2021

Novel Hedonic Games and Stability Notions

Jacob Schlueter
University of Kentucky, tjacobschlueter@gmail.com
Author ORCID Identifier:
https://orcid.org/0000-0002-8365-4186

Digital Object Identifier: https://doi.org/10.13023/etd.2021.116

Jacob Schlueter, Student

Dr. Judy Goldsmith, Major Professor

Dr. Zongming Fei, Director of Graduate Studies

Novel Hedonic Games and Stability Notions

DISSERTATION

A dissertation submitted in partial
fulfillment of the requirements for
the degree of Doctor of Philosophy
in the College of Engineering at the
University of Kentucky

By
Jacob Schlueter
Lexington, Kentucky

Director: Dr. Judy Goldsmith, Professor of Computer Science
Lexington, Kentucky 2021

ABSTRACT OF DISSERTATION

Novel Hedonic Games and Stability Notions

We present here work on matching problems, namely hedonic games, also known as coalition formation games. We introduce two classes of hedonic games, Super Altruistic Hedonic Games (SAHGs) and Anchored Team Formation Games (ATFGs), and investigate the computational complexity of finding optimal partitions of agents into coalitions, or finding – or determining the existence of – stable coalition structures. We introduce a new stability notion for hedonic games and examine its relation to core and Nash stability for several classes of hedonic games.

KEYWORDS: Computational Social Choice, Cooperative Game Theory, Hedonic Games

Author's signature: Jacob Schlueter

Date: May 11, 2021

Novel Hedonic Games and Stability Notions

By
Jacob Schlueter

Director of Dissertation: Judy Goldsmith

Director of Graduate Studies: Zongming Fei

Date: May 11, 2021

ACKNOWLEDGMENTS

I would like to thank my PhD mentor, Dr. Judy Goldsmith for her guidance and advice over the past several years. I would also like to thank Christian Addington, my undergraduate research assistant who has been a great help since he joined our research group more than a year ago. I also want to thank my family and friends for helping me find the motivation to complete my studies, even when I was struggling.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

This dissertation discusses matching problems that seek to assign individuals to groups based on those individuals' preferences. The goal in these problems is to find matchings that are unlikely to break down. Cooperative game theory provides the tools necessary to analyze and solve the problems.

Cooperative game theory models challenges in which agents must cooperate with each other agents to achieve a certain goal; this stands in contrast with non-cooperative game theory wherein there is no such expectation. Cooperative game theory has been applied to a wide variety of real-world challenges such as voting, resource allocation, and organization of teams.

An important consideration for any cooperative game, from the perspective of both player and modeler, is whether or not utility is transferable. In cooperative games with transferable utility, agents may discuss how to distribute the utility resulting from their cooperation. A classic example of transferable utility is a game with monetary winnings, since there are many ways to divide a single amount of money. An example of non-transferable utility is the enjoyment of being around a friend; friendship itself cannot be transferred between people at will, and there's no guarantee that the enjoyment people get from spending time with friends is consistent between individuals.

Our research focuses on cooperative games where the goal is to divide people into groups. Many situations exist wherein individuals will choose to act as a group, or coalition. Examples include social clubs, political parties, marriage partner selection, and legislative voting [9, 27, 68]. Matching problems which can model these situations are a common topic of interest in multiagent systems research since Gale and Shapley's work on college admission and the stable marriage problem [27]. Within the area of matching, we study coalition formation games; a generalization of the stable marriage problem studied by Gale and Shapley where the goal is assigning agents to coalitions that may contain many agents rather than just two. We are interested in a subclass of coalition formation games, hedonic games, which were first proposed by Drèze and Greenberg (1980) and later formalized by Banerjee, Konishi, and Sönmez (2001) and Bogomolnaia and Jackson (2002). Hedonic games are distinguished from general coalition formation games by the requirement that each agent's utility is wholly derived from the members of their own coalition [6, 9, 21].

A central problem in hedonic games research, and for coalition formation games in general, is deciding whether or not a proposed set of coalitions, or partition, is *stable* [68], meaning that agents will not leave, or deviate, from their coalitions. There are a number of factors to consider when trying to decide whether a partition is stable or not. Do agents always act alone, or do they act in concert with others? Do agents actively pursue better coalitions, or will they only deviate if their coalition is particularly bad? Do agents need the approval of others in order to join a new coalition or leave their current coalition? These are some of the most common factors to consider when evaluating the stability of a partition. Several concepts have been introduced to characterize the ways in which a partition is or is not stable, each with its own unique assumptions about how agents tend to deviate. Woeginger's (2013) survey and book chapters by Aziz and Savani (2016) and by Elkind and Rothe (2016) provide overviews of several stability notions [4, 22, 68]. One stability concept that is of perennial interest in coalition formation games is the core (See Definition 29). [3, 6, 9, 19, 20, 28, 29, 30, 45]. Stability concepts relevant to our work are formally introduced in section 2.3.

1.1 Questions of interest

In hedonic games research, there are a number of questions that researchers investigate. Here we briefly discuss a number of questions related to *stability*.

One of the most common questions is whether or not stable partitions are guaranteed to exist. The answer to this question varies depending on the type of hedonic game and the notion of stability being investigated. A related question is how difficult it is to verify whether a given partition is stable or not. We could also investigate how hard it is to determine whether a stable partition exists for a particular game.

While the questions thus far center on whether *any* stable partitions exist, we could also ask *how many* stable partitions there are. Related to this question is how hard it is to determine how many stable partitions exist. As with questions on whether any stable partitions exist, the answer to how many stable partitions exist will vary based on the type of game and the type of stability one is investigating.

1.2 Contributions of This Dissertation

We introduce super altruistic hedonic games (SAHGs), hedonic games where an agent's utility is impacted by the preferences of all agents in their coalition, as a natural extension of altruistic hedonic games (AHGs, see Definition 17) proposed by Nguyen et al. [45]. The central notion behind AHGs is that an agent's happiness is impacted by their friends' happiness and vice-versa. SAHGs extend this notion further to consider that an agent's happiness will be impacted by the happiness of all other agents in their coalition.

We introduce anchored team formation games (ATFGs) to evaluate coalition formation settings where the coalitions must contain a leading agent, or anchor. ATFGs are a completely new class of hedonic games inspired by tabletop role playing games such as Dungeons & Dragons and Pathfinder, wherein a group of individuals must contain a leader, or game master (GM), in order to play. We design one complete solver and three heuristic solvers for Nash stability (see Definition 24) in ATFGs. Complete solvers will always find a solution when one exists, but they scale poorly since they systematically evaluate the entire state space of the game. Heuristics are not guaranteed to find solutions, even when they exist, but they scale much better than complete solvers, which often makes them better in practice.

We introduce internal stability for partitions, a stability notion which examines whether each coalition in a partition is stable, which builds on work by Dimitrov et al. and Alcalde and Romero-Medina that introduced the notion of internal stability for coalitions. Alcalde and Romero-Medina use internally stable coalitions as a tool to investigate conditions that guarantee the existence of core stable partitions in hedonic games [1]. Dimitrov et al. use internal stability to define another stability notion, deviation stability, which they use to prove the existence of core stable partitions in friend and enemy-oriented hedonic games [20]. Since its introduction, the only other paper we have found which utilizes internal stability is a 2014 paper by Liu et al. that adapts it to matching and exchange contexts [41].

1.3 Dissertation Roadmap

We begin with Chapter 2 by covering definitions and terminology relevant to our work. After defining these preliminaries, we provide a survey of related works in Chapter 3. Next, we move into our work on new classes of hedonic games. In Chapter 4 we introduce SAHGs and present several complexity results. In Chapter 5 we introduce ATFGs and present some complexity results and results from our heuristic solvers. We then move on to our work with internal stability in Chapter 6 where we distinguish it from other stability notions and provide results for the price of internal stability. Finally, in Chapter 7 we conclude with final remarks and directions for future research.

Chapter 2 Definitions

Here, we cover a range definitions and terminology relevant to our work. We start by introducing relevant complexity classes, then move on to formally introduce hedonic games and classes of hedonic games relevant to our work. Next, we introduce several stability notion followed by three notions of optimality. We end the Chapter with descriptions of the price of stability and price of anarchy.

2.1 Complexity Definitions

Complexity theory provides a structure to describe and categorize computational problems based on the amount of time or space required to solve them. This helps us understand whether or not a problem can be solved in a reasonable amount of time with the resources available. In this section, we introduce time complexity classes utilized throughout the document. For more background see Papadimitriou's Computational Complexity [48].

Definition 1. *[48] Time complexity class **P** contains problems solvable in time that is polynomial based on the size of the input. Any problem with an upper time complexity bound of $O(n^p)$ where p is a fixed numeric value is contained in this class.*

Problems that fall within the time complexity class P are considered tractable. The closely related class NP contains all problems in P and introduces problems that may be intractable.

Definition 2. *[48] Complexity class **NP** contains problems for which a proposed solution's validity can be checked in polynomial time. A non-deterministic algorithm can guess a valid solution and confirm it in polynomial time, but a deterministic algorithm may be unable to find a valid solution in polynomial time.*

Many well-known problems fall within the NP complexity class. Some examples of problems that fall into the NP complexity class include General Satisfiability (SAT), Hamilton Path, Clique, and Exact Cover by 3-Sets [48].

Definition 3. *[48] Complexity class* ***coNP*** *is the class of languages whose complements are in NP.*

Exampmles of coNP problems include the SAT Complement and Hamilton Path Complement problems [48].

Definition 4. *[66] Complexity class* ***#P*** *contains all functions that compute the number of solutions to an NP problem.*

The #P complexity class is of interest, because problems that may admit an easy means to compute a solution may remain difficult when the goal becomes finding how many solutions exist.

Definition 5. *[31] Complexity class* Θ_2^p *is an alternative name for* $P^{NP[log]}$*. Games in this class are solvable by a P machine that can make* $O(\log n)$ *queries to an NP oracle.*

Before we introduce the last complexity class of interest, we first define concepts necessary to understand it.

Definition 6. *[57] A language A is* ***mapping reducible*** *to language B, written* $A \leq_m B$*, if there is a computable function* f *such that* $\forall w \in A : f(w) \in B$*.*

We are primarily interested in cases where the function f mapping between two languages is computable in polynomial time, typically written as $A \leq_m^P B$ when describing reducibility between two languages or as $\leq_m^P$ in more general contexts.

Definition 7. *[57] A language A is* ***Turing reducible*** *to language B, written* $A \leq_T B$*, if A is decidable relative to B.*

As with mapping reductions, we are primarily concerned with polynomial time Turing reductions.

Definition 8. *A problem is said to be **complete** for a complexity class if and only if it is in the complexity class and every other problem in the same complexity class is $\leq_m^P$-reducible to it.*

Because all problems in a complexity class are $\leq_m^P$-reducible to problems that are complete for the class, problems that are complete for a given complexity class are the hardest problems contained within that class. For example, the Exact Cover by 3-Sets problem is NP-complete in the general case [48]. This means that no problem within the NP complexity class is more computationally complex than general case Exact Cover. This also means that all other problems in NP can be reduced to Exact Cover. Further, all NP-complete problems are $\leq_m^P$-reducible to Exact Cover and vice versa.

Definition 9. ***Hardness** for a complexity class is a term used to define problems that generalize the hardest, or complete, problems for that class.*

Consider Exact Cover by 3-Sets, which is NP-complete. Exact Cover by 3-Sets is also NP-hard, because it generalizes all other NP-complete problems. However, the term NP-hard is more commonly used to refer to problems that generalize NP-complete problems, but which are not proven to be contained in NP.

Definition 10. *[47] The complexity class **DP** contains languages defined as the difference between two languages in NP.*

For example, let C be an NP-complete language, and let $L = \{\langle c_1, c_2\rangle : c_1 \in C \wedge c_2 \notin C\}$. Then $L = \{C \times \Sigma^*\} \setminus \{\Sigma^* \times C\}$ (where Σ^* is the set of all strings over the alphabet used to define C).

There is a significant body of work within cooperative games research on coalition formation games. The goal of coalition formation games is to partition a set of agents into coalitions, based on the individual agents' preferences. Our work focuses on hedonic coalition formation games, often shortened to hedonic games, where agents' preferences are limited to their own coalitions and utility is non-transferable.

2.2 Hedonic Games

In this section, we formally define hedonic games, and introduce a few specific hedonic games that were inspirations for our work. This includes Altruistic Hedonic Games and Additively Separable Hedonic Games, as well as several gaming-inspired hedonic games.

Below, we outline several classes of hedonic games. In each class, a particular game G consists of

1. N, a finite set of n agents, with

2. preference set $P = \{P_i : \ i \in N\}$, where P_i is the preference of each agent i over partitions of N into coalitions. P_i may exhaustively list the preferences of agent i or provide a succinct representation from which preferences are derived. Additionally, preferences over coalitions may be strict, meaning that for any pair of coalitions C_1, C_2 containing an agent i either $C_1 \succ_i C_2$ or $C_2 \succ_i C_1$; alternatively, preferences may permit indifference where it is possible to have some C_1, C_2 such that $C_1 \succeq_i C_2$ and $C_2 \succeq_i C_1$ (equivalent to $C_2 \sim_i C_1$).

3. When preferences are given as utilities, we generally assume that $u_i(\{i\}) = 0$: for each i, the utility of agent i for being in a coalition of size 1, is 0.

Definition 11. *[6, 9]* ***Hedonic games*** *are coalition formation games with nontransferable utility wherein players' preferences are concerned only with their own coalition.*

This inherently self-interested means of determining utility makes such games hedonic in nature.

Let $\mathcal{N}_i$ be the set of possible coalitions containing agent $i \in N$. A preference ordering of $\mathcal{N}_i$ is derived from the preference set $P_i \in P$. A solution for a game is a partition π, which is contained in the set of all distinct partitions Γ. Each player $i \in N$ has preferences over all partitions $\pi \in \Gamma$ based solely on their assigned coalition in each π.

Hedonic games are a broad category, so it can be useful to define classes of hedonic games that exhibit certain interesting or useful properties. Additively separable hedonic games are a well-studied class of hedonic games that is relevant to our work.

Definition 12. *[6]* ***Additively Separable Hedonic Games (ASHG)*** *are a class of hedonic game where each agent $i \in N$ assigns values to each agent $j \in N$, expressed as $v_i(j)$; $v_i(i)$ is always set to 0. The utility an agent derives from each $S \in \mathcal{N}_i$ is defined as $u_i(S) = \sum_{j \in S} v_i(j)$.*

$\mathcal{B}$ and $\mathcal{W}$ Games are two more classes of hedonic game where agents assign some value to each other, as in ASHGs. Instead of deriving utility from all the agents in a coalition, utility in $\mathcal{B}$ and $\mathcal{W}$ Games is derived from the single highest and least valued agents, respectively.

Definition 13. *[15]* ***$\mathcal{B}$ and $\mathcal{W}$ Games*** *are hedonic games where agents assign values to each other, much like in ASHGs. In $\mathcal{B}$ Games the utility some $i \in N$ derives from a coalition C is equal to $\max_{j \in C \setminus \{i\}} v(j)$; in $\mathcal{W}$ Games the utility is equal to $\min_{j \in C \setminus \{i\}} v(j)$.*

Fractional hedonic games are another class of hedonic games that builds off the definition of ASHGs. Instead of taking the value assigned to other agents as utility in and of itself, agents derive utility from the average value they assign to agents in their coalition.

Definition 14. *[3]* ***Fractional Hedonic Games (FHGs)*** *are a class of Hedonic Games where agents assign values to each other agent. In contrast to ASHGs, Fractional Hedonic Games define utility as an average rather than a sum:*

$$u_i(S) = \frac{\sum_{j \in S} v_i(j)}{|S|}.$$

While FHGs take the notion of agents assigning some value to each other in a different direction than ASHGs, there are also sub-classes of ASHGs that have been studied.

Definition 15. *[20]* ***Friend-oriented Hedonic Games (FOHGs)*** *are a class of hedonic game where each agent regards all other agents as either a friend or an enemy. FOHGs are often represented by graphs where an edge from some agent $i \in N$ to another agent $j \in N$ indicates that i regards j as a friend. Lack of an edge from i to j indicates that i regards j as an enemy. Utility for each agent is the sum of values they assign to other agents, friends being assigned a value of n while enemies are valued at -1.*

Definition 16. *[20]* ***Enemy-oriented Hedonic Games (EOHG)*** *are another class of graph-based hedonic game that follows the same basic principles of FOHGs; each agent views each other agent as a friend or an enemy. However, in EOHGs a friend's value is 1, while an enemy's value is $-n$.*

FOHGs and EOHGs inspired a number of modifications and extensions to their original conception, such as the introduction of neutral agents in work by Lang et al., Ohta et al., and unknowns by Barrot et al. [7, 39, 46]. Altruistic hedonic games are an extension of the FOHG concept that considers not just how many friends are in a coalition, but whether those friends are happy with the coalition or not [45]. The motivations behind altruistic hedonic games are a major inspiration for Super Altruistic Hedonic Games (SAHGs), which we examine in Chapter 4.

Definition 17. *[45] An* ***Altruistic Hedonic Game (AHG)*** *is a hedonic game in which agents derive utility from both their own basic preferences and those of any friends in the same coalition.*

Let each agent $i \in N$ have utility u_i, and let i partition other agents into friends and enemies, given by F_i, E_i. Three levels of altruism are considered in AHGs: ***selfish-first****,* ***equal treatment****, and* ***altruistic first****. The function used to determine an agent's utility depends on their altruism level and on pre-utility preference values calculated as the utility agents would have from some coalition C in a FOHG based on the same graph ($n|C \cap F_i| - |C \cap E_i|$). Two of these functions utilize a weight parameter of $M = n^5$ to ensure that one of the terms in the equation dominates the other. This weight value is the smallest whole number exponent of n which guarantees this for both equations that make use of M. Definitions for each altruism level and their utility functions are outlined below:*

1. ***Selfish-First:*** *agents prioritize their own preferences, but use the preferences of others to break ties.*

$$u_i = M(n|C \cap F_i| - |C \cap E_i|) + \sum_{a \in C \cap F_i} \frac{n|C \cap F_a| - |C \cap E_a|}{|C \cap F_i|}$$

2. ***Equal Treatment:*** *all preferences are treated equally.*

$$u_i = \sum_{a \in C \cap (F_i \cup \{i\})} \frac{n|C \cap F_a| - |C \cap E_a|}{|C \cap (F_i \cup \{i\})|}$$

3. ***Altruistic First:*** *agents prioritize the preferences of others, but use their own preferences to break ties.*

$$u_i = n|C \cap F_i| - |C \cap E_i| + M \cdot \sum_{a \in C \cap F_i} \frac{n|C \cap F_a| - |C \cap E_a|}{|C \cap F_i|}$$

AHGs introduce some interesting ideas by incorporating the preferences of others into utility computations in a polynomially computable fashion. The three levels of altruism provide a means to vary the degree to which agents consider the preferences of others, while also providing bounds on the weights needed to ensure the dominance of one term in the utility equation. However, only considering the preferences of friends and three variations of altruism limits the preferences and degrees of altruism that can be represented. In order to broaden the scope of representation, we combined ideas from AHGs and social distance games in order to create SAHGs (see Chapter 4). Social distance games are a class of hedonic games introduced in 2011 where agents are concerned with how socially close they are to their coalition members.

Definition 18. *[11] **Social Distance Games (SDG)** are a class of hedonic game where an agent's utility is based on their* social distance *from other agents in their coalition. A SDG instance is represented by an undirected, unweighted graph $G = (N, E)$. For any agent pair i, j in some coalition $C \subseteq N$ the social distance between i and j, $d_C(i, j)$, is the shortest path distance between them in the sub-graph $G_C = (C, E_C)$ where E_C contains all edges $(i, j) \in E$ such that $i, j \in C$. The utility some agent $i \in N$ derives from coalition $C \subseteq N : i \in C$ is defined as follows:*

$$u_i(C) = \frac{1}{|C|} \sum_{j \in C \setminus \{i\}} \frac{1}{d_C(i, j)}$$

Thus far, all games we have defined rely on utility values associated with each unique agent, but some hedonic games take a different approach. Roles and Teams Hedonic Games (RTHGs) are one example inspired by the game League of Legends, first introduced by Spradling et al.. In RTHGs, an agent's utility is based on the roles that make up their team and the role they are assigned to [60].

Definition 19. *[60] A **Roles and Teams Hedonic Game (RTHG)** instance consists of:*

- *P: A population of agents*
- *m: a team size (we assume that $|P|/m$ is an integer)*

- *R: a set of available member team roles*

- *C: a set of available team compositions, where a* team composition *is a set of m not necessarily unique roles in R.*

- *U: a utility function vector $\langle u_0, ..., u_{|P|-1} \rangle$, where for reach agent $p \in P$, composition $t \in C$, and role $r \in R$ there is a utility function $u_p(t, r)$ with*

$$u_p(t, r) = -\infty \text{ if } r \notin t.$$

A solution to an RTHG instance is a partition π of agents into teams of size m.

Later work by Spradling and Goldsmith introduced Role Based Hedonic Games, a generalization of RTHGs that does not impose a strict team size [59].

Definition 20. *[59] A* ***Role Based Hedonic Game (RBHG)*** *instance consists of:*

- *P: A population of agents*

- *R: a set of roles*

- *C: a set of available team compositions, where a composition $c \in C$ is a multiset (bag) of roles from R.*

- *U: $P \times R \times C \to \mathbb{Z}$ defines the utility function $u_i(r, c)$ for each player p_i. We assume that for all $p_i \in P$ and for all $r \in R$, $u_i(r, \{r\}) = 0$.*

Siler introduced Tiered Coalition Formation Games (TCFGs), another class of games inspired by the Pokémon series of games.

Definition 21. *[56] A* ***Tiered Coalition Formation Game (TCFG)*** *instance is a pair $(A, \succeq)$ consisting of a finite set of agents $A = \{a_1, a_2, ..., a_n\}$ and a preference profile $\succeq = \{\succeq_1, \succeq_2, ..., \succeq_n\}$, where $\succeq_i$ is a weak total order over subsets of A that contain a_i.*

*A (*tier list*) consists of a partition of A into a totally ordered set of k disjoint coalitions (*tiers*) $T = \{T_1, T_2, ..., T_k\}$.*

In a tier list T, we say that a_i sees a_j if a_i is in the same tier as a_j or a higher tier: Seen*$(a_i, T) = \cup_{l=1}^{m} T_l, a_i \in T_m$. For tier lists T and T', $\succeq_i$ specifies a_i's preference between* Seen*(a_i, T) and* Seen*(a_i, T').*

The Group Activity Selection Problem proposed by Darmann et al. presents another distinct approach to hedonic coalition formation [18].

Definition 22. *[18] An instance of the* ***Group Activity Selection Problem (GASP)*** *is given by a set of agents $N = \{1, ..., n\}$, a set of activities $A = A^* \cup \{a_\emptyset\}$, where $A^* = \{a_1, ..., a_p\}$, and a* profile *P, which consists of n* votes *(one for each agent): $P = (V_1, ..., V_n)$. The vote of agent i describes their preferences over the set of* alternatives *$X = X^* \cup \{a_\emptyset\}$, where $X^* = A^* \times \{1, ..., n\}$; alternative (a, k), $a \in A^*$, is interpreted as "activity a with k participants," and $a_\emptyset$ is the* void *activity.*

The vote V_i of an agent $i \in N$ (also denoted by $\succeq_i$) is a weak order over X^*; its induced strict preference and indifference relations are denoted by $\succ_i$ and $\sim_i$ respectively. We set $S_i = \{(a, k) \in X^* | (a, k) \succ_i a_\emptyset\}$; we say that voter i approves *of all alternatives in S_i, and refer to the set S_i as the* induced approval vote *of voter i.*

2.3 Stability

One of the major topics of hedonic games is *stability*, the idea that a partition will not be disrupted by individuals rejecting their assigned coalitions and moving to other coalitions. There are many sets of constraints placed on such disruptions, such as the number of agents that can move simultaneously; whether all moving agents must see an increase in utility; whether agents left behind by movers must see their utility increase, or whether agents being joined by movers must see their utility improve. We examine the stability of Super Altruistic Hedonic Games in Chapter 4 and Anchored

Team Formation Games in Chapter 5. We also introduce a new notion of stability in Chapter 6.

We next define stability notions used in this work. In these definitions π is a partition composed of a set of k disjoint coalitions $\{C_1, C_2, ...C_k\}$. The term $\pi(i)$ refers to the coalition $C \in \pi$ such that $i \in C$.

Definition 23. *[22] A partition is* ***individually rational*** *if no individual agent can improve their utility by leaving their current coalition to become a singleton.*

Individual rationality only considers the possiblity that agents will leave a unsatisfactory coalition to become a singleton. However, there may be cases where agents can join other coalitions instead of simply leaving their current one. Nash stability, adapted to the hedonic games space by Bogomolnaia and Jackson, considers the possibility that individuals may move between coalitions if doing so will improve their utility.

Definition 24. *[9] A partition is* ***Nash stable*** *for a coalition formation game if no individual agent can improve their utility by deviating from their current coalition to join another coalition or to become a singleton.*

Nash stability focuses only on the selfish behavior of an individual, meaning that agents can freely join or leave coalitions regardless of how such actions impact other agents. This freedom of movement between coalitions may not be realistic in all cases and imposing conditions that must be satisfied before an agent can move between coalitions may cause a coalition that is not Nash stable to become stable. Bogomolnaia and Jackson propose stability notions that impose such conditions [9].

Definition 25. *[9] A partition is* ***individually stable*** *if no agent can improve their utility by joining a different coalition without also decreasing the utility of at least one agent in their new coalition.*

A partition is individually stable when agents must receive permission from all members of a coalition they seek to join. Accordingly, agents will only grant permission to join if their own utility either increases or remains the same.

Definition 26. *[9] A partition is* ***contractually individually stable*** *if no agent can improve their utility by deviating from their current coalition without also negatively impacting the utility of one or more agents in either their current coalition or the coalition they seek to join.*

A partition is contractually individually stable when agents must receive permission from all agents in a coalition in order to join it or leave it. As with individual stability, an agent will only grant permission to join or to leave if doing so will not negatively impact their utility. One way to ensure that a coalition is stable is to guarantee that all agents in a coalition mutually benefit from each others' presence. Woeginger proposes a stability notion for graph based games based on this idea [68].

Definition 27. *[68] Partitions in games based on an undirected graphs are* ***wonderfully stable*** *when all coalitions in the partition are maximal (non-extendable) cliques.*

Stability notions discussed thus far have been focused primarily on whether individual agents can improve their utility by deviating from their current coalition. Another possibility to consider is whether or not groups of agents can improve their utility through cooperative deviation from the status quo. Bogomolnaia and Jackson introduced the notions of blocking coalitions, and core stability, building on work by Greenberg and Weber, and by Demange [9, 19, 29].

Definition 28. *[9] A coalition* ***blocks*** *a partition if all agents in the coalition prefer it over their current coalitions. Formally, a coalition C* ***blocks*** *partition π if $\forall i \in C : C \succ_i \pi(i)$.*

It is possible to understand core stability without first explaining the notion of a blocking coalition, but an understanding of blocking coalitions simplifies the definition of core stability quite a bit.

Definition 29. *[9] A partition is* ***core stable*** *if no coalition blocks it.*

Dimitrov et al. proposed a stronger stability notion that utilizes weaker criteria for a blocking coalition [20].

Definition 30. *[20] A coalition* ***weakly blocks*** *a partition if at least one agent strictly prefers the new coalition over their current coalition and all other agents in the coalition either prefer the new coalition over their current coalitions or are indifferent between the two. Formally, a coalition C weakly blocks partition π if $\forall i \in C : C \succeq_i \pi(i)$ and $\exists j : C \succ_j \pi(j)$.*

As with core stability, understanding the notion of a weakly blocking coalition makes defining strict core stability easier.

Definition 31. *[20] A partition is* ***strictly core stable*** *if no coalition weakly blocks it.*

The previously discussed stability notions have focused on the idea that agents must be incapable of achieving better outcomes than the present, potentially due to conditions that must be satisfied in order to improve their utility. Another way of thinking about stability is to treat it like a voting problem, selecting the partition with the most support from the overall set of agents. Lang et al. propose a stability notion based on the idea of popular support [39].

Definition 32. *[39] A partition π is* ***strictly popular*** *if it beats all other partitions $\pi' \neq \pi$ in pairwise comparisons. A partition π beats another partition π' in a pairwise comparison if and only if the number of agents that prefer π over π' is greater than the number of agents that prefer π' over π.*

While stable outcomes are unlikely to change, they don't necessarily guarantee that everyone gets the best outcome possible.

2.4 Optimality

Optimality, the notion of finding a utility-maximizing partition, is another major topic of hedonic games. Notions of optimality are subject to constraints which clarify what is being optimized, such as whether individual or collective (egalitarian or utilitarian) utility is being optimized, or whether utility can be improved for some at the expense of others (Pareto efficiency).

We next define notions related to optimality that are referenced throughout the proposal. In these definitions π is a partition composed of a set of k disjoint coalitions $\{C_1, C_2, ...C_k\}$. The utility an agent derives from their coalition $C \in \pi$ is represented by $u_i(\pi)$. The **total utility** of a partition is the sum of all agents' utilities, defined by $U_T = \sum_{i \in N} u_i(\pi)$.

Definition 33. *A partition π is **Pareto optimal** if no other partition that gives all agents at least as much utility as they receive in π and gives at least one agent more utility than they get in π [43].*

Another optimality notion proposed by Aziz et al. focuses on the sum total utility across all agents in a game [2].

Definition 34. *A partition π is optimal according to **utilitarian social welfare** if the sum of the utilities derived from π by all agents is greater than the sum of the utilities derived from all other partitions [2].*

Another natural optimality consideration is one which seeks to maximize the minimum amount of utility that any agent receives.

Definition 35. *A partition π is **maximin optimal** if the minimum amount of utility any agent receives under π is greater than or equal to the minimum amount of utility received in all other partitions.*

2.5 Stability versus Anarchy

Notions of stability are useful tools for predicting outcomes. However, there may be costs that result from the imposition of a given stability notion. Further, a given stability notion may admit good outcomes, but may also permit particularly subpar outcomes as well. The notions of price of stability and price of anarchy formally define these ideas.

Definition 36. *For stability notion X, the **price of stability ($\mathbf{PoS}_X$)** is the ratio between the overall utility-maximizing partition and the utility-maximizing X-stable partition. For example, we use PoS_{IS} to represent the price of internal stability.*

Where the price of stability examines the loss of utility required to satisfy a given stability notion, the price of anarchy examines the worst-case outcome of a given stability notion.

Definition 37. *For stability notion X, the **price of anarchy ($\mathbf{PoA}_X$)** is the ratio between the overall utility-maximizing partition and the utility-minimizing X-stable partition. For example, PoA_{IS} represents the price of anarchy for internal stability.*

In Chapter 3 we cover a range of works in the hedonic games field that are relevant to our research.

Chapter 3 Related Works

Our work is in the area of hedonic coalition formation games, often shortened to hedonic games. Hedonic games are a broad category of coalition formation games, formally codified by Banerjee et al. and Bogomolnaia and Jackson in 2001-2002, where each player's utility is wholly derived from their coalition and is non-transferable [6, 9]. Non-transferable utility means that agents cannot share the benefits they derive from the coalition directly with other agents. For example, the utility Hikaru derives from being in a group with their friend cannot be transferred to someone else. In coalition formation games with transferable utility, the benefits agents gain from their coalition can be readily divided up and shared. For example, if the utility gained through group formation is money, then it can be divided up however the group members like.

General case hedonic games can model a wide variety of problems, but it is difficult to generalize about the computational complexity of determining the existence, or finding, stable or optimal partitions over all hedonic games. However, there are some results for general hedonic games, which provide upper bounds on the complexity for all games they generalize, such as work by Ballester in 2004 which proved that deciding whether stable partitions exist is NP-complete for core stability, Nash stability, and individual stability [5].

Much hedonic games research focused on subclasses that exhibit certain useful properties. Early examples, which predate the formal codification of hedonic games, are Gale and Shapley's stable marriage and stable roommates problems introduced in 1962, which always have polynomially computable stable matchings [27]. In both problems players rank each other based on whom they want to be paired with, which restricts their scope and applicability; stable marriage further restricts rankings to members of a different gender [27] or other set in some bipartite graph (e.g., medical residents and hospitals [52]). Gale and Shapley proved that, when agents' preferences over who to marry or room with are strict, stable assignments always exist and also provided an $O(n^2)$ algorithm to compute such stable assignments.

Around the same time that hedonic games were first formalized, $\mathcal{B}$ and $\mathcal{W}$ games were introduced in 2001 by Cechlárová and Romero-Medina to address the stable roommates problem when it is not restricted to pairs [15]. Cechlárová and Romero-Medina provide polynomial time algorithms to compute strictly core stable partitions for $\mathcal{B}$ and $\mathcal{W}$ games with strict preferences when they exist. In 2003 Cechlárová and Hajduková demonstrated that determining if stable partitions exist for $\mathcal{B}$ games is NP-complete when preferences are not strict [13]. In 2004 Cechlárová and Hajduková expand on $\mathcal{W}$ games research by proving that, when preferences are not strict, determining the existence of core stable partitions is NP-complete [14].

Additively separable hedonic games (ASHGs, see Definition 12) were introduced by one of the seminal papers on hedonic games by Banerjee et al. in 2001 to see how the restriction of additively separable utility affected the existence of core stable partitions, and proved that ASHGs are not guaranteed to have core stable partitions [6]. Dimitrov et al. introduced friend oriented and enemy oriented hedonic games (FOHGs and EOHGs see Definitions 15 and 16), in 2006 as two restricted subclasses of ASHGs to see if the additional restrictions would elicit positive stability results [20]. Dimitrov et al. proved that FOHGs are guaranteed to have strictly core stable partitions that can be found in polynomial time [20]. EOHGs are guaranteed to contain core stable partitions, but finding them is NP-hard [20].

In 2014 Aziz et al. extended the core notions of ASHGs with fractional hedonic games (FHGs, see Definition 14) to find the average value of coalition members, contrasting them with $\mathcal{B}$ and $\mathcal{W}$ games, which rely on the computation of the best and worst (max and min) coalition members, and ASHGs, which compute the sum of values of coalition members. [3]. Aziz et al. proved that core stable partitions are not guaranteed to exist for FHGs and that when core stable partitions do exist, computing them is NP-hard and verifying that a partition is core stable is coNP-complete [3]. Nguyen et al. built on the foundations of FOHGs by introducing Altruistic Hedonic Games (AHGs, see Definition 17) in 2016, arguing that the happiness of your friends will affect your own happiness [45]. Nguyen et al. proved that Nash stable partitions always exist for AHGs and provided several hardness proofs for other stability notions [45]. Brânzei and Larson introduced social distance games (SDGs, see Definition 18) in 2011 to investigate group formation among agents that are socially *close* to each other [11]. Brânzei and Larson provided several optimality results as well as establish one property which guarantees the existence of core stable partitions and another property which precludes a coalition from being part of any core stable partition [11].

The work discussed thus far focused only on *who* agents want to be in a group with, but hedonic games can also incorporate *what* agents want to do once the group has been formed. For example, Darmann et al. introduced the Group Activity Selection Problem (GASPs, see Definition 22) in 2012 [18]. Darmann et al. provided several NP-completeness results for GASPs [18]. Spradling et al. introduced Roles and Teams Hedonic Games (RTHGs, see Definition 19) in 2013 to model team formation in the video game League of Legends [60]. Spradling et al. defined several stability notions for RTHGs and provide NP-hardness results for some, but also prove that individually stable partitions can be found in polynomial time [60]. Spradling and Goldsmith expanded on the ideas of RTHGs with the less rigidly defined Role Based Hedonic Games (RBHG, see Definition 20) in 2015, where they provided NP-completeness results for several notions of stability [59]. Siler introduced Tiered Coalition Formation Games (TCFGs, see Definition 21) in 2017. Siler provided NP-hardness results for TCFGs and also find the unusual result that core stability guarantees Nash stability in TCFGs.

Chapter 4 Super Altruistic Hedonic Games

Consider the process of choosing where to live. Let us consider the choice of *neighbors*, perhaps in a setting where students are choosing their dormitories/hostels. We can see the partitioning of students into living units (floors, buildings, etc.) as a hedonic game. It is clear that we value our friends' happiness with the living situation, as we will hear about it from them; our enemies' happiness could be assumed to also affect how they treat us. If we stopped there, we would be modeling evaluation as an *Altruistic Hedonic Game.* More generally we can also argue that our friends' friends' happiness will affect our friends', and thus indirectly, our own, and that this continues out friendship chains, with decreasing (or at least, non-increasing) effect as we increase the social distance from ourselves.

If we were building intranets, a node could evaluate the quality of the local network in terms of the bandwidth to reachable nodes. However, it would also need to take into account the quality of more distant connections, if it hopes to have its packets relayed. There are many other applications in which agents care not only about immediate connections, but also those farther away. We introduce a family of hedonic games that model such broad evaluations of coalitions: the Super Altruistic Hedonic Games.

4.1 Motivation

AHGs look at friends' preferences, but we believe that considering everyone in the coalition is better. Doing so makes it possible to model situations that are larger and more complex than what AHGs can model.

For example, we could use this to model an extension of the stable roommate problem to the problem of allocating all rooms on a dormitory floor. Since we are treating all residents of the floor as members of the same coalition, coalition members are not connected as closely as they are in the traditional roommate problem. However, coalition members still live in sufficiently close proximity that it is in their best interest to get along with other members.

4.2 Related Works

SAHGs are a natural extension and generalization of Altruistic Hedonic Games (AHGs, see Definition 17) wherein agents consider the preferences of other agents [45]. In AHGs, agents only consider the preferences of their friends. In SAHGS, agents consider the preferences of all agents in their coalition. In AHGs, friends are assigned a value of n where n is the number of agents. In SAHGs, friends are assigned a fixed non-negative value and enemies are assigned a fixed non-positive value, but there are no rules specifying what those values must be. Additionally, the preferences of all agents in a coalition are considered in SAHGs, often taking advantage of indirect relationships such as friends of friends to adjust the weight given to others' preferences. (Note that friendship is not transitive: a friend of a friend could be our enemy.)

While we expand the core notions of AHGs to consider all agents in a coalition instead of just friends, there are other ways to extend the concepts of AHGs. In 2020 Kerkmann and Rothe proposed the expansion of AHGs into the more general space of *coalition formation games*, by allowing an agent's altruistic behavior to extend beyond their coalition [37].

Social Distance Games (SDGs, see Definition 18) are a class of coalition formation games wherein an agent's utility is a measure of their closeness, or social distance, from the other members of their coalition [11]. SDGs have similarities to SAHGs, but we believe that SAHGs can better model realistic human interactions by combining the notion of social distance with the consideration of others' preferences proposed in AHGs.

As we later demonstrate with Proposition 1, SAHGs generalize Friends-oriented and Enemies-oriented Hedonic Games (FOHGs and EOHGs, see Definitions 15 and 16) [20]. In the former, agents seek coalitions that maximize the number of friends with a secondary goal of minimizing the number of enemies. In the latter, minimizing the number of enemies is the primary goal, while maximizing the number of friends becomes secondary. This finding makes sense because AHGs drew heavily on the core ideas of FOHGs [45].

Other research extends the concepts of FOHGs and EOHGs in different ways. Work by Ohta et al. investigates the impact that neutral agents have on these games, defining a neutral agent as one that is neither friend nor enemy [46]. Ohta et al. show that permitting neutral agents in EOHGs allows for games that have no core stable partition [46]. Core stable partitions are still guaranteed to exist in FOHGs with neutral agents; however, strict-core stable partitions are not [46].

Kerkmann et al. investigate games with friends, enemies, and neutrals where friends and enemies are ranked [38]. Kerkmann et al. prove that verification of possible and necessary stability is in P for individual rationality, Nash stability, individual stability, and contractual individual stability and is coNP-complete for core stability, strict core stability and strict popularity [38]. Kerkmann et al. find that proving the existence of possibly and necessarily stable partitions is in P for individual rationality and contractual individual stability, but is NP-complete for Nash stability and coNP-hard for strict popularity [38]. Deciding whether necessarily individually stable partitions exist is NP-complete, but deciding if possibly individually stable partitions exist is only known to be in NP [38].

Fichtenberger and Rey examine property testers with one-sided error for hedonic games with friends, enemies and neutrals and a bounded number of friend/enemy edges connected to each agent [24]. One-sided error, in this case, requires that a partition being tested for a certain property (e.g. Nash stability) will be always be correctly categorized if it has the desired property, but may be incorrectly categorized if it does not have the desired property. For example, a tester with one-sided error will correctly categorize all Nash stable partitions for a given game, but may incorrectly categorize some partitions that are not Nash stable.

Barrot et al. introduce FOHGs with unknowns, where *unknowns* are assigned an ϵ positive or negative value such that the number of unknowns will only decide an agent's preference between two coalitions when the number of friends and enemies is the same in both coalitions; this means that ϵ must be smaller than $1/n$. They prove that core stable and individual stable partitions may not exist when unknowns have ϵ positive values, while strictly core stable partitions (and thus, core stable partitions) are guaranteed to exist when unknowns have ϵ negative values [7]. Barrot et al. further prove that, when unknowns have ϵ positive values, deciding whether core stable partitions exist is NP^{NP}-complete and deciding whether individually stable paritions exist is NP-complete [7]. The findings of hedonic games with friends, enemies and either neutrals or unknowns do not readily translate to SAHGs, because SAHGs do not allow neutral or unknown agents. Neutral agents could be modeled as graph-based games by labeling appropriate edges as neutral, and unknowns could be modeled in a similar fashion, but SAHGs are focused on simple graph-based models, so the addition of neutral edges is beyond the scope of the work presented in this dissertation.

Work by Flammini et al. presents mechanisms to approximate optimal social welfare in FOHGs and EOHGs that are strategy-proof in regards to players' disclosure of who they view as friends and enemies [25]. Flammini et al. propose that FOHGs and EOHGs with neutrals are a natural extension of their work, since the assumption that all agents view each other as strictly friends or enemies is unrealistic [25].

Bermond et al. provide tighter bounds on the complexity of computing partitions composed of cliques with up to k agents in EOHGs for a fixed value of k [8]. The focus on cliques in the work by Bermond et al. makes sense, because all individually rational partitions in EOHGs are composed of cliques.

Brandt et al. investigate whether individually stable partitions can be reached by a sequence of deviations from some starting partition [10]. The techniques employed by Brandt et al. to prove or disprove the existence of such sequences are similar to the techniques we employ in our proof of Theorem 2.

There are graph-related hedonic games that depend on edge-weighted graphs. For instance, $\mathcal{B}$ and $\mathcal{W}$ games are a category of hedonic games in which an agent's utility is defined by the agents in their coalition that they rate as the best or the worst, respectively [15]. While these games fall into the category of hedonic games, we don't believe SAHGs can generalize $\mathcal{B}$ or $\mathcal{W}$ games. Similarly, we do not believe that either $\mathcal{B}$ or $\mathcal{W}$ games can generalize SAHGs. This is due to the differences between $\mathcal{B}$ and $\mathcal{W}$ games and SAHGS, such as the former two categories assuming each agent can assign a unique value to each other agent, while SAHGs restrict agents to placing others into one of two categories. Additionally, $\mathcal{B}$ and $\mathcal{W}$ games do not consider the preferences of others as SAHGs do.

The Coalition Structure Generation (CSG) problem presents a set of coalition formation games representable as graphs where each agent contributes a fixed value to their coalition [40]. There are several significant differences between the CSG problem and SAHGs, despite both falling under the broad scope of coalition formation games. One major difference is the way agent values are handled. In the CSG problem there can be as many values as there are agents and all agents agree on the values assigned to each agent, whereas SAHGs only allow one of two values to be assigned to each agent and agents aren't guaranteed to agree on those values. Additionally, the CSG problem assumes that utility is transferable, so all members of a coalition earn the same utility. SAHGs do not have transferable utility, so not all agents in a given coalition are guaranteed to have the same utility. As a result, there is no clear means to translate between the CSG problem and SAHGs.

4.3 Preliminaries

Super Altruistic Hedonic Games[1] **(SAHGs)** extend the core principal of Altruistic Hedonic Games (see Definition 17) so agents consider the preferences of all agents in their coalition. Agents weigh their consideration of each other's preferences according to some polynomially computable value.

Definition 38. *Let parameters (a, g, M, L) be non-negative weights where a and g represent the weights associated with friends and enemies, respectively, while M and L represent the weights associated with personal preference and the average of friends' preferences. Next, let $D(i, j)$ be a polynomial-time computable function that is non-increasing with the graph distance between i and j. Let the number of other agents in coalition C_i be $h_i = |C_i \setminus \{i\}|$. For each agent $i \in N$, let that agent's base preference be $b_i = a|C_i \cap F_i| - g|C_i \cap E_i|$, and let their utility be*

$$u_i = Mb_i + L \sum_{j \in C_i \setminus \{i\}} \frac{D(i, j) \cdot b_j}{h_i}.$$

If $C_i = \{i\}$ then the sum is set to 0. The default definition of D is the inverse graph distance function: for any pair of agents $i, j \in N : i \neq j$, let d_{ij} be the shortest path distance between them, then let $D(i, j) = 1/d_{ij}$. The ***total utility*** *of a partition π is given by $U_T = \sum_{i \in N} u_i$.*

We next show how SAHGs generalize friend-oriented and enemy-oriented hedonic games (see Definitions 15 and 16) as well as one type of AHGs.

Observation 1. *SAHGs generalize several graph-based hedonic games.*

- *A Friends-oriented Hedonic Game is a SAHG with parameters $(a, g, M, L) = (n, 1, 1, 0)$, and an Enemies-oriented Hedonic Game is a SAHG with parameters*

[1] We considered calling them "Super Kinda Altruistic ex-Hedonic Fun Games," even though the sound of it was really quite atrocious.

$(1, n, 1, 0)$. (Because $L = 0$, it does not matter how we define D.) Thus, all hardness results for FOHGs and EOHGs are inherited by SAHGs.

- *SAHGs also model Altruistic Hedonic Games under the selfish-first criterion ($(a, g, M, L) = (n, 1, n^5, 1)$ and $D(i, j) = 1$ if $j \in F_i$ and $D(i, j) = 0$ if $j \notin F_i$).*
- *If $D \equiv 1$ and $(a, g, M, L) = (n, 1, 1, 1)$ then we capture the notion of a friend-oriented hedonic game on the transitive closure of the friendship graph.*

4.4 Properties of Partitions

In this section we look at cliques in SAHGs. We also consider the existence of stable partitions.

Proposition 1. *If a coalition comprises a single clique, C, then individual utilities are given by a linear function of the number of agents and coalition utility is defined by a geometric function of the number of agents.*

Proof. We first recall that the base preference of each agent $i \in N$ is given by $b_i = a|C_i \cap F_i| - g|C_i \cap E_i|$ where C_i is the coalition to which i belongs, and that $h_i = |C_i \cap F_i| + |C_i \cap E_i|$ defines the number of agents in $C_i \setminus \{i\}$. Next recall that each agent $i \in N$ has utility given by

$$u_i = Mb_i + L \sum_{j \in C_i \setminus \{i\}} \frac{D(i, j) \cdot b_j}{h_i}.$$

The total utility of a partition is defined by $U_T = \sum_{i \in N} u_i$.

Now we define the total utility of a coalition as $U_C = \sum_{i \in C} u_i$. Because C is a clique, we know that $\forall i, j \in C, i \neq j$ $D(i, j) = 1$. We also know that all $i \in C$ have $h_i = |C| - 1$ and $b_i = a(|C| - 1)$. We use this to calculate

$$u_i = M \cdot a(|C| - 1) + L \sum_{j \in C_i \setminus \{i\}} \frac{a(|C| - 1)}{1(|C| - 1)},$$

which simplifies to $u_i = (M + L) \cdot a(|C| - 1)$.

The total utility of the coalition is $U_C = \sum_{i \in C} u_i$, which simplifies to $U_C = (M + L)(a(|C|^2 - |C|))$. Thus, we have demonstrated that, given a coalition C comprised of a single clique, the individual utility is a linear function of $|C|$ and the coalition utility is a geometric function of $|C|$. □

Next, we analyze the impact of cliques in SAHGs.

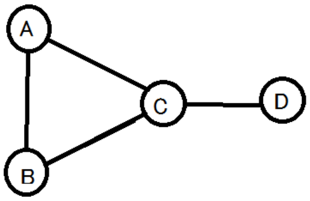
Proposition 2. *Different partitions of a set of agents into cliques may have different utilities.*

Proof. Consider that coalition utility scales geometrically with the number of agents if the coalition is a clique. Unless the clique-coalitions are all of equivalent size, then the net utility will be different.

We can also prove by contradiction with a game based on Figure 4.1 with parameters $(a, g, M, L) = (1, 1, 1, 1)$. For each $i \in N$ we have:

- $b_i = |C_i \cap F_i| - |C_i \cap E_i|$
- $u_i = b_i + \sum_{j \in C_i \setminus \{i\}} \frac{D(i,j) \cdot b_j}{h_i}$.

Figure 4.1: Unequal Cliques



Consider two partitions:

$$\pi_1 = \{\{A, B, C\}, \{D\}\} \text{ and } \pi_2 = \{\{A, B\}, \{C, D\}\}.$$

In π_1, we have $b_A = b_B = b_C = 2$ and $b_D = 0$. We also have $u_A = u_B = u_C = 4$ and $u_D = 0$ and $U_T(\pi_1) = 12$. In π_2, we have $b_A = b_B = b_C = b_D = 1$ and $u_A = u_B = u_C = u_D = 2$. Thus, $U_T(\pi_2) = 8$.

Since we have two partitions into cliques with different total utility values, we can conclude that partitioning agents into cliques does not ensure a consistent total utility. However, given partitions π_3 and π_4 dividing agents into equal numbers of cliques of each size, the total utilities of the two partitions will be the same. □

Next we demonstrate that it is possible to construct stable partitions for SAHGs under any notion of stability.

Proposition 3. *For all parameter values, for all stability notions considered in this paper, there exist SAHGs with stable partitions.*

Proof. Let G be the SAHG with structure given by a graph with n nodes and no edges with parameters (a, g, M, L). For the partition of singletons, each agent i has utility $u_i = 0$. Since there are no edges in the graph, no agent would benefit from forming a coalition with any other agent or set of agents, so the partition of singletons is stable. □

In Theorem 1 we prove that strictly core stable partitions are not guaranteed to exist in SAHGs.

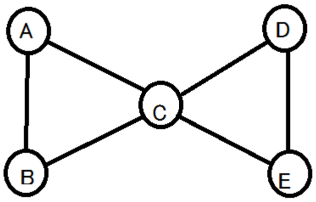
Theorem 1. *Not all SAHGs have strictly core stable partitions.*

Proof. Consider a game based on Figure 4.2 with parameters $(a, g, M, L) = (1, 1, 1, 1)$. For each agent $i \in N$, we have:

- $b_i = |C_i \cap F_i| - |C_i \cap E_i|$
- $u_i = b_i + \sum_{j \in C_i \setminus i} \frac{D(i,j) \cdot b_j}{h_i}$.

This game contains two equal-sized cliques connected by a single intermediate agent, C. The grand coalition is weakly blocked by $\{A, B, C\}$ and $\{C, D, E\}$. If one of these weakly blocking coalitions splits off from the grand coalition, we either have

Figure 4.2: Graphical Representation of Friends



$$\pi_1 = \{A, B, C\}, \{D, E\} \text{ or } \pi_2 = \{A, B\}, \{C, D, E\}.$$

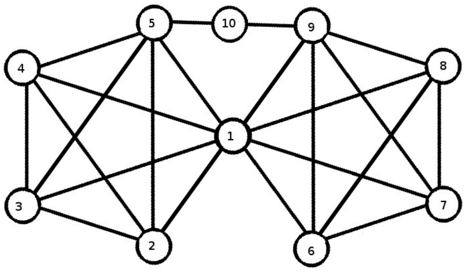
π_1 is weakly blocked by $\{C, D, E\}$ and π_2 is weakly blocked by $\{A, B, C\}$.

The utility of A and B is maximized in $\{A, B, C\}$, while $\{C, D, E\}$ maximizes the utility of C and D. The utility of agent C is maximized by the grand coalition and by $\{A, B, C\}$ and $\{C, D, E\}$. As such, all possible partitions are weakly blocked by $\{A, B, C\}$, $\{C, D, E\}$, or both. Thus there is no strictly core stable partition. □

Next we prove that Nash stable partitions don't always exist for SAHGs, even when the SAHGs are based on undirected graphs.

Theorem 2. *Not all SAHGs based on undirected graphs have Nash stable partitions.*

Figure 4.3: Game with no *Nash stable* partition



Proof. Let G be the SAHG with structure given in Figure 4.3, and weight parameters $(a, g, M, L) = (1, 1, 1, 3)$. This gives us:

- $b_i = |C_i \cap F_i| - |C_i \cap E_i|$
- $u_i = b_i + 3 \sum_{j \in C_i \setminus i} \frac{D_(i,j) \cdot b_j}{h_i}$.

This game has two equal-sized cliques which are connected to each other through two intermediate agents. The first connecting agent, agent 1, is connected to all agents in both cliques. The second connecting agent, agent 10, is connected to a single agent in each clique and is not connected to agent 1. The first clique is composed of agents 1–5 and the second of agents 1, 6–9.

Because the only member common to both cliques is agent 1, it is reasonable to expect that no stable coalition containing one clique will contain any members from the other, except for agent 1. If members from two cliques form into coalitions which do not include agents 1 and 10, then these two remaining agents would prefer to remain as singletons rather than forming a two-person coalition with each other. In this case, the utility of an agent in one of the two clique coalitions is 12, while the utility of agents 1 and 10 are zero since they are singletons. This describes partition $\pi_1 = \{\{1\}, \{2, 3, 4, 5\}, \{6, 7, 8, 9\}, \{10\}\}$ with total utility $U_T = 96$.

The partition π_1 is unstable, because agent 1 can improve their utility by joining one of the two clique coalitions. Since agent 1 is connected to all agents in both cliques, its joining either coalition will increase the size of the clique by 1, increasing the utility of all agents in the coalition from 12 to 16. Agent 1 is indifferent between the two cliques. This presents two possible partitions
$\pi_2 = \{\{1, 2, 3, 4, 5\}, \{6, 7, 8, 9\}, \{10\}\}$ and
$\pi_3 = \{\{2, 3, 4, 5\}, \{1, 6, 7, 8, 9\}, \{10\}\}$,
each of which has total utility $U_T = 128$.

Both π_2 and π_3 are also unstable because agent 10 can also improve its own utility by joining a coalition. If agent 10 chooses to join the coalition that agent 1 did not, it derives utility $u_{10} = 3.25$, while it derives utility $u_{10} = 3.6$ if it joins the same coalition as agent 1. Thus, agent 10 prefers to join whichever coalition agent 1 joined, which results in either $\pi_4 = \{\{1, 2, 3, 4, 5, 10\}, \{6, 7, 8, 9\}\}$ or $\pi_5 = \{\{2, 3, 4, 5\}, \{1, 6, 7, 8, 9, 10\}\}$. The total utility of this new partition is $U_T = 104$.

Still, π_4 and π_5 are unstable because agent 1 can improve its utility by leaving the current coalition to join the other clique, thereby restoring its utility to 16. This gives either $\pi_6 = \{\{2,3,4,5,10\},\{1,6,7,8,9\}\}$ or $\pi_7 = \{\{1,2,3,4,5\},\{6,7,8,9,10\}\}$. In these partitions, the total utility is $U_T = 110.5$. However, π_6 and π_7 are unstable since agent 10 can improve its utility by following agent 1, which creates a cycle of four partitions, none of which are Nash stable. Thus we conclude that there is no Nash stable partition for this game, and, by extension, that not all SAHGs are guaranteed to have Nash stable partitions. □

Theorem 2 contrasts with existing results for AHGs, which always have Nash stable partitions [45], and FOHGs, which always have strictly core stable partitions [20].

Notice that the game in the proof of Theorem 2 has core stable partitions: $\{\{1,2,3,4,5\},\{6,7,8,9\},\{10\}\}$ and $\{\{2,3,4,5\},\{1,6,7,8,9\},\{10\}\}$. The 5-member cliques weakly block the opposing partition, but there are no coalitions that block either partition. Additionally, agent 10 would not be accepted in either coalition, since its presence decreases the utility of every other member in the coalition.

Observation 2 follows from previous work by Ballester (2004) proving that all hedonic games have contractually individually stable partitions.

Observation 2. *[5] Contractually individually stable partitions are guaranteed to exist, for all hedonic games.*

In section 4.5, we cover computational complexity results for SAHGs.

4.5 Computational Complexity

Our first result addresses the complexity of computing the sum of the utility each agent derives from a partition.

Proposition 4. *Computing the utility of a partition for a SAHG is in P.*

Proof. Consider a partition π of some game G. The steps to evaluate the partition are:

1. $\forall i \in N$ and $\forall j \in \pi(i)$ compute $D(i,j)$
2. $\forall i \in N$ compute h_i and b_i
3. $\forall i \in N$ compute u_i
4. compute $U_T(\pi)$.

We assume that intermediate values are computed once and stored.

In the default case where $D(i,j)$ is the graph distance between i and j, we can use the Floyd-Warshall algorithm to compute this distance for all $(i,j) \in N \times N$ in time $\mathcal{O}(n^3)$ [17], otherwise, it is $\mathcal{O}(n^2)t(n)$, where $t(n)$ is the time needed to compute any $D(i,j)$ for a SAHG of size n. We compute h_i and b_i in time $\mathcal{O}(n^2)$ by checking each entry in $\pi(i)$ against the lists F_i and E_i. Computing h_i and b_i for all $i \in N$ requires time $\mathcal{O}(n^3)$. Calculating u_i requires time $\Theta(|\pi(i)|) < \mathcal{O}(n)$. So the time required to compute u_i for all $i \in N$ is $\mathcal{O}(n^2)$. $U_T(\pi)$ can then be computed in time $\mathcal{O}(n)$. The overall time required to evaluate a partition is $\mathcal{O}(n^3)$ when $t(n) \in \mathcal{O}(n)$ and $\mathcal{O}(n^2)t(n)$ otherwise. Thus a partition of a Super Altruistic Hedonic Game can be evaluated in polynomial time. □

Next we establish that Nash stability can be verified in polynomial time, which proves that checking whether Nash stable partitions exist is in NP.

Proposition 5. *Deciding whether a partition is Nash stable is in time $\mathcal{O}(n^2 \cdot e(n))$, where $e(n)$ is the time needed to evaluate the utility of a coalition.*

Proof. Consider a partition π of some game G. To determine if π is Nash stable, $\forall i \in N$ and $\forall C \in \pi : C \neq \pi(i)$ we compare $u_i(\pi(i))$ with $u_i(C \cup \{i\})$. If $\nexists(i,C)$ such that $u_i(C \cup \{i\}) > u_i(\pi(i))$, then π is Nash stable.

There are at most n coalitions in π in the case of the partition of singletons, and for each $C \in \pi$ n utility values must be computed. At most n^2 utility values must be computed to determine if π is Nash stable. Determining if a partition π is Nash stable requires time $\mathcal{O}(n^2 \cdot e(n))$ where $e(n)$ is the time needed to compute the utility of a coalition. □

Table 4.1: Known Complexity Results

Nash Stable Existence		
AHG	Always exist	[45]
Individually Stable Existence		
AHG	Always exist	[45]
Contractually Individually Stable Existence		
AHG	Always exist	[45]
Wonderfully Stable Existence		
EOHG	DP-hard	[50, 51]
Core Stable Existence		
EOHG	Always exist	[20]
FOHG	Always exist	[20]
AHG (selfish-first)	Always exist	[45]
Strict Core Stable Existence		
EOHG	DP-hard	[50, 51]
FOHG	Always exist	[50, 51]
AHG (selfish-first)	Always exist	[45]
Core Stable Verification		
EOHG	coNP-complete*	[62]
Strict Core Stable Verification		
EOHG	coNP-complete*	[62]
Strict Core Stable Computation		
FOHG	P	[20]
Strictly Popular Verification		
AHG	coNP-complete	[45]
Strictly Popular Existence		
AHG (selfish-first)	coNP-hard	[45]
* Corollary of their result for additive games.		

We have previously demonstrated that FOHGs, EOHGs, and selfish-first AHGs are generalized by SAHGs. As a result, SAHGs inherit the complexity results of these games as lower bounds on the hardness of general case SAHGs. Known complexity results for these games are outlined in Table 4.1.

Corollary 1.

- *Determining if wonderfully stable partitions exist is DP-hard [50, 51].*
- *Determining if strictly core stable partitions exist is DP-hard [50, 51].*
- *Verifying a partition is (strictly) core stable is coNP-hard [68].*
- *Determining if strictly popular partitions exist in SAHGS is coNP-hard [45].*
- *Verifying a partition is strictly popular is coNP-hard [45].*

In the last section, we review our contributions and outline several open questions related to our work on SAHGs.

4.6 Conclusions and Open Questions

We introduce SAHGs as an extension of the ideas behind AHGs, introduced by Nguyen et al. [45]. We show that SAHGs generalize several graph-based hedonic games: FHGs, EHGs, and AHGs under the selfish-first criterion in proposition 1. What distinguishes SAHGs from the games they generalize is the consideration of the preferences of all other agents in one's coalition. This difference allows SAHGs to better model partitioning problems with a larger scope than roommate assignment problems.

We examine several properties of SAHGs in Propositions 1–3. We prove that stable partitions may not exist, even when friendship relations are symmetric in Theorems 1 and 2. We show that the total utility of a partition can be computed in polynomial time in Proposition 4. Proposition 5 demonstrates how Nash stability can be verified in polynomial time. Corollary 1 clarifies lower bounds on complexity that SAHGs inherit from games they generalize.

There are many questions relating to SAHGs that remain open. How large can a general case SAHG be before deterministic solvers become impractical? What sorts of heuristics can solve SAHG instances efficiently? Our experimental work on Anchored Team Formation Games provides a couple of potential heuristics we could investigate (see Chapter 5). Additionally, Miles [42] crafted a useful web program to simulate a variety of hedonic games, which provided insight for our work on Anchored Team Formation Games, and may be another useful resource to help construct deterministic and heuristic solvers for SAHGs.

Are SAHGs guaranteed to have internally stable, core stable, or individually stable partitions? What are the necessary and sufficient conditions to guarantee the existence of stable partitions in SAHGs for stability notions? We are particularly interested in stability notions where general-case SAHGs are not guaranteed to have stable partitions, such as Nash stability.

Do SAHGs generalize any social network problems? As with FOHGs and EOHGs, social network problems generalized by SAHGs will provide lower complexity bounds on a number of questions of interest for SAHGs. Do any well-studied social network problems generalize SAHGs? If so, this would place an upper bound on the complexity of many SAHG problems.

Future SAHGs research could build off extensions to FOHGs and EOHGs. What impact would neutral agents similar to those proposed by Ohta et al. for FOHGs have on SAHGs[46]? How would unknowns, as proposed by Barrot et al., impact SAHGs [7]? Flammini et al. recognize that agents are not always truthful when reporting their preferences [25]. Can strategyproof mechanisms similar to those proposed by Flammini et al. be constructed for SAHGs?

Chapter 5 Anchored Team Formation Games

Consider tabletop role playing games (TRPG) such as Dungeons and Dragons. In order to play a TRPG, a group must have players and a *game manager (GM)*; the former working through challenges set up by the latter. The expectation of enjoying a particular group to play such a game is based on expectations that the GM will set up a good story line, with sufficient challenges and rewards, and that the fellow travelers will offer good measures of cooperation and competition. We introduce *anchored team formation games (ATFGs)* to investigate the formation of groups with a leader, or anchor, using the formation of Tabletop Role Playing Game (TRPG) groups as an example application. While we refer to the gaming application throughout the paper, the anchor could be a team lead in a programming or engineering team, in business, class project groups, or outdoor adventuring. Whatever the application, the driving question is, how do we divide individuals into groups that will get the job (or game) done, and choose leaders for the groups, in a way that is consistent with the individuals' preferences?

We are interested in determining whether *stable partitions* exist for a given ATFG, and how to find them. Given our proposed use case, we believe Nash stability is the most relevant stability notion in the literature. However, the problem of finding a Nash stable partition is NP-hard. We present experimental results with three effective heuristics to check the existence of Nash stable partitions in ATFGs. Our first algorithm starts by selecting anchors around which to build coalitions, then divide players in a round robin fashion consisting of several rounds where each coalition chooses one player to add. Our second algorithm is a local search implementation that attempts to minimize the number of blocking players, those who can improve their utility by unilaterally deviating from their assignment. Our third algorithm daisy-chains the previous two, using the output of the round-robin algorithm as a starting place for a local search.

Experiments on 10–12-agent instances show that all three heuristics perform quite well. On larger instances, however, we see that daisy-chaining is a significant improvement over either individual algorithm. These experiments indicate that good partitions can be found. In the next section, we provide a more detailed motivation for thise work. We then formally define ATFGs and present the algorithms used, the experimental set-up, and the results.

5.1 Motivation

Consider about a university club focused on playing TRPGs. A key challenge for the club is to divide up the club members so everyone can participate in a game while also keeping everyone happy. They may be able to quickly determine approximately how many groups should be formed, but what may be less clear is *who* should be in which group. Not everyone will agree on who makes a good GM. There are differences in opinion over what sorts of stories and challenges the best games should feature, which can have a big impact on the types of games someone creates when they are a GM or the types of games someone enjoys playing. This means that Azar may view Dawa as an excellent GM, while Cleo may find Dawa intolerable as a GM based on the types of games Dawa tends to make and the types of games that Azar and Cleo like to play. People also have preferences over who they play with. Some people may prefer to play with people who are good at engaging with the game's story, while others may prefer to play with people who are skilled at building their characters with excellent combat abilities. As with GMs, not everyone will agree who the best people to play with are.

Alternatively, consider an academic conference such as AAAI with breaks separating the various sessions throughout each day. During the breaks, most attendees will engage in casual conversation with each other. Oftentimes, attendees will tend to move around for several minutes at the beginning of the break before settling in with one group they'll spend the majority of the break talking with. Many attendees are university professors, post-docs, or students. Professors and post-docs will often have some degree of recognition in the research community, with professors typically having more. Students, in contrast, will typically be largely unknown. We propose the professors, and post-docs to a lesser degree, will often serve as anchor points for conversation groups during these breaks. As with TRPGs, not everyone will agree on who the best researchers to speak with are; for example, many people may prefer to spend their time talking with other researchers specializing in topics similar to their own.

5.2 Related Works

Banerjee et al. and Bogomolnaia and Jackson introduced additively separable hedonic games (ASHGs), where players assign utility values to each other, similar to the rankings in Gale and Shapley's roommates problem, but there are no restrictions on coalition size [6, 9, 27]. ASHGs are of particular interest to our work on ATFGs, because they are generalized by ATFGs; there are many other hedonic games inspired by ASHGs [3, 20, 39, 45, 55]. There are many hardness results for ASHGs and their variants, which establish some baselines for our work [2, 49, 51, 54, 68].

Our work on ATFGs expands the body of hedonic games inspired by gaming. Hedonic games inspired by gaming include *Tiered Coalition Formation Games (TCFGs)* inspired by *Pokémon*, *Roles and Teams Hedonic Games (RTHGs)* inspired by League of Legends, and *Role Based Hedonic Games (RBHGs)*, which generalize RTHGs [56, 59, 60].

We expand the body of research applying heuristics to hedonic games with our work on ATFGs. Spradling et al. gives experimental results for individual stability in RTHGs using a local search heuristic and another greedy heuristic that models agents as voters then holds a series of elections to assign agents to coalitions [60]. Spradling et al. compare their heuristic results to MaxMin and MaxSum brute force algorithms [60]. Later work by Spradling uses greedy voting and greedy local search heuristics to find optimal partitions in RBHGs [61]. We use a local search heuristic (see Algorithm 4), a round robin heuristic (see Algorithm 2), and a combination of the two to find Nash stable partitions for ATFGs. We employ a complete algorithm based on depth first search to find Nash stable partitions in ATFGs as a baseline to which to compare our heuristics.

Keinänen and Keinänen develop local search heuristics for core stability verification and social welfare optimization in hedonic games [33, 34, 36]. We conjecture that Keinänen does not extend this to local search algorithms to construct stable partitions, as we do, in part because the fitness function we use is unusual, and is specific to Nash stability. However, Keinänen presents an EXPTIME breadth first search algorithm to compute all Nash stable partition in ASHGs [35], which they were able to run on (many) instances of size 10. Our complete algorithm has successfully run 20-player instances, despite ATFGs being more complicated the ASHGs. We hypothesize the improvements in CPU architecture and differences in the number of instances tested explain why Keinänen only examined instances of size 10, while we were able to process instances of size 20; however, their runtimes are based on number of partitions checked, where ours are based on CPU and wall clock time, so they cannot be directly compared.

Waxman et al. utilize simulated annealing and leximin heuristics to optimize egalitarian welfare in ASHGs where a fixed number of coalitions must be formed [67]. Taywade et al. give experimental results with three heuristics to optimize social welfare in decentralized matching [65], which is significantly different from our centralized setting, as well as being focused on optimality rather than stability. Collins et al. use a Monte Carlo implementation to generate core stable partitions for a large number of hedonic games with strict preferences and up to 13 agents [16].

Gairing and Savani examine how many steps are required for local search to converge on local optima for symmetric ASHGs for Nash stability, individual stability, and contractual individual stability [26]. Gairing and Savani characterize their results according to the polynomial local search (PLS) complexity class [26]. Informally, a problem is in PLS when a polynomial time algorithm can construct an initial candidate solution, compute the cost or utility of a candidate solution, and find a neighboring solution that improves upon a candidate solution if one exists [32]. Gairing and Savani show that for ASHGs finding Nash stable or individually stable partitions is PLS-complete, while finding contractually individually stable partitions can be found in polynomial time with local search [26]. Since ATFGs generalize ASHGs, as shown in Proposition 8, we conclude that finding Nash stable partitions in ATFGs is a PLS-complete problem.

5.3 Preliminaries

We now introduce our model of team formation for gaming. A standard play group in many TRPGs consists of four players and a fifth person, typically referred to either as the game manager (GM) or dungeon manager (DM), we shall hereafter use the term GM. The GM provides the game's setting and leads the players through the story. The number of players varies, but there is only one GM. A GM's performance can easily make or break a game. A good GM keeps players engaged, while a bad GM can make for a poor experience.

We introduce Anchored Team Formation Games to model the formation of such groups. One of the biggest differences between our setting and most others is that we assume that players may have opinions not only on other individuals, but also on *pairs* of players, either positively[1] or negatively.

Anchored Team Formation Games are a class of cooperative coalition formation games in which each coalition must contain an anchor, or leading player, and in which upper and lower bounds limit the permissible sizes of coalitions.

Definition 39. *An* ***Anchored Team Formation Game (ATFG)*** *is a tuple $\langle N, V, P, D, c_u, c_l \rangle$, where N is a set of n players, and V, P, and D define weight values such that each $i \in N$, there are vectors v_i, p_i, and d_i of lengths n, n^2, and n respectively. The $v_i[j]$ is the utility i gets for being in a coalition with player $j \in N$; $p_i[j,k]$ is the utility i gets for being in a coalition with the pair of players (j,k); $d_i[j]$ is the utility i gets if j is the anchor (GM) for i's coalition. Values contained in each v_i, p_i, and d_i are assumed to either be integers or unknowns.*

The values c_u and c_l define upper and lower bounds on coalition size. Any valid coalition $C \subseteq N$ must satisfy $c_u \geq |C| \geq c_l$. Further, all valid coalitions $C \subseteq N$ must contain a designated player $g(C)$ who serves as the anchor, or coalition leader. In order for a partition γ to be valid, it must consist solely of valid disjoint coalitions.

The utility $u_i(\gamma)$ a player $i \in N$ derives from a valid partition γ is defined as follows:

$$d_i[g(C_i)] + \sum_{j \in C_i \setminus \{i\}} v_i[j] + \sum_{\{j,l\} \subset C_i \setminus \{i\}} p_i[\{j,l\}].$$

Evaluating the utility of a given coalition is a relatively fast operation, as shown in Proposition 6.

Proposition 6. *Evaluating $u_i(C)$ takes time $O(n^2)$.*

Note that evaluating $u_i(C)$ takes time $O(m^2) \subseteq O(n^2)$, where m is an upper bound on $|C|$, due to the p_i summation. Thus, evaluating the total utility of a

[1] Consider James and Elyse Willems (`https://www.youtube.com/channel/UCboMX_UNgaPBsUOIgasn3-Q`): most players would yield a higher utility from having both James and Elyse in a TRPG campaign as opposed to either, singly.

coalition is $O(m^2)$ (and thus $O(n^2)$) and of an entire partition is $O(n^3)$, or more precisely, $O(nm^2)$.

Stability

An assignment, formally a partition, of players to tables is only useful if the players consent to the assignment. We presume that, if they are aware of an assignment that they perceive as better than the one on offer, they will attempt to move to the better assignment. A partition is *stable* when it will not be disrupted by players rejecting their assigned coalitions and moving to other coalitions. There are many sets of constraints placed on such disruptions, such as the number of players that can move simultaneously; whether all moving players must see an increase in utility; whether players left behind by movers must see their utility increase, or whether players being joined by movers must see their utility improve. We focus on Nash stability (see Definition 24, which was adapted to hedonic games by Bogomolnaia and Jackson [9].

Note that there are some trivial cases of stable partitions, which we ignore. For instance, if coalitions must have size ≥ 3, then the partition of all singletons is Nash stable, since no pair is a valid coalition. We assume that all partitions contain at least one valid coalition.

Our local search heuristic (Algorithm 4) defines fitness by the number of blocking players, players who want to unilaterally leave their current coalition to join another or to become a singleton. This measure, the *Degree of Instability (DoI)*, was suggested by Roth and Xing for matching in decentralized markets [53]. A more robust notion of instability for pair matching was proposed by Eriksson and Häggström [23]. We use the simpler count of blocking players as it is more appropriate for Nash stability.

5.4 Computational Complexity

We now discuss computational complexity results for ATFGs, starting with a proof that Nash stability verification is in P.

Proposition 7. *Nash stability verification for ATFGs runs in polynomial time.*

Proof. Consider some ATFG (N, V, P, D, c_u, c_l) with partition $\gamma = \{C_1, ..., C_k\}$. For each $i \in N$, recall that $u_i(\gamma)$ is defined by:

$$d_i[g(C_i)] + \sum_{j \in C_i \setminus \{i\}} v_i[j] + \sum_{\{j,l\} \subset C_i \setminus \{i\}} p_i[\{j, l\}].$$

For each i and each $C' \in \gamma$, we compute $u_i(C' \cup \{i\})$ in time $O(n^2)$, by Observation 6. If $\exists i : u_i(C' \cup \{i\}) > u_i(\gamma)$, the partition is not Nash stable. This computation is order $O(n^4)$, as there are n players and $O(n)$ coalitions. □

Proposition 8 highlights the relationship between ATFGs and ASHGs.

Proposition 8. *ATFGs generalize ASHGs.*

Proof. We can convert any ASHG into an ATFG as follows:

1. Taking the values each player assigns to each other player as-is.
2. For all $p_i \in P$ set all values $p \in p_i$ to 0.
3. For all $d_i \in D$ set all values $d \in d_i$ to 0.
4. Set $c_u = n$.
5. Set $c_l = 0$.

Following this conversion, the anchor becomes irrelevant, as they have no impact on utility. The bounds set on coalitions are such that they impose no limits whatsoever, as any coalition size obtainable from a set of n players is permitted. □

Sung and Dimitrov prove that determining if a Nash stable partition exists is NP-complete for ASHGs [63]. Their proof shows that this holds even when there is a bound of 7 on the size of coalitions. Because ATFGs generalize ASHGs, we know that the Nash stability existence problem is NP-hard for ATFGs. Proposition 7 shows that Nash stability verification is in P, so Nash stability existence is in NP. Thus Nash stability existence is NP-complete.

An ATFG constructed from an ASHG does not make use of coalition size restrictions or utility values assigned to anchors or pairs of players. Our focus is on settings where coalition sizes are restricted and a coalition's anchor is an important part of each player's utility.

5.5 Algorithms

We present a complete algorithm used as a comparison basis for our heuristics. For the heuristic algorithms, the first step is to choose the GMs, and then to assign players to coalitions. We present the GM selection function, and then describe how agents' utilities are represented. We then describe round-robin and local search algorithms. In all algorithms, W represents an arbitrary ATFG instance (N, V, P, D, c_u, c_l).

Complete search

As a baseline for our heuristics, we have a branch and bound algorithm that finds all Nash stable partitions of a given instance. We refer to this as the *complete algorithm.* Development of the algorithm started with a relatively pure implementation of a depth-first search tree, where nodes at each level consist of coalitions that can be constructed from as-yet unassigned agents. To eliminate redundant branches, each node in the search tree contains the lowest-numbered available agent. In order to improve the algorithm's runtime, we've implemented a series of optimizations that ultimately resulted in the branch and bound algorithm we have now.

Originally, we stored a list of possible partitions as the tree structure was explored, then later used multiprocessing tools to verify the stability of the partitions in parallel either after the tree was fully explored or after a certain number of partitions were held in storage, based roughly on available RAM. This was based on the assumption that the majority of the computation time for the algorithm was spent on verifying stability. Testing revealed that this was incorrect, so we adjusted the implementation to use multiprocessing for tree exploration, then allowed the individual processes to verify the partitions they store. Before making this adjustment, the wall clock runtime for instances with 15 agents routinely hit a 20 minute time limit imposed during development. Afterwards, the wall clock runtime was reduced to an average of 65 seconds and a maximum of 263 seconds (or 4 minutes and 23 seconds).

We later restricted the algorithm so that singletons are only considered if 1 is explicitly defined as a valid coalition size. This reduced the wall clock runtime average from 65 to 28 seconds and the maximum from 263 to 120 seconds. Next we imposed a restriction to immediately prune any branches that introduce instability to the existing coalition set. This reduced the wall cock runtime average to 6 seconds and the maximum to 17 seconds.

We ran the complete algorithm on 20 instances with 15 agents and 25 instances with 20 agents. The 15 agent instances averaged 22.7 seconds of wall clock runtime with a standard error of 3.8 seconds on our virtual machine, which was a Linux machine with a CPU of Intel Xeon 2 cores at 2.1GHz, and 4GB of RAM. The 20 agent instances averaged 6489.5 seconds of runtime with a standard error of 1270.4 seconds on the same machine. We have successfully run a single 24 agent instance through the complete solver; the computation took 281621.9 seconds (78.2 hours) to complete.

In addition to our complete algorithm, we constructed three heuristic solvers for Nash stability in ATFGs. The first step in each of these heuristics is to decide which agents will be GMs.

We use a **GM Selection** algorithm (Algorithm 1) to select agents who are willing to be GMs. The algorithm checks the value each agent derives from being a GM, then places all agents that derive positive value from being a GM into a list of potential GMs. Next, a number of GMs, $g_s \leftarrow s \in [\lceil \frac{|N|}{c_u} \rceil, \lfloor \frac{|N|}{c_l} \rfloor]$, are selected at random from that list. As GMs are selected, they are moved to the final GM list. The list of remaining agents becomes the list of players. Note that the GM Selection algorithm runs in time $O(n)$, assuming that the random number generation takes constant time.

Next, we introduce introduce several algorithms to assign players to tables and their corresponding GMs.

Algorithm 1 GM Selection

```
Pick GMs for Algorithms 2 and 4
  Input: ATFG $W$
  Output: GM set $G$, player set $A^*$
  $g_s \leftarrow s \in [\lceil \frac{|N|}{c_u} \rceil, \lfloor \frac{|N|}{c_l} \rfloor]$ %number of GMs
  $G' \leftarrow \{i \in N : d_i[i] > 0\}$
  $G \leftarrow g_s$ unique random options from $G'$
  $A^* \leftarrow \{i \in N \setminus G\}$ non-GM players
  return: $G, A^*$
```

Heuristics

Our first heuristic uses a greedy round robin selection process. The **Utilitarian Round Robin (URR)** algorithm (Algorithm 2) first chooses GMs (Algorithm 2). The GMs then take turns choosing players whose addition to the table maximizes total utility for the table, this utility value is computed by Algorithm 3.

The URR algorithm maintains a matrix of valuations of players by GMs, representing the value of adding a player to that GM's table. Choosing a best addition is $O(n)$ for each of the $O(n)$ GMs. After each round of additions, the $O(n^2)$ matrix is updated, with each update taking $O(n^2)$, by Proposition 6. There are $O(n)$ rounds, so an iteration runs in time $O(n^5)$. Early on, we also developed a selfish round

Algorithm 2 Utilitarian Round Robin

```
**Input:** ATFG $W$, GM set $G$, player set $A^*$
**Output:** $\gamma$, Potentially Nash stable partition
$\gamma \leftarrow \emptyset$
**for** $g \in G$ **do**
  $\gamma_g \leftarrow \{g\}$
  $\gamma \leftarrow \gamma \cup \gamma_g$
**end for**
$M \leftarrow \emptyset$ %player, marginal utility info (per GM)
**for** $g \in G$ **do**
  $m_g \leftarrow$ Algorithm 3$(W, g,\ A^*, \gamma_g)$
  $M \leftarrow M \cup \{m_g\}$
**end for**
**while** $|A^*| > 0$ **do**
  $F \leftarrow \emptyset$ % Bids for current favorites
  **for** $g \in G$ **do**
    **if** $|\gamma_g| < c_u$ **then**
      $f_g(uid) \leftarrow k$ s.t. $m_k = \max_i\{m_i|\ \langle i, m_i\rangle \in M_g\}$
      $f_g(util) \leftarrow m_{f_g(uid)}$
      $F \leftarrow F \cup \{\langle f_g(uid), f_g(util)\rangle\}$
    **else**
      $F \leftarrow F \cup \{\emptyset\}$
    **end if**
  **end for**
  % Distribute players based on bids
  **for** $g, j \in G \times G : g \neq j$ **do**
    **if** $f_g(uid) = f_j(uid)$ **then**
      **if** $f_g(util) < f_j(util)$ **then**
        $f_g \leftarrow \emptyset$
      **else if** $f_g(util) > f_j(util)$ **then**
        $f_j \leftarrow \emptyset$
      **else if** $|\gamma_g| < |\gamma_j|$ **then**
        $f_j \leftarrow \emptyset$
      **else if** $|\gamma_g| > |\gamma_j)|$ **then**
        $f_g \leftarrow \emptyset$
      **else**
        $f_k \leftarrow \emptyset$ : random $k \in \{g, j\}$
      **end if**
    **end if**
  **end for**
  **for** $g \in G$ **do**
    **if** $f_g \neq \emptyset$ **then**
      $\gamma_g \leftarrow \gamma_g \cup \{f_g(uid)\}$
      $A^* \leftarrow A^* \setminus \{f_g(uid)\}$
    **end if**
    $m_g \leftarrow$ Algorithm 3$(W, g,\ A^*, \gamma_g)$
  **end for**
**end while**
**return:** $\gamma$
```

$^+$ See Algorithm 3

Algorithm 3 Update coalition values

```
Input: ATFG $W$, GM $g$, player set $A^*$, Coalition $C$ s.t. $g \in C$
Output: $M_g$ % Pairs $\langle i, m_i \rangle$ where $m_i$ is the marginal utility of adding $i$ to $C$
$M_g \leftarrow \emptyset$
$util_C \leftarrow 0$
for $k \in C$ do
    $util_C \leftarrow util_C + u_k(C)$
end for
for $i \in A^*$ do
    $util \leftarrow 0$
    for $k \in C \cup \{i\}$ do
        $util \leftarrow util + u_k(C \cup \{i\})$
    end for
    $m_i \leftarrow util - util_C$
    $M_g \leftarrow M_g \cup \{\langle i, m_i \rangle\}$
end for
return: $M_g$
```

robin algorithm where only the preferences of the GM were considered. However, it performed substantially worse than the URR algorithm, so it was abandoned.

Algorithm 4 is a **local search** heuristic, with the goal of minimizing the number of blocking players. Each iteration of the algorithm chooses a random set of players to be GMs, and randomly assigns other players to the GMs' tables. It then repeatedly chooses a neighboring partition (defined by a single player being assigned to a different table) that improves (lowers) the fitness.

Note that it is possible that a partition examined by Algorithm 4 is not Nash stable, but has no improving neighbor — even if there is a Nash stable partition for that ATFG instance. This local-but-not-global optimum is a typical phenomenon of local search algorithms; to handle this, we use multiple random re-starts.

An iteration of local search has $O(n)$ improvements of fitness, since at most n agents can *want* to deviate at a given time. Further, since an agent only is allowed to move if the movement improves the fitness, we avoid potential loops. The algorithm maintains a GM × players matrix of valuations, indicating the value each player derives from each GM's table. A scan of a player's row can identify if they wish to move; to test stability requires scanning the $O(n^2)$ matrix, and potentially updating it if a player moves. Each move of player i from GM j to GM k requires $O(n^2)$-time updates to the utilities of all $O(n)$ players for the coalitions with GMs j and k. The hardest part of the algorithm is determining which agent, if any, want to move. For each of the n potential movers, i, and for each of the $O(n)$ coalitions C they might join, we check, for each other agent j, does j newly want to move? Particularly, do they want to move to to $C \cup \{i\}$, to the coalition i just left, or another coalition if $j \in C$ and j doesn't like the addition of i? Determining if j wants to move takes $O(n^2)$, by Observation 1. Thus, the entire check for acceptable moves takes $O(n^5)$, and an iteration of LS, involving $O(n)$ such checks, is $O(n^6)$.

Algorithm 4 Local Search

```
Input: ATFG $W$, GM set $G$, player set $A^*$
Output: $\gamma$, Potentially Nash stable partition
$\gamma \leftarrow \emptyset$ %randomly assigned partition
for all $g \in G$ do
    $\gamma_g \leftarrow \frac{|A^*|}{|G|}$ random players $^+$
    $\gamma \leftarrow \gamma \cup \{\{g\} \cup \gamma_g\}$
end for
$M \leftarrow \emptyset$
for all $i \in N$ do
    if $i$ can receive higher utility by moving then
        $m_i \leftarrow$ # players wanting to move after $i$ moves
        $M \leftarrow M \cup \{m_i\}$
    end if
end for
while $\exists m \in M : m < |M|$ do
    choose $\{i : m_i = \min(M)\}$
    $\gamma \leftarrow \gamma$ modified by $i$'s move
    $M \leftarrow \emptyset$
    for all $i \in N$ do
        if $i$ can receive higher utility by moving then
            $m_i \leftarrow$ # players wanting to move after $i$ moves
            $M \leftarrow M \cup \{m_i\}$
        end if
    end for
end while
return: $\gamma$
```

$^+$ $\frac{|A^*|}{|G|}$ is $\lceil \frac{|A^*|}{|G|} \rceil$ or $\lfloor \frac{|A^*|}{|G|} \rfloor$ based on players available

Our third heuristic, *DC,* daisy-chains URR and local search, by using the partition output by URR to hot-start the local search algorithm. The complexity of DC is $O(n^6)$. In order to facilitate the creation of new coalitions over the course of several movements, both local search and daisy-chain are permitted to create partitions with singletons even in games where singletons are not considered valid coalitions. If singletons are not valid coalitions for a given instance, then the partition is invalid and cannot be stable. The code we use to verify whether a partition is Nash stable specifically checks the partition to ensure that all coalitions are of a valid size; partitions containing one or more coalitions of invalid size are considered unstable.

5.6 Experiments

Our experimental contributions are results for our three greedy heuristic algorithms, demonstrating the relative benefits and detriments of each and showing that all three outperform a deterministic algorithm on instances with 12 or more players.

Experimental Setup

Our heuristics were coded in Python 3.7 and currently do not rely on third-party packages. We tested our heuristics against three hand-crafted benchmark instances with known Nash stable partitions, two with 10 players and one with 25 players. Each 10-player instance contains a known stable partition consisting of a 4-player coalition and two 3-player coalitions. The 25-player instance contains a known stable partition consisting of 5-player coalitions. Initially these instances were used primarily for proof-of-concept checks and to ensure that heuristics and the complete solver produced correctly formatted output. Recently, we conducted 10 trials on each of the 3 hand-crafted instances with a cap of 10,000 restarts per instance. We collected data on the total number of restarts before a Nash stable partition was found, and the time duration of the trials, as well as the number of restarts and time per size of instance. These trials were run on Windows 10 Pro with a AMD Ryzen 1700 Processor clocked at 3.0GHz and 32GB of RAM; the hardware differs from trials for randomly generated instances because of time constraints.

We also tested the heuristics against randomly generated instances; 50 with 10 players; 20 with 12 players; 25 with 20 players; 25 with 24 players; 20 with 25 players; 25 with 30 players. To randomly generate these instances, we used coalition size limits of 3–5 for 10, 12, and 15 player instances and 3–8 for 20, 24, 25, and 30 player instances, asymmetric valuations between -n and n inclusive for individuals, pairs, and GMs, and a pair valuation probability of 0.15.

Using the complete algorithm, we found Nash stable partitions for 24 of the randomly generated 10-player instances and 12 of the 12-player instances. We then ran the heuristic algorithms with fixed maximum numbers of random restarts. (Note that the heuristic algorithms halt as soon as they find a single Nash stable partition for the given instance.)

For each heuristic, we conducted 10 trials each for the 10, 12, 15, 20, 24, 25, and 30 player instances. Trials for the utilitarian round robin heuristic were capped at 100,000 restarts for 10 and 12-player instances, and 200,000 for 15, 20, 24, 25, and 30-player instances, while trials for local search and daisy-chain were capped at 10,000 restarts.[2] Our experiments were run on Linux Ubuntu with an Intel Xeon Processor clocked at 2.1GHz and 4GB of RAM. We collected data on the total number of restarts before a Nash stable partition was found, and the time duration of the trials, as well as the number of restarts and time per size of instance.

At the recommendation of reviewers for our recent paper submissions, we generated a new, more consistent set of test instances to run our heuristics on. We generated 25 15-player, 25 20-player, 25 25-player, 25 and 30-player instances with coalition size limits of 3–6, symmetric valuations between -n and n inclusive for indi-

[2] daisy-chain was given 50,000 restarts for the 24-player instances

viduals, pairs, and GMs, and a pair valuation probability of 0.15. We also generated new 10-player instances intended to be used for additional testing; however, due to a very low percentage of stabilizable instances and time constraints, we chose not to conduct tests on additional 10-player instances. Due to a bug discovered after the trials finished running, the URR heuristic continued running until a stable partition was found whenever DC found a stable partition, so the number of restarts was not capped correctly. (Note that DC can take an unstable partition and improve it until it's stable, so URR may "know" that the instance is stabilizable, in this setting, even if URR has not yet found a stable partition.)

On these new instances, we conducted 10 trials for each heuristic. Trials were capped at 10,000 restarts for all heuristics. Most of these experiments were run on Linux Ubuntu with an Intel Xeon Processor clocked at 2.1GHz and 4GB of RAM; however, the LS trials on the 30-player instances were run on Windows 10 Pro with a AMD Ryzen 1700 Processor clocked at 3.0GHz and 32GB of RAM due to time constraints.

While updating the code to run these new trials, we discovered a significant bug that affected the results of the local search heuristic on the original randomly generated instances. Extensive examination of our output data has proved that *only* the local search heuristic results were affected by this bug. We have not re-run all of the local search trials on the old instances. To maintain the integrity of our work, we have removed the results from the LS trials on the old randomly generated instances. We have since fixed the bug and have conducted extensive testing to ensure that this issue has been resolved, so we include LS results for the trials on the hand-crafted instances and the new randomly generated instances.

In order to ensure consistent results, the URR and DC trials are run simultaneously in the trials on the hand-crafted instances and the new randomly generated instances; the output from each URR iteration being used as input for the local search portion of the DC heuristic. Once a stable partition has been found by the DC heuristic, no further URR outputs are fed into the local search portion of DC. This means that the runtime for URR is inflated for some of its iterations due to waiting on the local search portion of the DC heuristic to finish.

Results

In this section we discuss the results of our experiments. In the oldest experiments, Randomly Generated Instances: Part 1, we have omitted the results from LS testing due to the previously-discussed bug discovered in the code. LS results are included in the trials on the hand-crafted instances and new randomly generated instances because the bug was fixed before those tests were run.

Randomly Generated Instances: Part 1

We observe that the daisy-chain heuristic is significantly slower per iteration than URR, so our tests of DC use a factor of 10 fewer restarts with most tests to achieve more comparable runtimes.

While URR was not as effective as we had hoped in finding Nash stable partitions, it often provided partitions that were close to stable, i.e., with *low degree of instability.* Remember that the degree of instability is the number of agents that would individually prefer to deviate from their assigned coalitions. Using local search to relocate those individuals worked significantly better than either algorithm individually. (See Table 5.1.)

We observe that the percentage of instances for which the heuristics found Nash stable partitions is quite small for instances of sizes 25 and 30. We attribute this to an insufficiency of random restarts; with a larger number of cores on our VMs, we are confident that the algorithms will run significantly faster, permitting more restarts which should increase these percentages. We were unable (so far) to verify the existence of Nash stable partitions on those instances using the complete algorithm, but with high probability, a larger percentage of them have Nash stable partitions than these results indicate.

Table 5.1: Average runtime (CPU seconds) and percentage of instances for which NS coalitions are found.

# Players	URR		DC	
	time	%	time	%
10	24.5	47.3	2.0	36.5
12	43.2	17	22.9	**52.5**
15	239.0	65.5	44.6	**71.5**
20	898.4	20.4	178.1	**25**
24	268.4	**94.4**	748.5	79.6
25	1745.7	0	591.9	**4**
30	2605.8	**7.6**	1533.8	2.5

Note: For the 24 agent DC test, 50,000 restarts were allowed.

Of the randomly generated instances: 48% of our 10-player instances are stabilizable, 60% of 12-player instances are stabilizable, 95% of 15-player instances are stabilizable, and 96% of 20-player instances are stabilizable. We initially conjectured that a smaller percentage of 10 and 12-player instances are stabilizable because they both have utility ranges from -20 to +20 (i.e. larger than the number of agents), whereas the other instances were generated with limits no wider than -n to +n where n is the number of agents. After examining newly created 10-player instances intended to be used for new experiments, this conjecture appears to be incorrect. In future work, we plan to examine parameters that make it more or less likely for a given instance to be Nash stabilizable.

We provide information on the degree of instability (DoI), the number of agents who prefer to move, of coalitions found by DC and URR. This provides additional evidence that DC generally improves on the performance of URR.

Table 5.2: Average DoI

# Players	Number of movers	
	URR	DC
15	0.345	**0.045**
20	0.952	**0.116**
24	0.056	**0.028**
25	2.455	**1.105**
30	1.3	**1.18**

Hand-crafted instances

The results of our experiments on the hand-crafted instances, summarized in Table 5.3, show that local search is much worse at finding stable partitions than either of the other heuristics. The number of possible partitions is *much* larger than the number of Nash stable partitions for most of these instances, so LS has to be extremely lucky to be started near a local optimum that is in fact globally optimal. This is evidenced by its comparatively poor performance. However, by applying local search to the result of the round robin heuristic, we see a substantial improvement in runtime and the number of restarts needed to find a stable partition.

Table 5.3: Average runtime (CPU seconds), average restarts, and percentage of tests where NS coalitions are found.

# Players	URR			DC			LS		
	time	restarts	%	time	restarts	%	time	restarts	%
10	0.10	111.9	100	**0.02**	5.1	100	41.8	9001	10
10	0.08	81.1	100	**0.03**	11.5	100	19.8	8001	20
25	7.4	2210.2	100	**0.01**	1	100	210.9	5005	50

Note: All hand-crafted instances contain at least one known Nash stable partition.

All three heuristics reached partitions with a DoI of 0 on all trials. This seems to clash with the rate at which LS succeeded in finding Nash stable partitions; however, many of the local search results included singleton agents who did not wish to join any of the available coalitions. Since singletons are not valid coalitions for any of the hand-crafted instances, such partitions are invalid and cannot be Nash stable.

In contrast, URR and DC succeeded in finding Nash stable partitions on all three instances very quickly.

Randomly Generated Instances: Part 2

The results of our trials on the new randomly generated instances are outlined in Table 5.4. As with experiments on the hand-crafted instances, local search is not particularly good at finding stable partitions on its own. Due to a bug in the code, DC and URR find stable partitions an equal percent of the time, which caused the URR to continue running until a stable partition was found whenever DC found a stable partition. Though not intentional, this bug highlights the difference in the number of restarts necessary for DC and URR to find stable partitions on randomly generated data. While each iteration of URR runs much faster than each iteration of DC, URR requires many more restarts than DC to find a Nash stable partition.

Table 5.4: Average runtime (CPU seconds), average restarts, and success percentage of tests where NS coalitions are found.

# Players	URR[1]			DC			LS		
	time[2]	restarts	%	time	restarts	%	time	restarts	%
15	102.5	55643.2	25.2	54.6	8229.2	25.2	94.6	9924.8	0.8
20	307.7	87570.2	9.6	141.4	9302.3	9.6	269.1	9963.3	0.4
25	840.0	141436.1	1.2	369.8	9951.4	1.2	689.7	10000.0	0.0
30	153.0	6013.6	94.0	127.5	1916.18	94.0	2393.3 [3]	9759.2	6.8

[1] Due to a bug in the code, the URR heuristic ran until a stable partition was found in any trial where DC found a stable partition.
[2] The URR heuristic has to wait for the local search aspect of DC before running another iteration, so its times are bloated.
[3] The 30 agent LS trials were run on different hardware than the other trials.

Table 5.5 describes the DoI for our trials on the new randomly generated instances. As with the original test data, we find that DC produces the best DoI on average. Similar to the tests run on the hand-crafted instances, local search often found partitions where no player wants to move, but the partition is invalid due to the presence of singleton coalitions. However, in this set of tests, the same thing also happens with the DC heuristic.

Table 5.5: Average DoI

# Players	Number of movers		
	URR	DC	LS
15	1.232	**0.0**	**0.0**
20	2.256	**0.008**	0.036
25	2.504	**0.02**	0.208
30	0.084	**0.012**	0.412

We used the complete algorithm to examine the 15-player instances and found that 15 of the 25 (60%) are stabilizable. We do not know yet why this set of instances has a lower percentage of stabilizable instances than the previous set of 15-player instances. We plan to investigate this in future work. Due to computational complexity, we did not examine the 20-player or larger instances with the complete algorithm.

5.7 Conclusions and Open Questions

We have introduced a novel coalition formation game for a subject dear to our hearts, namely social and (in pre-pandemic days) in person gaming. Like many such cooperative games, finding stable partitions is NP-hard. However, we have introduced two fast and effective heuristic algorithms to find Nash stable partitions. We focus on Nash stability because we believe it is, socially, the most relevant notion in the literature; though we can imagine social cliques that would be blocking coalitions for core stability, we imagine that such cliques would not throw themselves into the centralized assignment process.

What conditions guarantee or increase the likelihood that Nash stable partitions exist? Could decentralized matching algorithms in the style of Taywade et al. [64, 65] be used to solve ATFGs efficiently? We hypothesize that work by Gairing and Savani

[26] can be extended to better characterize the theoretical efficacy of local search in ATFGs.

Chapter 6 Internal Stability

In the last two chapters, we looked at novel hedonic games. We conclude the technical part of the dissertation by looking at a novel notion of stability that arose naturally in work by Taywade et al. [64].

One stability concept that is of perennial interest in coalition formation games is core stability [3, 6, 9, 19, 20, 28, 29, 30, 45]. Core stability is defined as the lack of agents who are incentivized to leave their assigned coalition(s) and form a *blocking coalition*. This assumes that agents can easily identify others to form a blocking coalition; however, this is not always realistic. Consider the task of partitioning first-semester students into project groups. Some of the assigned groups may find that a subgroup would rather work together, abandoning the rest of their group. Since the students were not acquainted beforehand, they are less likely to coordinate across separate groups. We could also consider dividing gamers up into play groups at a convention. To address this possibility, we focus on the situation where any new groups formed after the semester starts will consist solely of subgroups of previously-existing groups.

Cases where blocking coalitions were confined to subcoalitions that split away from an existing coalition were first considered in the context of hedonic games by Dimitrov et al. (2006) and Alcalde and Romero-Medina (2006) who referred to it as *internal stability*. We extend the notion of an internally stable coalition to partitions, by defining that internal stability holds for a partition of agents into coalitions when no subgroup of any assigned coalition is incentivized to break away. This stability notion has appeared in other hedonic games works such as Taywade, Goldsmith, and Harrison's (2018) work on decentralized hedonic games where coalitions are only blocked by subcoalitions. That work developed a decentralized heuristic for several classes of hedonic games using a grid world where agents with limited memory move around the world in order to form coalitions. Due to the grid world model and agents' limited memory, internal stability was a natural option to consider when evaluating a partition's stability.

A natural question is what the relationship is between internal stability and Nash stability. We show that Nash stability implies internal stability in some classes of hedonic games, but not in others. Table 6.1 at the end of section 6.4 provides a summary of our Nash stability results. While communication channels such as Slack have increased workers' abilities to interact with distant co-workers, the workers may still defect from the larger group and form private sub-channels. In such myopic situations, internal stability seems a more realistic measure of the sustainability of a work-group assignment.

Stability notions focus on outcomes that are likely to occur due to agents' selfish behavior. Stable outcomes may provide the best individual outcomes for all agents, but this is not always the case. The price of stability (PoS) provides a metric to gauge the utility lost in order to achieve stability.

We introduce here a natural and important extension of Dimitrov et al. (2006) and Alcalde and Romero-Medina's (2006) internal stability. We investigate the relationships of internal stability to other, more common notions, and show that it is distinct from core and Nash stability, for multiple types of hedonic games. We investigate the prices of anarchy and stability for internal stability with respect to several hedonic games. This work gives insight into a key stability notion when agents are myopic.

6.1 Motivation

Consider a search and rescue effort following a large avalanche. Ideally, those caught up in the avalanche will all be rescued without endangering members of the rescue effort. Unfortunately, this is rarely possible. Some victims may be killed by the avalanche itself, while others may die before rescuers can reach them. Additionally, the rescue operations can be dangerous, and the possibility of another avalanche must be considered. Further, the effort will have to cope with incomplete information about the situation.

Shortly after the avalanche, the rescue effort will likely have some information about the situation. There may be estimations of how many victims are likely to be found in different parts of the disaster area, but it's unlikely that the precise locations of victims will be known initially. Further, there may be area-based predictions on the likelihood victims survived the avalanche itself; this is important, since survivors should take priority over casualties, but survivors with more severe injuries may require higher priority in order to survive. Finally, the location(s) of operations base(s) will be known in relation to the disaster zone.

In order to carry out the rescue mission successfully, several tasks need to be carried out in parallel: victims must be located, victims must be rescued (or their bodies retrieved), status of rescue teams must be monitored, and the disaster zone conditions must be monitored for additional hazards, which may include another avalanche. Not all agents will have the same skill set, and most teams will benefit from a variety of agents. For example, teams sent to rescue victims will be more effective if they include agents skilled at locating victims, agents capable of rescuing victims from potentially hazardous locations, agents capable of providing emergency medical aid, and agents able to monitor their immediate surroundings for previously unknown or impending hazards. The size of a team should also be appropriate for the task they're assigned; a team sent to rescue five to ten victims whose vital status is unknown will require more agents than a team sent to rescue one victim who is stranded, but is known to be alive and relatively unharmed.

It is in the best interests of all involved to minimize the likelihood that teams will not split up. When a team splits, the resulting new teams may not be as well balanced or as well positioned as they otherwise could have been. They also will likely be unable to join agents from another team for a number of reasons (distance, obstacles, unawareness of each other, etc.). In this example, we do not consider team members acting in parallel in close proximity to each other splitting up (ie. a team sent to rescue five people could reasonably extract those people in parallel if they are close enough together).

6.2 Related Works

Works published in 2006 by Dimitrov et al. and Alcalde and Romero-Medina introduce internal stability for coalitions, observing that a singleton coalition is always internally stable [1, 20]. Alcalde and Romero-Medina use internally stable coalitions as a tool to investigate conditions that guarantee the existence of core stable partitions in hedonic games [1]. Dimitrov et al. use internal stability to define another stability notion, deviation stability, which they use to prove the existence of core stable partitions in friend and enemy-oriented hedonic games [20]. Since its introduction, the only other paper that discusses internal stability is a paper by Liu et al. which adapts it to matching and exchange contexts [41]. One can view internal stability as relevant when agents are *myopic,* meaning that their awareness of the desirability of other agents is limited to those agents that are nearby, namely in the same coalition. Another work that considered agents with limited preference knowledge is PAC-stability, proposed by Sliwinski and Zick, which could be understood as resistance to random attempts by myopic agents to form blocking coalitions [58].

Carosi et al. proposed a notion of local core stability for Simple Symmetric Fractional Hedonic Games (SS-FHGs): FHGs where all edges are undirected and have a weight of 1 [12]. A partition is locally core stable when there is no clique such that all agents within the clique derive more utility from the clique than from their current coalition(s) [12]. Carosi et al. also prove that the price of stability for local core stability is between 2 and 8/3 for SS-FHGs [12].

Nguyen and Rothe proposed three notions of local fairness in hedonic games; a partition is said to be locally fair if all players' coalitions are at least as good as a threshold coalition assigned to each player based upon their preferences [44]. Nguyen et al. examine the existence of locally fair coalitions in general hedonic games and ASHGs [44]. Nguyen and Rothe also study the complexity of determining if locally fair coalitions exist, and they investigate the price of fairness for the notions they propose [44]. Fairness is beyond the scope of our work on internal stability, but the interest in more localized partition properties and on the cost associated with desirable properties of partitions is similar to our own.

6.3 Preliminaries

Internal stability considers group deviations where all deviating agents must come from the same coalition.

Definition 40. *A coalition C is **internally stable** if there is not subset $D \subset C$ such that all of the agents of D are better off leaving C and forming a new coalition [1, 20].*

*A partition π is **internally stable** if all coalitions $C \in \pi$ are internally stable.*

In the next section we clarify the relationship between internal stability and two of the most common stability notions in hedonic games.

6.4 Relationship to Core and Nash Stability

It might seem, at first, that internal stability is very similar to Nash stability, or perhaps to core stability. We investigate the relationship of internal stability to Nash stability and core stability for a variety of hedonic games.

Observation 3 immediately follows from Dimitrov et al.'s (2006) observation that all coalitions in a core stable partition are internally stable.

Observation 3. *Core stability implies internal stability for all classes of hedonic games.*

Note that all singleton partitions (i.e., each agent in a coalition of size 1) are trivially internally stable. However, in most instances of most hedonic games, a singleton partition will neither be core stable nor Nash stable. Thus, internal stability does not, generally, imply core stability or Nash stability. Theorem 3 shows our first step in analyzing the relationship between Nash stability and internal stability.

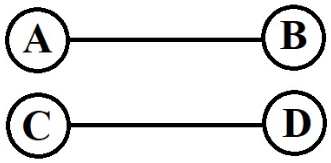
Theorem 3. *Nash stability does not guarantee internal stability in FOHGs.*

Proof. Consider a graph $G = (V, E)$ with $V = \{A, B, C, D\}$ and $E = \{(A, B), (C, D)\}$. (See Figure 6.1.) Consider a FOHG based on this graph such that the set of agents $N = V$ and each edge $(i, j) \in E$ defines a mutual friendship. All pairs $(i, j) \notin E$ indicate mutual enmity.

Now consider the grand coalition for this FOHG. Each agent $i \in N$ gains 4 utility from the presence of their mutual friend, but also loses 2 utility due to the 2 enemies present. This gives each agent i a net utility of 2, making the grand coalition individually rational. The only deviation a single agent can make to the grand coalition is to leave the coalition and become a singleton, thereby earning 0 utility. Thus, the grand coalition is Nash stable. If we view the grand coalition from the viewpoint of internal stability, however, we see that for pairs of agents (A, B) and (C, D), each agent pair can leave the grand coalition and increase the utility of both agents from 2 to 4. Thus, the grand coalition is not internally stable and we conclude that not all Nash stable partitions are internally stable. □

Since FOHGs are a subclass of ASHGs, Corollary 2 immediately follows from Theorem 3.

Figure 6.1: Nash stable, not internally stable



Corollary 2. *Nash stability does not imply internal stability for ASHGs.*

While internal stability is not implied by Nash stability in general case ASHGs, there are subclasses where Nash stability does imply internal stability. We show that enemy-oriented hedonic games are one such subclass; in particular, we show that not only does Nash stability imply internal stability, but individual rationality is sufficient to imply internal stability.

Lemma 1. *Individual rationality guarantees internal stability in EOHGs.*

Proof. By the definition of EOHGs the only individually rational coalitions for some agent $i \in N$ are those which contain none of i's enemies. Equivalently, all individually rational coalitions can be described by cliques in a graph of friendship relations.

Since all agents in an individually rational coalition are friends with each other, no subset of agents in such a coalition can increase their utility by leaving. Therefore, all individually rational coalitions are internally stable. A partition is only individually rational if all its coalitions are individually rational and, therefore, internally stable. Thus, all individually rational partitions are internally stable. □

Theorem 4. *Nash stability guarantees internal stability for EOHGs.*

Proof. Because all Nash stable partitions are individually rational, it follows from Lemma 1 that Nash stability implies internal stability. □

Theorems 3 and 4 show that the relationship between Nash and internal stability varies between subclasses of ASHGs. We expand our understanding of the relationship between Nash and internal stability beyond the scope of ASHGs in Theorem 5 by analyzing fractional hedonic games.

Theorem 5. *Nash stability does not guarantee internal stability in fractional hedonic games.*

Proof. Construct a fractional hedonic game, G, with agents $\{A, B, C, D\}$ with $u_A(B) = u_B(A) = 4$ and $u_X(Y) = 1$ for all other $X \neq Y$. Consider the grand coalition, S. Then $u_A(S) = u_B(S) = \frac{4+1+1}{4} = \frac{6}{4}$ and $u_C(S) = u_D(S) = \frac{1+1+1}{4} = \frac{3}{4}$. This coalition is individually rational, since each agent has positive utility. It is also Nash Stable, since an agent's only defection option is to leave S and form a singleton coalition, with utility 0. However, the coalition $\{A, B\}$ provides utility 2 for each of A and B, so S is not internally stable. Therefore, for FHGs, Nash stability does not imply internal stability. □

We now state the relationship between Nash and internal stability in altruistic hedonic games in Theorem 6.

Theorem 6. *Nash stability does not guarantee internal stability for any of the three altruism levels of altruistic hedonic games (AHGs).*

Proof. Consider an AHG based with agents $\{A, B, C, D\}$. Let $\{A, B\}$ and $\{C, D\}$ be mutual friends that regard all other agents as enemies. Since there are 4 agents in this game, we set $M = n^5 = 4^5 = 1024$.

In the grand coalition all agents achieve a utility of 2050 in the *Selfish-First* and *Altruistic First* paradigms and 2 in *Equal Treatment.* Individual agents can only deviate by becoming a singleton, so the grand coalition is Nash stable for all three paradigms.

Now consider if $\{A, B\}$ or $\{C, D\}$ broke away; the agents breaking away derive a utility of 5000 in the *Selfish-First* and *Altruistic First* paradigms and a utility of

4 in the *Equal Treatment* paradigm. The grand coalition is not internally stable for any of the three paradigms, because both $\{A, B\}$ and $\{C, D\}$ have incentive to break away. □

Because Schlueter and Goldsmith (2020) showed that selfish-first AHGs are special cases of Super Altruistic Hedonic Games, Corollary 3 follows from Theorem 6.

Corollary 3. *Nash stability does not imply internal stability in Super Altruistic Hedonic Games.*

The proofs for Theorems 3–6 also give Observation 4.

Observation 4. *If there exists an instance of a coalition formation game such that the grand coalition is individually rational, but not internally stable, then Nash stability does not imply internal stability for that class of games.*

The games we have examined thus far assume that agents' utility is based directly on the other agents in their coalition. In Role Based Hedonic Games, utility is based instead on an agent's assigned role and the team's role composition.

Theorem 7. *Nash stability does not imply internal stability in RBHGs.*

Proof. Consider an RBHG instance with the following setup: $P = \{p_1, p_2, p_3, p_4\}$, $R = \{r_1, r_2\}$, $C = \{\{r_1, r_2\}, \{r_1, r_1, r_1, r_1\}, \{r_1\}, \{r_2\}\}$. Now $\forall i \in P$ let $u_i(r_1, \{r_1, r_1, r_1, r_1\}) = 1$ and $\forall (r, c) \in R \times C$ let $u_{p_1}(r, c) = u_{p_3}(r, c)$ and let $u_{p_2}(r, c) = u_{p_4}(r, c)$. Let $u_{p_1}(r_1, \{r_1, r_2\}) = 2$ and $u_{p_1}(r_2, \{r_1, r_2\}) = -1$. Let $u_{p_2}(r_1, \{r_1, r_2\}) = -1$ and $u_{p_2}(r_2, \{r_1, r_2\}) = 2$. As usual for singletons, $\forall i \in P$ let $u_i(x, \{x\}) = 0$ for $x = r_1, r_2$.

Now let a partition π form where all four agents are put in the grand coalition with the composition $\{r_1, r_1, r_1, r_1\}$. Since every role in the composition is r_1, all agents derive a utility of 1 from this partition. The only way an individual can deviate from this partition is to leave to become a singleton of either role r_1 or r_2; in either case, agents derive zero utility as a singleton. Since the only deviations available to individuals will reduce their utility from 1 to 0, the partition of all agents in the grand coalition with composition $\{r_1, r_1, r_1, r_1\}$ is Nash stable.

While the grand coalition with composition $\{r_1, r_1, r_1, r_1\}$ is Nash stable, it is not internally stable. Either agent p_1 or p_3 could join agent p_2 or p_4 and split away to form a new coalition with composition $\{r_1, r_2\}$ with p_1 or p_3 taking role r_1 and p_2 or p_4 taking role r_2; doing this would increase the utility of both deviating agents from 1 to 2. Thus, Nash stability does not imply internal stability in RBHGs. □

Roles and Teams Hedonic Games [60] are a subclass of RBHGs that impose a strict team size rule that makes it impossible for a subset of a valid existing team to break away to form a new, valid team.

Observation 5. *All valid partitions are internally stable in RTHGs.*

In instances of the Group Activity Selection Problem, agents derive utility from the selected activity and the size of their coalition.

Theorem 8. *Nash stability does not imply internal stability in GASPs.*

Proof. Consider a GASP instance where there are n agents, and activity set $A = \{a_1, a_\emptyset\}$ and $\forall i \in N$ $(a_1, n-2) \succ_i (a_1, n) \succ_i a_\emptyset$; all alternatives not included in the preference profile are seen as worse than the void activity.

Now consider the case where all n agents form a single coalition to participate in activity a_1. We can see from the preference profile shared by all agents that this outcome is preferable to the void activity and to participating in activity a_1 alone. Since the only way for an individual to deviate from this outcome is to leave and do nothing, or participate in a_1 by themselves, no individual agent has incentive to deviate from this outcome. Thus, the outcome is Nash stable.

From the perspective of internal stability, we see that any subset of $n-2$ agents could break away and achieve a more favorable outcome. Thus, the grand coalition participating in a_1 is not a internally stable outcome. □

In this section, we have shown that that Nash stability implies internal stability in some classes of hedonic games, but not in others. We also observe that core stability guarantees internal stability for all classes of hedonic games. Table 6.1 provides a summary of our Nash stability results.

Table 6.1: Summary of Relationship of Nash Stability to Internal Stability

Hedonic Game	Nash Stability ⇒ Internal Stability?	Where?
Additively Separable	No	Corollary 2
Enemy-oriented	Yes	Theorem 4
Friend-oriented	No	Theorem 3
Fractional	No	Theorem 5
Altruistic	No	Theorem 6
Super Altruistic	No	Corollary 3
Role Based	No	Theorem 7
Roles and Teams	Yes	Observation 5
Group Activity Selection	No	Theorem 8

6.5 Price of Stability (PoS)

We present some results on the PoS of internal stability in hedonic games. The price of stability for a particular notion of stability and a particular hedonic game is a measure of the coupling between stability and utility for that game. A fixed, finite PoS tells us that the utility of any stable partition is bounded by a constant times the utility of any other, whereas an unbounded PoS tells us that utility is not coupled with this notion of stability.

Theorem 9. *$PoS_{IS}(G)$ is unbounded in hedonic games where agents assign asymmetric values to each other. This includes general case ASHGs and FHGs.*

Proof. Consider a hedonic game with the following setup: $N = \{1, 2, 3\}$, $v_1(2) = 10$, $v_1(3) = -1$, $v_2(1) = -1$, $v_2(3) = -1$, $v_3(1) = -1$, $v_3(2) = 10$. In both ASHGs and FHGs, the utility is maximized by the grand coalition, but the only stable partition is the partition of singletons. Since the partition of singletons has a sum utility of zero, PoS$= \infty$. This scenario can be adapted to any hedonic game where agents derive utility from potentially asymmetric values assigned to each other. □

While $\text{PoS}_{\text{IS}}(G)$ is unbounded for some classes of games, we show that it is bounded in others in Theorem 10.

Theorem 10. *PoS_{IS} for symmetric ASHGs is 1.*

Proof. Consider a symmetric ASHG. Consider an optimal coalition A and suppose that it contains a blocking subcoalition $B \subset A$. Note that the total utility for A is the sum of utilities of agents within B, agents in $A \setminus B$, and utilities (in each direction) between B and $A \setminus B$: $\sum_{x \in B} u_x(B) + \sum_{y \in A \setminus B} u_y(A \setminus B) + 2 \cdot \sum_{x \in B, y \in A \setminus B} u_x(y)$. Since B is a blocking subcoalition $\sum_{x \in B} u_x(B) > \sum_{x \in B} u_x(A) = \sum_{x \in B} u_x(B) + \sum_{x \in B, y \in A \setminus B} u_x(y)$. Therefore, $\sum_{x \in B, y \in A \setminus B} u_x(y) < 0$.

Thus, the total utility for the partition $B, A \setminus B$ has utility $\sum_{x \in B} u_x(B) + \sum_{y \in A \setminus B} u_y(A \setminus B) > \sum_{x \in B} u_x(B) + \sum_{y \in A \setminus B} u_y(A \setminus B) + 2 \cdot \sum_{x \in B, y \in A \setminus B} u_x(y)$, contradicting the optimality of A.

Thus, any optimal coalition has maximum utility over sub-partitions, and is internally stable. Therefore, $\text{PoS}_{\text{IS}}(G) = 1$ for all symmetric ASHGs. □

In theorem 11 we provide another class of games where price of internal stability is bounded.

Theorem 11. *PoS_{IS} for symmetric FHGs is bounded by 2.*

Proof. We define an FHG, G, which maximizes the ratio between the utility-maximizing partition and the utility-maximizing internally stable partition. If $\text{PoS}_{\text{IS}}(G) > 1$, then the utility-maximizing partition is not internally stable. Without loss of generality, we consider a single coalition. (The maximum PoS will occur when each coalition is split; since we take the average utility over all agents, it suffices to consider a single coalition A to find the maximum PoS as if that were the grand coalition.) Let $B \subset A$

be a blocking coalition. We also assign $a = |A|$, $b = |B|$, and $r = a - b$. Keep in mind that, for an agent in B, its average utility in B is higher than its average utility in A. Because, for FHGs, agents' utilities for other agents in their coalition are averaged, and agent's utilities for others need only distinguish three cases. We define v_b as the value agents in B assign to each other; $\forall i, j \in B, v_i(j) = v_j(i) = v_b$. We define v_r as the value agents in $A \setminus B$ assign to each other; $\forall i, j \in A \setminus B, v_i(j) = v_j(i) = v_r$. We define v_a as the value agents in B assign to agents in $A \setminus B$ and vice-versa; $\forall i \in B, j \in A \setminus B, v_i(j) = v_j(i) = v_a$.

In order for agents in B to be incentivized to break away from the rest of A, the value each agent in B receives in B (which approaches v_b as $|B|$ grows) is greater than the value it receives in A, which is $\frac{(b-1)v_b + rv_r}{b+r}$. This implies that $v_b > v_r$. We see that the grand coalition maximizes the sum of agents' utilities when $v_b - \frac{1}{b} + v_r - \frac{1}{r} < 2v_a$. We are able to show that the ratio between $v_b : 2v_a$ is a hard upper bound on the ratio between the sum utilities of the grand coalition and $\{A \setminus B, B\}$, allowing us to construct examples G where the the $\text{PoS}_{\text{IS}}(G)$ is arbitrarily close to 2. □

Related to PoS is the price of anarchy (PoA), which gauges the potential loss of utility caused by agents' selfish behavior; PoA is defined as the ratio of the highest-utility partition over the lowest-utility stable partition. Since the partition into singletons is vacuously internally stable, we have $\text{PoA}_{\text{IS}}(G) = \infty$ for any game where singletons derive zero utility.

6.6 Conclusions and Open Questions

We have extended the notion of internal stability for coalitions [1, 20] to a partition stability notion. Internal stability is important whenever agents have a local view of their preferences. While communication channels such as Slack have increased workers' abilities to interact with distant co-workers, the workers may still defect from the larger group and form private sub-channels. Thus, in such myopic situations, internal stability seems a more realistic measure of the sustainability of a work group assignment. The relationship of internal stability to other, more commonly investigated notions turns out to depend on the particularity of preference representation used in the hedonic game. For instance, any individually rational hedonic game has an internally stable partition of singletons; finding others is expected to be more difficult computationally.

We showed that internal stability is, in some hedonic game types, not equivalent to Nash or core stability. We have seen instances where the price of stability for internal stability is bounded, and where it is unbounded. We predict that this internal stability will continue to generate interesting insight into locally-aware coalition formation games.

Our research leaves some questions open for later research. One question is how hard it is to find a partition of maximal internally stable coalitions; note that such a partition is, by definition, internally stable. How would this problem be related, if at all, to the max-clique problem, wonderful stability, or strict popularity? Could fast heuristics be used to provide additional insight into these questions? What are the necessary and sufficient conditions to guarantee the existence of non-singleton internally stable partitions? How does internal stability compare to local core stability, proposed by Carosi et al. for SS-FHGs, if local core stability were adapted to games besides SS-FHGs?

Taywade et al. used the concept of internal stability in their proposal of budding in decentralized approach to hedonic games [64]. The algorithms implemented by Taywade et al. in 2018 and Taywade et al. in 2020 may provide good starting points to construct heuristics to find internally stable partitions in several classes of hedonic games [64, 65].

Chapter 7 Conclusion

In this work, we introduced two novel classes of hedonic games and a new notion of stability for hedonic games. We have demonstrated that hedonic games have a wide range of applications and can model myriad social phenomena. While we have answered a number of questions in the space of hedonic games, there are many questions that remain unanswered and even questions that have yet to be properly formalized.

Many problems in the hedonic games space are computationally hard to solve. Fortunately, many of these hard problems can be solved by heuristics; something we made particularly clear with our work on Anchored Team Formation Games. Still, many questions in the field remain unanswered. For example, can we create good heuristics to investigate internal stability? If so, how will they vary from one class of hedonic game to another? We've demonstrated that internal stability is distinct from both Nash and core stability for ATFGs, but are there other stability notions that exist along this spectrum of local to global scope? If so, are these notions continuous in nature, or are they discrete?

We could adapt ideas from other research areas into the hedonic games space to further our understanding of this vast field of research. For example, we could adapt the voting theory notions of possible and necessary winners, which are used to handle voters with unknown preferences, into possibly and necessarily stable partitions to handle agents with unknown preferences in hedonic games. Accommodating for agents with unknown preferences will make it possible to construct more robust models of real world behavior of humans and artificial agents. And that is, after all, the goal of this work.

Bibliography

[1] José Alcalde and Antonio Romero-Medina. Coalition formation and stability. *Social Choice and Welfare*, 27(2):365–375, 2006.

[2] Haris Aziz, Felix Brandt, and Hans Georg Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 195:316 – 334, 2013.

[3] Haris Aziz, Felix Brandt, and Paul Harrenstein. Fractional hedonic games. In *AAMAS*, pages 5–12. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[4] Harris Aziz and Rahul Savani. *Handbook of computational social choice*, chapter 15, pages 356–376. Cambridge University Press, 2016.

[5] Coralio Ballester. NP-completeness in hedonic games. *Games and Economic Behavior*, 49(1):1 – 30, 2004.

[6] Suryapratim Banerjee, Hideo Konishi, and Tayfun Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.

[7] Nathanaël Barrot, Kazunori Ota, Yuko Sakurai, and Makoto Yokoo. Unknown agents in friends oriented hedonic games: Stability and complexity. In *AAAI*, 2019.

[8] Jean-Claude Bermond, Augustin Chaintreau, Guillaume Ducoffe, and Dorian Mazauric. How long does it take for all users in a social network to choose their communities? *Discrete Applied Mathematics*, 270:37–57, 2019.

[9] Anna Bogomolnaia and Matthew O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.

[10] Felix Brandt, Martin Bullinger, and Anaëlle Wilczynski. Reaching individually stable coalition structures in hedonic games. In *AAAI*, 2021.

[11] Simina Brânzei and Kate Larson. Social distance games. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[12] Raffaello Carosi, Gianpiero Monaco, and Luca Moscardelli. Local core stability in simple symmetric fractional hedonic games. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 574–582, 2019.

[13] Katarina Cechlárová and Jana Hajduková. Computational complexity of stable partitions with b-preferences. *International Journal of Game Theory*, 31(3): 353–364, 2003.

[14] Katarina Cechlárová and Jana Hajduková. Stable partitions with w-preferences. *Discrete Applied Mathematics*, 138(3):333–347, 2004.

[15] Katarína Cechlárová and Antonio Romero-Medina. Stability in coalition formation games. *International Journal of Game Theory*, 29(4):487–494, 2001.

[16] Andrew J Collins, Sheida Etemadidavan, and Wael Khallouli. Generating empirical core size distributions of hedonic games using a monte carlo method. *arXiv preprint arXiv:2007.12127*, 2020.

[17] Thomas H Cormen, Charles E Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2009.

[18] Andreas Darmann, Edith Elkind, Sascha Kurz, Jérôme Lang, Joachim Schauer, and Gerhard Woeginger. Group activity selection problem. In *International Workshop on Internet and Network Economics*, pages 156–169. Springer, 2012.

[19] Gabrielle Demange. Intermediate preferences and stable coalition structures. *Journal of Mathematical Economics*, 23(1):45–58, 1994.

[20] Dinko Dimitrov, Peter Borm, Ruud Hendrickx, and Shao Chin Sung. Simple priorities and core stability in hedonic games. *Social Choice and Welfare*, 26(2): 421–433, 2006.

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

[22] Edith Elkind and Jörg Rothe. *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 3. Springer Texts in Business and Economics, 2016.

[23] Kimmo Eriksson and Olle Häggström. Instability of matchings in decentralized markets with various preference structures. *International Journal of Game Theory*, 36(3-4):409–420, 2008.

[24] Hendrik Fichtenberger and Anja Rey. Testing stability properties in graphical hedonic games. *arXiv preprint arXiv:1812.09249*, 2018.

[25] Michele Flammini, Bojana Kodric, and Giovanna Varricchio. Strategyproof mechanisms for friends and enemies games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1950–1957, 2020.

[26] Martin Gairing and Rahul Savani. Computing stable outcomes in symmetric additively separable hedonic games. *Mathematics of Operations Research*, 44(3): 1101–1121, 2019.

[27] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[28] Joseph Greenberg and Shlomo Weber. Strong tiebout equilibrium under restricted preferences domain. *Journal of Economic Theory*, 38(1):101–117, 1986.

[29] Joseph Greenberg and Shlomo Weber. Stable coalition structures with a unidimensional set of alternatives. *Journal of Economic Theory*, 60(1):62–82, 1993.

[30] Roger Guesnerie, Claude Oddou, et al. *Second best taxation as a game.* Centre d'études prospectives d'économie mathématique appliquées à la planification, 1979.

[31] Lane A. Hemachandra. The strong exponential hierarchy collapses. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 110–122, 1987.

[32] David S Johnson, Christos H Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of computer and system sciences*, 37(1):79–100, 1988.

[33] Helena Keinänen. Local search algorithms for core checking in hedonic coalition games. In *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems: First International Conference, ICCCI 2009, Wroclaw, Poland, October 5-7, 2009, Proceedings*, volume 5796, page 51. Springer Science & Business Media, 2009.

[34] Helena Keinänen. Simulated annealing for multi-agent coalition formation. In *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, pages 30–39. Springer, 2009.

[35] Helena Keinänen. An algorithm for generating Nash stable coalition structures in hedonic games. In *International Symposium on Foundations of Information and Knowledge Systems*, pages 25–39. Springer, 2010.

[36] Helena Keinänen. Algorithms for coalitional games. Master's thesis, Turku School of Economics Ae-2: 2011, 2011.

[37] Anna Maria Kerkmann and Jörg Rothe. Altruism in coalition formation games. In *Proc. International Joint Conferences on Artificial Intelligence Organization.* IJCAI, 2020.

[38] Anna Maria Kerkmann, Jérôme Lang, Anja Rey, Jörg Rothe, Hilmar Schadrack, and Lena Schend. Hedonic games with ordinal preferences and thresholds. *Journal of Artificial Intelligence Research*, 67:705–756, 2020.

[39] Jérôme Lang, Anja Rey, Jörg Rothe, Hilmar Schadrack, and Lena Schend. Representing and solving hedonic games with ordinal preferences and thresholds. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1229–1237. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

[40] Julien Lesca, Patrice Perny, and Makoto Yokoo. Coalition structure generation and CS-core: Results on the tractability frontier for games represented by MC-nets. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, pages 308–316. International Foundation for Autonomous Agents and Multiagent Systems, 2017.

[41] Yicheng Liu, Pingzhong Tang, and Wenyi Fang. Internally stable matchings and exchanges. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[42] Luke Harold Miles. A simulator for hedonic games. *arXiv preprint arXiv:1706.08501*, 2017.

[43] Thayer Morrill. The roommates problem revisited. *Journal of Economic Theory*, 145(5):1739–1756, 2010.

[44] Nhan-Tam Nguyen and Jörg Rothe. Local fairness in hedonic games via individual threshold coalitions. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 232–241. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[45] Nhan-Tam Nguyen, Anja Rey, Lisa Rey, Jörg Rothe, and Lena Schend. Altruistic hedonic games. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 251–259. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[46] Kazunori Ohta, Nathanaël Barrot, Anisse Ismaili, Yuko Sakurai, and Makoto Yokoo. Core stability in hedonic games among friends and enemies: Impact of neutrals. In *Proceedings of the 2017 International Joint Conferences on Artificial Intelligence*, pages 359–365, 2017.

[47] C. H. Papadimitriou and M. Yannakakis. The complexity of facets (and some facets of complexity). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 255–260, New York, NY, USA, 1982. ACM.

[48] Christos H Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

[49] Dominik Peters. Precise complexity of the core in dichotomous and additive hedonic games. In *International Conference on Algorithmic DecisionTheory*, pages 214–227. Springer, 2017.

[50] Anja Rey. *Beyond Intractability: A Computational Complexity Analysis of Various Types of Influence and Stability in Cooperative Games.* PhD thesis, Heinrich-Heine-Universität Düsseldorf, 2016.

[51] Anja Rey, Jörg Rothe, Hilmar Schadrack, and Lena Schend. Toward the complexity of the existence of wonderfully stable partitions and strictly core stable coalition structures in enemy-oriented hedonic games. *Annals of Mathematics and Artificial Intelligence*, 77(3):317–333, 2016.

[52] Alvin E Roth. The evolution of the labor market for medical interns and residents: a case study in game theory. *Journal of political Economy*, 92(6):991–1016, 1984.

[53] Alvin E Roth and Xiaolin Xing. Turnaround time and bottlenecks in market clearing: Decentralized matching in the market for clinical psychologists. *Journal of political Economy*, 105(2):284–329, 1997.

[54] Jacob Schlueter and Judy Goldsmith. Internal stability in hedonic games. In *FLAIRS*, 2020.

[55] Jacob Schlueter and Judy Goldsmith. Super altruistic hedonic games. In *FLAIRS*, 2020.

[56] Cory Siler. Tiered coalition formation games. In *The Thirtieth International Flairs Conference*, 2017.

[57] Michael Sipser. *Introduction to the Theory of Computation*, volume 2. Thomson Course Technology Boston, 2006.

[58] Jakub Sliwinski and Yair Zick. Learning hedonic games. In *Proceedings of the 2017 International Joint Conferences on Artificial Intelligence*, 2017.

[59] Matthew Spradling and Judy Goldsmith. Stability in role based hedonic games. In *The Twenty-Eighth International FLAIRS Conference*, 2015.

[60] Matthew Spradling, Judy Goldsmith, Xudong Liu, Chandrima Dadi, and Zhiyu Li. Roles and teams hedonic game. In *International Conference on Algorithmic Decision Theory*, pages 351–362. Springer, 2013.

[61] Matthew Jordan Spradling. Optimizing expected utility and stability in role based hedonic games. In *The Thirtieth International Flairs Conference*, 2017.

[62] Shao Chin Sung and Dinko Dimitrov. On core membership testing for hedonic coalition formation games. *Operations Research Letters*, 35(2):155 – 158, 2007.

[63] Shao-Chin Sung and Dinko Dimitrov. Computational complexity in additive hedonic games. *European Journal of Operational Research*, 203(3):635–639, 2010.

[64] Kshitija Taywade, Judy Goldsmith, and Brent Harrison. Decentralized multiagent approach for hedonic games. In *16th European Conference on Multi-Agent Systems*, 2018.

[65] Kshitija Taywade, Judy Goldsmith, and Brent Harrison. Decentralized marriage models. In *The Thirty-Third International Flairs Conference*, 2020.

[66] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[67] Naftali Waxman, Sarit Kraus, and Noam Hazon. On maximizing egalitarian value in k-coalitional hedonic games. *arXiv preprint arXiv:2001.10772*, 2020.

[68] Gerhard J Woeginger. Core stability in hedonic coalition formation. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 33–50. Springer, 2013.

Vita

Name

Jacob Schlueter

Education

- 2021 Ph.D. in Computer Science, University of Kentucky Lexington, Kentucky
- 2014 B.S. in Computer Science and Japanese Language and Literature, University of Kentucky Lexington, Kentucky
- 2014 M.B.A. University of Kentucky Lexington, Kentucky

Awards and Honors

- 2021 Invited to Daghstuhl Seminar 201241
- 2018–2019 Lexmark Fellowship, University of Kentucky
- 2016–2017 Graduate Fellowship, University of Kentucky
- 2014 Academic Excellence Scholarship, University of Kentucky
- 2013–2014 Stuckert Scholarship, University of Kentucky
- 2011–2013 Ralph G. Anderson Scholarship, University of Kentucky
- 2009–2013 Presidential Scholarship, University of Kentucky
- 2010–2011 Engineering Scholarship, University of Kentucky
- 2009–2010 Core Engineering Scholarship, University of Kentucky
- 2008 Eagle Scout

Professional Experience

- Starting August 2021 Postdoctoral Researcher,
- January 2019 – May2021 Graduate Teaching Assistant, University of Kentucky Lexington, Kentucky
- May 2020 – June 2020 Graduate Teaching Assistant, University of Kentucky Lexington, Kentucky
- May 2019 – June 2019 Graduate Teaching Assistant, University of Kentucky Lexington, Kentucky

- August 2018 – May 2019 Graduate Research Assistant, University of Kentucky Lexington, Kentucky

- September 2018 – November 2018 Visiting Research Intern, National University of Singapore, Singapore

- August 2017 – May 2018 Graduate Teaching Assistant, University of Kentucky Lexington, Kentucky

- April 2015 – July 2016 Associate Programmer, Selman & Company Cleveland, Ohio

- August 2014 – December 2014 Undergraduate Teaching Assistant, University of Kentucky Lexington, Kentucky

- January 2012 – March 2012 Software Testing Intern, Archvision Lexington, Kentucky

Publications

- Jacob Schlueter, Christian Addington, and Judy Goldsmith.“Anchored Team Formation Games” Proc. FLAIRS 2021, May 2021.

- Jacob Schlueter and Judy Goldsmith.“Internal Stability in Hedonic Games” Proc. FLAIRS 2020, May 2020.

- Jacob Schlueter and Judy Goldsmith. “Super Altruistic Hedonic Games” Proc. FLAIRS 2020, May 2020.

- Jacob Schlueter. “Novel Hedonic Games and Lottery Systems” Proceedings of the AAMAS 2019, May 2019.

Supervised Students

- Christian Addington March 2020 – Present

Service

- Subreviewer for the Computational Social Choice Workshop 2021.

- Reviewer for the Games Agents and Incentives Workshop 2021.

- Reviewer for the Association for the Advancement of Artificial Intelligence Conference 2021.

- Subreviewer for the Multidisciplinary Workshop on Advances in Preference Handling 2020.

- Subreviewer for the Transactions on Autonomous and Adaptive Systems Journal 2020.

- Subreviewer for the Artificial Intelligence, Ethics, and Society Conference 2020.
- Subreviewer for the Algorithmic Decision Theory Conference 2019.
- Subreviewer for the Computational Social Choice Workshop 2018.