n-1) + c'_{\mathcal{S}}(n-1) + \frac{1}{n}\left(\frac{2}{3} - \epsilon + \frac{1}{n} - f_{\mathcal{S}}(n-1) - c'_{\mathcal{S}}(n-1)\right)
\end{aligned}$$

The inequality (5.17) follows from $c'_{\mathcal{S}}(n) \geq \frac{1}{3} + \epsilon - \frac{1}{n}$ being true for every n. To see why, let $c_{\mathcal{S}}(n)$ denote the competitive ratio of the algorithm on $\mathcal{S}_n$. Then, $\frac{1}{3} + \epsilon \leq c_{\mathcal{S}}(n) \leq c'_{\mathcal{S}}(n) + \frac{1}{n}$. In order to simplify the following calculations, we perform a substitution and let $s(n) := f_{\mathcal{S}}(n) + c'_{\mathcal{S}}(n)$ which gives us a recursively upper bounded sequence.

$$\begin{aligned}
s(n) \leq& s(n-1) + \frac{1}{n}\left(\frac{2}{3} - \epsilon + \frac{1}{n} - s(n-1)\right)\\
=&\frac{1}{n^2} + \frac{2-3\epsilon}{3n} + \frac{n-1}{n}s(n-1)
\end{aligned}$$

The solution to the recurrence is

$$s(n) \leq \frac{3\epsilon + 1 + 3H_n}{3n} + \frac{2}{3} - \epsilon$$

where $H_n := \sum_{k=1}^{n} \frac{1}{k}$ denotes the nth harmonic number. And since $f_{\mathcal{S}}(n) = s(n) - c'_{\mathcal{S}}(n)$ and $c'_{\mathcal{S}}(n) \geq \frac{1}{3} + \epsilon - \frac{1}{n}$, this means

$$f_{\mathcal{S}}(n) \leq \frac{3\epsilon + 1 + 3H_n}{3n} + \frac{2}{3} - \epsilon - \left(\frac{1}{3} + \epsilon - \frac{1}{n}\right) = \frac{3\epsilon + 4 + 3H_n}{3n} + \frac{1}{3} - 2\epsilon. \qquad (5.18)$$

Combining this with our calculations for bistars, we know

$$\begin{aligned}
\frac{1}{3}+\epsilon &\overset{(5.7)}{\leq} \frac{1}{n}+\frac{2}{n(n-1)}\sum_{k=1}^{n-1}\Pr\left[\text{do not match on } \mathcal{S}_k\right]\cdot k \\
&= \frac{1}{n}+\frac{2}{n(n-1)}\sum_{k=1}^{n-1} f_{\mathcal{S}}(k)\cdot k \\
&\overset{(5.18)}{\leq} \frac{1}{n}+\frac{2}{n(n-1)}\sum_{k=1}^{n-1}\left(\frac{3\epsilon+4+3H_k}{3k}+\frac{1}{3}-2\epsilon\right)\cdot k \\
&= \frac{1}{n}+\frac{2}{n(n-1)}\sum_{k=1}^{n-1}\epsilon+\frac{4}{3}+H_k+\left(\frac{1}{3}-2\epsilon\right)\cdot k \\
&= \frac{1}{n}+\frac{n+1}{3(n-1)}-\frac{2n\epsilon}{n-1}+\frac{2H_n}{n-1}+\frac{4\epsilon}{n-1}-\frac{8+6\epsilon}{3(n-1)n}.
\end{aligned}$$

But

$$\lim_{n\to\infty}\left(\frac{1}{n}+\frac{n+1}{3(n-1)}-\frac{2n\epsilon}{n-1}+\frac{2H_n}{n-1}+\frac{4\epsilon}{n-1}-\frac{8+6\epsilon}{3(n-1)n}\right)=\frac{1}{3}-2\epsilon$$

shows that for sufficiently high n we get a contradiction. Consequently, an algorithm cannot have a competitive ratio of $\frac{1}{3}+\epsilon$ on $\mathcal{S}\cup\mathcal{B}$.

For the argumentation leading to the contradiction we assumed the algorithm to have a strict (rather than asymptotic) competitive ratio of $\frac{1}{3}+\epsilon$. Notice however that

- it is a natural assumption. Initially (for $n \leq 1$), any algorithm has a competitive ratio of 1 as well as a probability 1 of not having matched. Since the algorithm does not know n, it makes no sense for it to perform poorly (and push the competitive ratio under $\frac{1}{3}+\epsilon$) up to some point but then eventually reach $\frac{1}{3}+\epsilon$.
- when n_0 is the step from which on the algorithm achieves a strict competitive ratio of $\frac{1}{3}+\epsilon$, we can, in Equations (5.8) and (5.12), consider the constants $f_{\mathcal{S}}(n_0)$ and $c_{\mathcal{S}}(n_0)$ instead. In the following, we would then restrict our attention to steps $n > n_0$ (e.g., for Equations (5.7) and (5.17)) and all arguments work the same.

□

In the proof, we used edges of high negative weight to ensure certain agents do not end up in a coalition together. Moreover, we effectively restricted the possible coalition size to a maximum of 2. That means the bound holds for matching too (the negative edges there even could be replaced by edges of weight 0).

Corollary 5. *For MWM with random arrival, no algorithm has an asymptotic competitive ratio better than* $\frac{1}{3}$.

This upper bound matches the lower bound given by Corollary 4. Finally, we want to mention that the upper bound proven in Theorem 12 applies to ASHGs and MFHGs too and therefore is the new best known bound for online random arrival in those models (to the best of our knowledge, online coalition formation for MFHGs has not been studied yet at all).

6 Free Dissolution

The last setting we study is the same as the one in Chapter 4, except that we give more options to the algorithms. At the time a new agent arrives, an algorithm may decide to dissolve an existing coalition, i.e., all agents of that coalition are then in singleton coalitions. Only after the possible dissolving of a coalition, the new agent has to be assigned to one.

With this modification, it is easy to see that the algorithm proposed by Flammini, Monaco, Moscardelli, et al. (2021) for the adversarial arrival, which creates a maximal matching, is not optimal anymore. Consider the instance in Figure 6.1. The first two agents who arrive share an edge of weight 1. The algorithm will put them into one coalition. The third agent shares an edge of weight w with the first one. Similarly, the fourth and last agent shares an edge of weight w with the second one. The maximal matching algorithm will put both of them into singleton coalitions, yielding a social welfare of 1, whereas the optimal solution would form the coalitions $\{1, 3\}$ and $\{2, 4\}$, resulting in a social welfare of $2w$. As a consequence, for this instance, the maximal matching algorithm is only $2w$-competitive.

In contrast, we show that using free dissolution, a constant competitive ratio is possible. To this end, we make use of a MWM algorithm again.

Theorem 13 (Badanidiyuru Varadaraja, 2011; Bullinger and Romen, 2023)**.** *For online* MWM *under free dissolution, the* $(1 + \frac{\sqrt{2}}{2})$*-dissolution threshold algorithm (DTA) is* $\frac{1}{3+2\sqrt{2}}$*-competitive*[1]*. No deterministic algorithm achieves a better competitive ratio.*

[1] Bullinger and Romen (2023) showed this version of the theorem in unpublished improvements of their work.

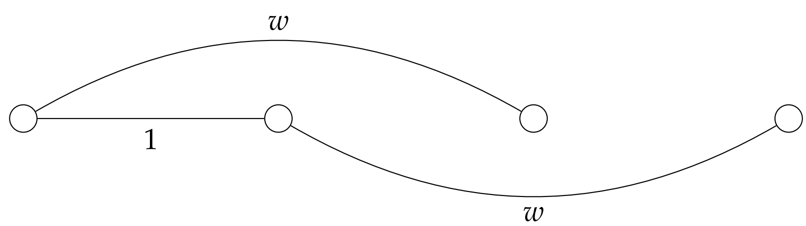

Figure 6.1: A bad instance for the maximal matching algorithm. The arrival order of the vertices is from left to right. The algorithm will match the edge of weight 1 because it arrives first. The optimal solution would be to match the two other edges, both of weight $w > 1$.

Together with Theorem 2 we obtain the following result.

Theorem 14. *For online* SocialWelfareFHG *with free dissolution, the* $(1+\frac{\sqrt{2}}{2})$*-DTA is* $\frac{1}{6+4\sqrt{2}}$*-competitive.*

The work of Badanidiyuru Varadaraja (2011) also allows for a convenient proof of an upper bound of the possible competitiveness.

Theorem 15. *For* SocialWelfareFHG *with free dissolution, no deterministic algorithm can achieve a competitive ratio better than* $\frac{1}{3+2\sqrt{2}}$.

Proof. The family of graphs that Badanidiyuru Varadaraja (2011) uses to show the upper bound of Theorem 13 consists of trees only. As we have seen in previous chapters, in trees we can insert edges of highly negative weight between vertices which did not share an edge before to arrive at an FHG instance where every partition with non-negative social welfare is a matching.

If we now assume that an algorithm could achieve a competitive ratio better than $\frac{1}{3+2\sqrt{2}}$ for FHGs with free dissolution, that algorithm would also achieve a better competitive ratio for MWM with free dissolution, which contradicts Theorem 13. □

7 Conclusion and Future Work

We have investigated the maximization of social welfare in FHGs both in an offline and online setting. In the offline setting, the optimal solution is not efficiently computable. The best known approximation ratio feasible in polynomial time remains to be $\frac{1}{2}$, achieved by computing a MWM. For simple FHGs, no better guarantee is known either, and even computing a minimum clique cover only is a $\frac{1}{2}$-approximation because of instances where it reduces to a MCM. On the side of hardness, we made progress by showing that no FPTAS exists. The exact inapproximability of the problem remains an interesting open question. All of this, of course, only holds under the assumption that P≠NP.

Since in the online setting under adversarial arrival, a constant competitive ratio is only possible for FHGs with bounded utilities, we provided improved results for the natural class of simple FHGs, showing that for symmetric simple FHGs, a randomized algorithm achieves $\frac{1}{4} + 2\epsilon$ (where $\epsilon > 0$ is a small constant) and outperforms all deterministic ones, where the known bound is $\frac{1}{4}$. The same algorithm achieves $\frac{1}{8} + \epsilon$ for general simple FHGs. With the goal of getting more positive results, we considered the online models of random arrival and free dissolution. In both versions, a constant competitive ratio is possible. For random arrival, we achieve a lower bound of $\frac{1}{6} - \mathcal{O}(\frac{1}{n})$ and show an upper bound of $\frac{1}{3}$. For simple FHGs, we even get a competitive ratio of $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$. For free dissolution, we achieve a lower bound of $\frac{1}{6+4\sqrt{(2)}}$ and show an upper bound of $\frac{1}{3+2\sqrt{(2)}}$. It remains to see whether these upper or lower bounds for FHGs are tight. Moreover, it would be interesting to study more restrictive models with bounded utilities or a bounded coalition size or number in these online settings. Or maybe, interesting results could be achieved in the framework of algorithms with predictions. Overall, in online models, matching algorithms seem to be the best approach for social welfare maximization. Last but not least, our study of the random arrival scenario has led to intriguing new results about matching with random arrival. There, we achieve a lower bound of $\frac{1}{3} - \mathcal{O}(\frac{1}{n})$ in the weighted case, which we show to be asymptotically tight. Since our proof of the upper bound only uses a family of bipartite graphs, the bounds also apply for the restriction of the setting to bipartite graphs. Note, however that this differs from the usual understanding of online bipartite matching where one side of the vertices is present offline. In the unweighted case, we achieve a lower bound of $\frac{2}{3} - \mathcal{O}(\frac{1}{n})$. It is an open question whether this bound is tight.

An interesting generalization of the free dissolution model would be to allow for arbitrary restructuring of existing coalitions but have the algorithm pay a cost accordingly. Investigating that setting would be a further step towards real-world applications. It has also not been studied yet, which guarantees can be achieved for MFHGs in online settings. Another parameter that could be changed for different results is the welfare notion. The maximization of Nash welfare or egalitarian welfare in online hedonic games has not been studied yet.

Abbreviations

FHG fractional hedonic game

ASHG additively separable hedonic game

MFHG modified fractional hedonic game

MCM maximum cardinality matching

MWM maximum weight matching

DTA dissolution threshold algorithm

FPTAS fully polynomial-time approximation scheme

List of Figures

List of Tables

List of Algorithms

Bibliography

Athanassopoulos, S., Caragiannis, I., Kaklamanis, C., & Kyropoulou, M. (2009). An improved approximation bound for spanning star forest and color saving. In R. Královi & D. Niwiski (Eds.), *Mathematical foundations of computer science 2009 (MFCS 2009)* (pp. 90–101, Vol. 5734). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-03816-7_9

Aziz, H., Brandl, F., Brandt, F., Harrenstein, P., Olsen, M., & Peters, D. (2019). Fractional hedonic games (D. Pennock & I. Segal, Eds.). *ACM Transactions on Economics and Computation (TEAC)*, *7*(6), 1–29. https://doi.org/10.1145/3327970

Aziz, H., Gaspers, S., Gudmundsson, J., Mestre, J., & Täubig, H. (2015, July 25). Welfare maximization in fractional hedonic games. In Q. Yang & M. Wooldridge (Eds.), *Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI 2015)* (pp. 461–467). AAAI Press.

Aziz, H., & Savani, R. (2016, May 5). Hedonic games. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, & A. D. Procaccia (Eds.), *Handbook of computational social choice* (pp. 356–376). Cambridge University Press. https://doi.org/10.1017/cbo9781107446984.016

Badanidiyuru Varadaraja, A. (2011). Buyback problem - approximate matroid intersection with cancellation costs. In L. Aceto, M. Henzinger, & J. Sgall (Eds.), *Automata, languages and programming: 38th international colloquium, ICALP 2011* (pp. 379–390). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-22006-7_32

Banerjee, S., Konishi, H., & Sönmez, T. (2001). Core in a simple coalition formation game. *Social Choice and Welfare*, *18*, 135–153. https://doi.org/10.1007/s003550000067

Bilò, V., Fanelli, A., Flammini, M., Monaco, G., & Moscardelli, L. (2015, May 4). On the price of stability of fractional hedonic games. In G. Weiss & P. Yolum (Eds.), *AAMAS'15: Proceedings of the 2015 international conference on autonomous agents and multiagent systems*.

Bilò, V., Fanelli, A., Flammini, M., Monaco, G., & Moscardelli, L. (2018). Nash stable outcomes in fractional hedonic games: Existence, efficiency and computation (F. Rossi, Ed.). *Journal of Artificial Intelligence Research*, *62*, 315–371. https://doi.org/10.1613/jair.1.11211

Bogomolnaia, A., & Jackson, M. O. (2002). The stability of hedonic coalition structures (E. Kalai, Ed.). *Games and Economic Behavior*, *38*, 201–230. https://doi.org/10.1006/game.2001.0877

Brandl, F., Brandt, F., & Strobel, M. (2015, May 4). Fractional hedonic games: Individual and group stability. In G. Weiss & P. Yolum (Eds.), *AAMAS'15: Proceedings of the 2015 international conference on autonomous agents and multiagent systems*.

Bullinger, M. (2020, May 13). Pareto-optimality in cardinal hedonic games. In A. E. F. Seghrouchni & G. Sukthankar (Eds.), *AAMAS'20: Proceedings of the 19th international conference on autonomous agents and multiagent systems* (pp. 213–221). International Foundation for Autonomous Agents; Multiagent Systems.

Bullinger, M., & Romen, R. (2023, August 30). Online coalition formation under random arrival or coalition dissolution. In I. L. Gørtz, M. Farach-Colton, S. J. Puglisi, & G. Herman (Eds.), *31st annual european symposium on algorithms (ESA 2023)* (27:1–27:18, Vol. 274). Schloss Dagstuhl – Leibniz-Zentrum für Informatik. https://doi.org/10.4230/LIPIcs.ESA.2023.27

Bullinger, M., & Romen, R. (2024, March 24). Stability in online coalition formation. In M. Wooldridge, J. Dy, & S. Natarajan (Eds.), *Proceedings of the thirty-eighth AAAI conference on artificial intelligence* (pp. 9537–9545, Vol. 38). Association for the Advancement of Artificial Intelligence (AAAI). https://doi.org/10.1609/aaai.v38i9.28809

Correa, J., Dütting, P., Fischer, F., & Schewior, K. (2019, June 17). Prophet inequalities for i.i.d. random variables from an unknown distribution. In A. Karlin (Ed.), *EC'19: Proceedings of the 2019 ACM conference on economics and computation* (pp. 3–17). Association for Computing Machinery (ACM). https://doi.org/10.1145/3328526.3329627

Cover, T. M. (1987). Pick the largest number. In T. M. Cover & B. Gopinath (Eds.), *Open problems in communication and computation* (p. 152). Springer-Verlag Berlin Heidelberg New York.

Drèze, J. H., & Greenberg, J. (1980). Hedonic coalitions: Optimality and stability (H. Sonnenschein, Ed.). *Econometrica: Journal of the Econometric Society*, *48*(4), 987–1003. https://doi.org/10.2307/1912943

Elkind, E., Fanelli, A., & Flammini, M. (2016, February 21). Price of pareto optimality in hedonic games. In D. Leake, J. Lester, Z. Kolter, C. Monteleoni, P. Doherty, & M. Ghallab (Eds.), *Proceedings of the thirtieth AAAI conference on artificial intelligence* (pp. 475–481, Vol. 30). Association for the Advancement of Artificial Intelligence (AAAI). https://doi.org/10.1609/aaai.v30i1.10048

Ezra, T., Feldman, M., Gravin, N., & Tang, Z. G. (2022, July 13). General graphs are easier than bipartite graphs: Tight bounds for secretary matching. In D. M. Pennock (Ed.), *EC'22: Proceedings of the 23rd ACM conference on economics and computation* (pp. 1148–1177). Association for Computing Machinery (ACM). https://doi.org/10.1145/3490486.3538290

Flammini, M., Kodric, B., Monaco, G., & Zhang, Q. (2021). Strategyproof mechanisms for additively separable and fractional hedonic games. *Journal of Artificial Intelligence Research (JAIR)*, *70*, 1253–1279. https://doi.org/10.1613/jair.1.12107

Flammini, M., Monaco, G., Moscardelli, L., Shalom, M., & Zaks, S. (2021). On the online coalition structure generation problem. *Journal of Artificial Intelligence Research (JAIR)*, *72*, 1215–1250. https://doi.org/10.1613/jair.1.12989

Gabow, H. N., & Tarjan, R. E. (1991). Faster scaling algorithms for general graph matching problems. *Journal of the ACM*, *38*(4), 815–853. https://doi.org/10.1145/115234.115366

Gamlath, B., Kapralov, M., Maggiori, A., Svensson, O., & Wajc, D. (2019). Online matching with general arrivals. In Y. Rabani (Ed.), *2019 IEEE 60th annual symposium on foundations of computer science (FOCS)* (pp. 26–37). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/FOCS.2019.00011

Huang, Z., Tang, Z. G., & Wajc, D. (2024). Online matching: A brief survey (I. Lo & S. Taggart, Eds.). *ACM SIGecom Exchanges*, *22*(1), 135–158.

Kaklamanis, C., Kanellopoulos, P., & Papaioannou, K. (2016). The price of stability of simple symmetric fractional hedonic games. In M. Gairing & R. Savani (Eds.), *Algorithmic game theory: 9th international symposium* (pp. 220–232). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-53354-3_18

Kaplan, H., Naori, D., & Raz, D. (2020). Competitive analysis with a sample and the secretary problem. In S. Chawla (Ed.), *Proceedings of the fourteenth annual ACM-SIAM symposium on discrete algorithms (SODA)* (pp. 2082–2095). Society for Industrial; Applied Mathematics (SIAM). https://doi.org/10.1137/1.9781611975994.128

Karp, R. M. (1972). Reducibility among combinatorial problems. In R. E. Miller, J. W. Thatcher, & J. D. Bohlinger (Eds.), *Complexity of computer computations: Proceedings of a symposium on the complexity of computer computations* (pp. 85–103). Springer, Boston, MA. https://doi.org/10.1007/978-1-4684-2001-2_9

Kesselheim, T., Radke, K., Tönnis, A., & Vöcking, B. (2013). An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In H. L. Bodlaender & G. F. Italiano (Eds.), *Algorithms — ESA 2013* (pp. 589–600). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40450-4_50

Moran, S., Snir, M., & Manber, U. (1985). Applications of ramsey's theorem to decision tree complexity. *Journal of the ACM (JACM)*, *32*, 938–949. https://doi.org/10.1145/4221.4259

Newman, M. E. J. (2004). Detecting community structure in networks. *The European Physical Journal B: Condensed Matter and Complex Systems*, *38*, 321–330. https://doi.org/10.1140/epjb/e2004-00124-y

Olsen, M. (2012, January 31). On defining and computing communities. In J. Mestre (Ed.), *Proceedings of the eighteenth computing: The australasian theory symposium* (pp. 97–102). Australian Computer Society, Inc.

Ramsey, F. P. (1930). On a problem of formal logic. *Proceedings of the London Mathematical Society*, *s2-30*, 264–286. https://doi.org/10.1112/plms/s2-30.1.264