Roles and Teams Hedonic Game

Matthew Spradling, Judy Goldsmith, Xudong Liu, Chandrima Dadi, Zhiyu Li
University of Kentucky, USA
mjspra2@uky.edu, goldsmit@cs.uky.edu, liu@cs.uky.edu, cda232@g.uky.edu, zhiyu.li@uky.edu

Abstract

We introduce a new variant of hedonic coalition formation games in which agents have two levels of preference on their own coalitions: preference on the set of "roles" that makes up the coalition, and preference on their own role within the coalition. We define several stability notions and optimization problems for this model. We prove the hardness of the decision problems related to our optimization criteria. We introduce a heuristic optimizer for coalition formation in this setting and evaluate it with respect to to brute-force MinSum and MinMax solvers.

1 Introduction

Consider the online game, League of Legends, developed by Riot Games, Inc. According to a recent market research study, League of Legends is the most played PC video game in North America and Europe by number of hours played per month [DFC, 2012], with 70 million registered users and an average of 12 million daily active players [Riot, 2012]. Players sign on, and are matched with other players with similar Elo ratings. Once matched in a team of 3 or 5, they each choose champions from a finite set. The game experience could be enhanced if teams were matched on the basis of strategic combinations of champions.

Matching players by their preferences on their own teams is a hedonic coalition formation game [Drèze and Greenberg, 1980]. Hedonic coalition formation games are characterized by agents' utilities depending only on the coalition they are assigned to, not on others. A game consists of a set of agents and their preferences for their possible roles and team compositions.

One of the aspects of the partitioning problem for League of Legends is the two-stage team formation: Players may be matched based on their shared interest in a team consisting of roles A, B, and C, but it may transpire that all three wish to play role A. A better partition algorithm would also use players' preferences on individual roles. We refer to this notion of a hedonic game as a *Roles and Teams Hedonic Game* (RTHG).

Recent work on hedonic coalition games has touched on notions comparable to stability in the stable marriage problem [Drèze and Greenberg, 1980; Banerjee *et al.*, 2001a; Bogomolnaia and Jackson, 2002; Pini *et al.*, 2011; Aziz and Brandl, 2012], etc. It is known that finding certain stable coalitions for hedonic games is NP-hard (see, for instance, [Conitzer and Sandholm, 2006; Aziz *et al.*, 2011]). Some papers considered restrictions on preferences that allow stable partitions, others presented heuristic algorithms for finding stable partitions.

Due to the two-stage team formation procedure in RTHG, we observe that the notions of Nash stable (NS) and individually stable (IS) partitions are quite different in this model compared to other hedonic games. We propose definitions for NS and IS partitions which address both the stability of role assignments within coalitions and permutations of agents within coalition assignments.

A different problem of optimizing social utility has also been investigated. In graphical games with unbounded treewidth, very recent work has been done to address the bicriteria problem of maximizing both stability and social utility [Ismaili *et al.*, 2013]. We provide hardness results for the decision problems related to Perfect, MinSum and MinMax partitions in RTHG. We introduce a quadratic time greedy heuristic optimizer for coalition formation and compare to brute-force MinSum and MinMax solvers.

2 Roles and Teams Hedonic Games

Definition 2.1 *An 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 team member 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* $\langle u_0, \ldots, u_{|P|-1}\rangle$*, where for each agent* $p \in P$*, composition* $t \in C$*, and role* $r \in R$ *there is a utility function* $u_p(t, r)$ *with* $u_p(t, r) = \infty$ *if* $r \notin t$*.*

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

In this paper, we consider lower U values to be preferable, so our optimization problems are minimizations.

Table 1: Example RTHG instance with $|P| = 4, m = 2, |R| = 2$

$\langle r,t\rangle$	$u_{p_0}(r,t)$	$u_{p_1}(r,t)$	$u_{p_2}(r,t)$	$u_{p_3}(r,t)$
$\langle A, AA\rangle$	2	2	0	0
$\langle A, AB\rangle$	0	3	2	2
$\langle B, AB\rangle$	3	0	3	3
$\langle B, BB\rangle$	1	1	1	1

3 Related Work: Hedonic Partition Games

The original motivation for studying hedonic games was economic [Drèze and Greenberg, 1980], but there are also many computational applications. Saad, et al. have proposed hedonic coalition formation game models for a variety of multi-agent settings, including distributed task allocation in wireless agents [Saad *et al.*, 2011a], communications networks [Saad *et al.*, 2010], and vehicular networks [Saad *et al.*, 2011b], among others.

In anonymous hedonic games [Banerjee *et al.*, 2001b], agents have preferences over group size and are matched to teams for a single type of activity. The group activity selection problem (GASP) includes preferences over a variety of activities given the number of agents engaged in the activity [Darmann *et al.*, 2012]. Agents in these games are homogeneous—every member of a coalition is equivalent. In RTHG, agents are heterogeneous while team size and group activity are fixed for a given instance. An RTHG agent holds preferences over its own role and the roles of its teammates. Furthermore, while GASP preferences are binary, RTHG agent preferences are not guaranteed to be.

Desirable partitioning in additively separable hedonic games (ASHG) [Aziz *et al.*, 2012] has been investigated. ASHGs allow for agents to place values on each other, making the agent population heterogeneous. The value an agent places on its coalition in such a game is the sum total value it gives other agents in its coalition. This model considers agent-to-agent valuation, but these values are fixed for any given agent-to-agent relation. ASHGs do not consider the context of the composition an agent is in. In RTHG, values are placed on team compositions and roles rather than individual agents.

Each agent has a variable role in RTHG and has preferences over which role to select for itself given a team composition.

For instances where $|C|m$ is smaller than $|P|$, the required input data for RTHG instances will be smaller than the required input for ASHG. Input for an ASHG instance requires each agent to hold a specific utility for each other agent within the population. This could be represented as a $|P| \times |P|$ matrix of utility values, U, where $U[i, j]$ is the utility that p_i holds for p_j. In RTHG, the input can be represented as a $|C|m \times |P|$ matrix. While there are millions of players in League of Legends [Riot, 2012], there are only around 10 basic roles to potentially fill (Healer, Mage, Assassin, etc.) and a maximum team size of 5. The input required for team formation in this setting will be orders of magnitude smaller in RTHG than if this game were treated as an ASHG.

Consider the following setting. In capstone computer science courses, students are sometimes grouped into equally-sized project teams. For a team of five students, one student may prefer a team of 2 skilled programmers, 1 designer, and 2 writers. Her second choice might be 1 programmer, 2 designers, 2 writers. In the first case, the student wants to be a programmer. In the second, she wants to be a designer, and definitely *not* a programmer.

This problem can be modeled as an RTHG. The GASP model does not apply. The ASHG model allows students to express utility values for each other, but ASHG preferences are *context-free* agent-to-agent assessments. John may wish to join Linda's coalition when she needs a programmer, but not when she needs a writer. In RTHG, an agent need only express preferences on which roles and compositions she prefers. This self-evaluation may be easier to accurately poll.

Matching students to groups in a manner that optimizes utility for the class would be a useful endeavor. In a perfect world, each student would be matched to his or her most-preferred team. We show that such a *perfect* partition is not always possible in RTHG.

A *MinSum* partition would, in a utilitarian fashion, optimize the sum total utility of the resulting coalitions. A *MinMax* partition would take an egalitarian approach. It is unclear which metric (MinSum or MinMax) will best raise teaching evaluations in capstone computer science courses.

4 Evaluation of Solutions

A perfect partition for general hedonic games has been defined as one such that each agent is in one of her most preferred coalitions [Aziz and Brandl, 2012].

For RTHG, we define a perfect partition to be one in which each agent gets a most-preferred coalition composition and role within that composition. Note that, in the general RTHG model, there may be multiple equivalently-valued coalitions and roles. Therefore these preferences are not necessarily strict.

Definition 4.1 *A* perfect partition *is a partition of agents to coalitions so that, for each $p \in P$:*

- *the assigned coalition is preferred to all other possible coalition compositions, and*
- *the assigned role in the coalition is preferred to all other roles within that coalition.*

We show that a perfect partition is not always possible.

Observation 4.2 *A perfect partition is impossible for some RTHG instances.*

Proof. Let $m = 2$ and $P = \{\text{Alice}, \text{Bob}\}$. Both Alice and Bob strictly prefer the team composition of Mage and Assassin to all others. However, both Alice and Bob strictly prefer to play Assassin over Mage in that team composition. Therefore, no partition will be a perfect partition. □

Because a perfect partition might not exist, we consider finding a good solution to an RTHG instance by generating a partition such that either the total payoff of all agents in the population is maximized (MinSum partition) or such that the minimum total payoff for a single team is maximized (MinMax partition).

Definition 4.3 *Given an instance I of RTHG, a* MinSum partition *is one that achieves the minimum value of* $\Sigma_{i<|P|}u_{p_i}$.

Definition 4.4 *Given an instance I of RTHG, a* MinMax partition *is one that achieves the minimum value of* $\max_{p\in P} u_p$.

In most hedonic game variants, a partition is considered *Nash stable (NS)* iff no agent p_i can benefit by moving from her coalition to another (possibly empty) coalition T. A partition is considered *individually stable (IS)* iff no agent can benefit by moving to another coalition T while not making the members of T worse off [Aziz and Brandl, 2012]. These definitions of stability do not fit well with RTHG.

Because team sizes in RTHG are fixed at m, an agent cannot simply choose to leave her coalition and join another. Rather, if an agent p_i is to move from coalition S to T, she must take the place (role in a particular coalition) of another agent p_j in T. This could be done as a swap, or it could be a more complex set of moves made among several agents. For example: Consider the case where Alice prefers Bob's (current) place, while Bob prefers Clover's place, while Clover prefers Alice's place. No pair of agents can swap to improve utility, but the three could move in concert. Note that should some $S \subseteq P$ collaboratively change positions, this permutation would not change the utilities of the compositions for the agents in $\overline{S}$. All existing compositions remain intact.

Another option in RTHG is for an agent to remain within her coalition but change roles. This converts the existing composition to another the agent may prefer. Note that this would change the utility of the composition for her coalition, but otherwise does not affect the utility of the partition for any agent outside of her coalition.

We define NS and IS in terms of role swaps and team swaps, as follows:

Definition 4.5 *A partition π is* Nash role stable (NRS) *iff no agent p_i can improve her utility by changing from her current role r to a new role r'.*

Definition 4.6 *A partition π is* individually role stable (IRS) *iff no agent p_i can improve her utility by changing from her current role r to a new role r' without reducing the utility of any other agent in her coalition.*

Definition 4.7 *A partition π is* Nash team stable (NTS) *iff no set $S \subseteq P$ of agents can improve the sum of their utilities by a new permutation of their positions in their coalitions.*

Definition 4.8 *A partition π is* individually team stable (ITS) *iff no set $S \subseteq P$ of agents can improve the sum of their utilities by a new permutation of their positions in their coalitions without reducing the utility of the partition for any single agent in S.*

Definition 4.9 *A partition π is* Nash stable (NS) *iff it is both NRS and NTS.*

Definition 4.10 *A partition π is* individually stable (IS) *iff it is both IRS and ITS.*

5 Complexity

Definition 5.1 *An instance of Special RTHG is an instance of RTHG such that*

- *for each agent $p \in P$, $u_{C,p} : C \to \{0,1\}$ and a mapping $u_{R,p} : R \to \{0,1\}$;*
- *for each agent $p \in P$, $u_{C,p} \not\equiv 1$; for each agent $p \in P$, each $c \in C$, if $u_{C,p}(c) = 0$ then c is* uniform, *namely consists of m copies of a single role r, and $u_{R,p}(r) = 0$.*

In other words, each agent finds some non-empty set of single-role team compositions acceptable (utility 0), and no other types of team compositions acceptable.

Definition 5.2 *The language* PERFECT RTHG *consists of those instances of RTHG for which a perfect partition exists, and* PERFECT SPECIAL RTHG *consists of those instances of Special RTHG for which a perfect partition exists.*

If we look at Special RTHG instances, then finding the MinMax optimal partition tells us whether there's a perfect matching (everyone gets an acceptable assignment). Thus, the question of a perfect matching reduces to the problem of finding a MinMax partition, or the decision problem of whether there's an assignment with MinMax value 0.

Consider the EXACT COVER problem:
GIVEN a set $S \subseteq \mathcal{P}(\{1, ..., r\})$ where all elements of S have size 3,
IS THERE a subset $T \subseteq S$ such that T partitions $\{1, ..., r\}$?

In other words, such that the union over sets in $T = \{1, ..., r\}$ and the intersection of any two distinct sets in T is empty. Note that r must be divisible by 3.

EXACT COVER is NP-complete [Goldreich, 2008].

Theorem 5.3 PERFECT SPECIAL RTHG *is NP-complete.*

Proof. To show that PERFECT SPECIAL RTHG is in NP, consider the following NP algorithm. Given an instance of PERFECT SPECIAL RTHG, guess a partition and evaluate its MinMax value. To compute the MinMax value, compute the utility of each of the $|P|/m$ coalitions (time $\mathcal{O}(m * t)$ for each coalition, where t is the complexity of table lookup for an individual's utility for a particular team and role), stopping and rejecting if any coalition has utility 1, else accepting. This checking is in time polynomial in the size of the input.

To show NP-hardness, we show that EXACT COVER $\leq_m^P$ SPECIAL PERFECT RTHG. In other words, given an instance $E = \langle r, S\rangle$ of EXACT COVER, we construct an instance R_E of SPECIAL PERFECT RTHG such that $E \in$ EXACT COVER iff $R_E \in$ SPECIAL PERFECT RTHG.

R_E will have the property that, for each agent, the only acceptable teams are uniform, i.e., consist of m copies of a single role. Thus, the question is whether they can be assigned to an acceptable team; the role for that team will be acceptable.

Consider $E = \langle r, S\rangle$. R_E will have one team composition for each set in S, and one agent for each of $\{1, ..., r\}$. The desired team size is $m = 3$. Each agent i desires those team compositions s such that $i \in s$.

There is an exact cover of $\{1, ..., r\}$ iff there is an assignment of agents to teams of size 3 such that each team corresponds to an element of S.

Therefore, the PERFECT SPECIAL RTHG problem is NP-hard. □

Corollary 5.4 *The general case of* PERFECT RTHG *is NP-hard.*

Proof. We observe that if there were a fast algorithm to decide the general case of PERFECT RTHG then this same algorithm would also decide PERFECT SPECIAL RTHG.

Therefore the general case of PERFECT RTHG is NP-hard. □

Definition 5.5 *The language* MINSUM RTHG *consists of pairs $\langle G, k\rangle$, where G is an instance of RTHG, k is an integer, and the MinSum value of G is $\leq k$;* MINSUM SPECIAL RTHG *consists of those instances of Special RTHG for which the MinSum value is* 0.

Definition 5.6 *The language* MINMAX RTHG *consists of pairs $\langle G, k\rangle$, where G is an instance of RTHG, k is an integer, and the MinMax value is $\leq k$;* MINMAX SPECIAL RTHG *consists of those instances of Special RTHG for which the MinMax value is* 0.

Theorem 5.7 MINMAX RTHG *and* MINSUM RTHG *are both NP-hard.*

Proof. A Special RTHG partition π for G is perfect iff $\sum_{p \in P} u_p(\pi) = 0$ iff MinMax(π) = 0 iff $\langle G, 0\rangle \in$ MINSUM RTHG iff MinSum(π) = 0 iff $\langle G, 0\rangle \in$ MINMAX RTHG. Therefore MINMAX RTHG and MINSUM RTHG are both NP-hard. □

6 Greedy Heuristic Partitioning

By modeling agents as voters in an election and their preferences over team compositions and roles as votes, the *scoring* voting rule can be applied to hold a series of elections and democratically (but not necessarily optimally) assign agents to teams. A *voting rule* is a function mapping a vector a of voters' votes to one of the b candidates in a candidate set c.

Definition 6.1 *[Conitzer and Sandholm, 2005] We define scoring rules for elections as follows. Let $a = \langle a_1, \cdots, a_m\rangle$ be a vector of integers such that $a_1 < a_2 < \ldots < a_m$. For each voter, a candidate receives a_1 points if it is ranked first by the voter, a_2 points if it is ranked second, etc. The score s_c of candidate c is the total number of points the candidate receives .*

For our procedure, a $|C|m \times |P|$ matrix of agent utility values becomes the candidate set c. An "election" is run upon the candidate set to select the most-preferred coalition. A set of m voters with the highest utility for that coalition is selected to form a team and removed from the population. Their votes are removed, and a new election is held on the reduced candidate set. This procedure continues until all $|P|$ agents have been matched to $|P|/m$ teams. We assume that m evenly divides $|P|$. The following pseudocode describes this greedy algorithm:

Observation 6.2 *The time complexity of GreedyRTHGPartiton is $O(|P|^2/m)$, or $O(|P| \cdot |C| \cdot m)$ if $|P| < |C| \cdot m^2$.*

Algorithm 1 GreedyRTHGPartiton(RTHG instance G, empty partition π)

```
for |C| compositions c_0 → c_{|C|-1} do
  for m positions r_0 → r_{m-1} ∈ c_i do
    calculate the sum of agent votes on ⟨c_i, r_j⟩. O(|P|)
  end for
end for
for |P|/m coalitions t_0 → t_{|P|/m-1} to assign to π do
  find the set of compositions C_min for which the sum of
  total votes is minimized. O(|C| · m)
  select one composition c_i uniformly at random from
  within the set.
  for m positions r_0 → r_{m-1} ∈ c_i do
    find the set of agents P_min(c_i, r_j) for whom the
    individual agent's vote for ⟨c_i, r_j⟩ is minimized.
    This takes time O(|P|/m), given that the population
    shrinks by m agents as each team is formed and re-
    moved.
    select one agent p_j uniformly at random from within
    the set.
    add agent p_j to the coalition t_k.
    for |C| compositions c_0 → c_{|C|-1} do
      for m positions r_0 → r_{m-1} ∈ c_i do
        remove agent p_j's vote from the population,
        decrementing the sum total vote on ⟨c_i, r_j⟩.
      end for
    end for
  end for
  append team t_k to the partition π.
end for
```

7 Testing and Results

For our experiments we chose *Strictly Ordered RTHG* instances. In a Strictly Ordered RTHG instance, each agent's first choice of composition or role is weighted equivalently to other agents' first choices, as is her second choice, etc. The system does not value one agent's preferences over another.

Forty-eight instances of Strictly Ordered RTHG were generated by a uniformly random procedure we developed. This number of cases allowed us to test $|P|$ ranging from 6 to 15 agents, $|R|$ ranging from 3 to 6, and m ranging from 3 to 5.

We began with $|P|$ = 6, $|R|$ = 3, and m = 3 in the minimal case. Two random preference matrices were generated with these arguments. We then incremented $|R|$ by 1 and generated two new random preference matrices, up to $|R|$ = 6. This process was repeated for $\langle m, |P|\rangle = \langle 4, 8\rangle, \langle 5, 10\rangle, \langle 3, 12\rangle, \langle 4, 12\rangle, \langle 5, 15\rangle$. The resulting 48 instances can be evaluated as $|R|$, $|P|$ and m increase. These upper bounds were chosen because larger values would have dramatically increased the time required for the brute force solvers to process the input.

Optimal results were calculated for each of these instances by MinSum and MinMax brute force implementations we developed. There are $O(|P|!/m!^{|P|/m} \cdot |P| \cdot |C| \cdot (m-1)!)$ possible partitions in an instance of RTHG. We generate all of them and find the MinSum and MinMax values for each instance considered. Our implementation of *GreedyRTHGPartiton* ran

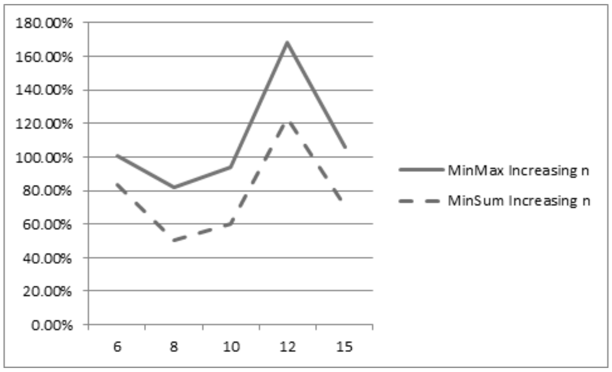

Figure 1: Percent overestimate of MinSum and MinMax using *GreedyRTHGPartiton* as $|P|$ increases.

each instance of Strictly Ordered RTHG 500 times, in order to limit random error. We compared the mean of those results to the optimal results.

Computations were run on a machine using 8 GB of RAM and a 2.50 GHz Intel(R) Core(TM) i5-3210M CPU. MinSum and MinMax brute force algorithms were implemented in C++, while *GreedyRTHGPartiton* was implemented in Python 3.3.

Results of *GreedyRTHGPartiton* are compared to MinSum and MinMax partitions in Figures 1, 2, and 3. We show the percent by which *GreedyRTHGPartiton* overestimates the actual MinSum and MinMax payoffs for each test case. The lower the percent overestimation the better. Each figure shows the mean overestimation for each run of the heuristic algorithm.

GreedyRTHGPartiton produces consistently better results for estimating MinSum compared to MinMax. This is not entirely surprising, given that the algorithm is at each iteration only concerned with approximating the current best candidate coalition. As an example: Suppose there are 6 agents A, B, C, D, E, and F being matched to 3 teams each of size 2. The best coalition is AB, while the four worst coalitions are CD, CE, CF, and EF. If A and B form a coalition together in the first iteration, then the remaining two coalitions selected will be among the worst possible. Indeed, it may transpire that CD is the next team to be formed (being the best among the final four), *even if EF happens to be the worst coalition of all*.

Because *GreedyRTHGPartiton* has no mechanism to look ahead and avoid such scenarios, it has weakened performance for approximating MinMax solutions. MinSum solutions are more closely estimated because, in the aggregate, weaker coalitions selected at the end are balanced out by strong selections made at the beginning. In our experiments, an average run of *GreedyRTHGPartiton* overestimates the MinSum partition's total utility by 85.22%. The MinMax partition's minimum coalition utility is overestimated by 119.79% on average. The rate of accuracy does not appear to decrease as $|P|$, $|R|$ or m increases. There is actually an improvement in accuracy as m increases.

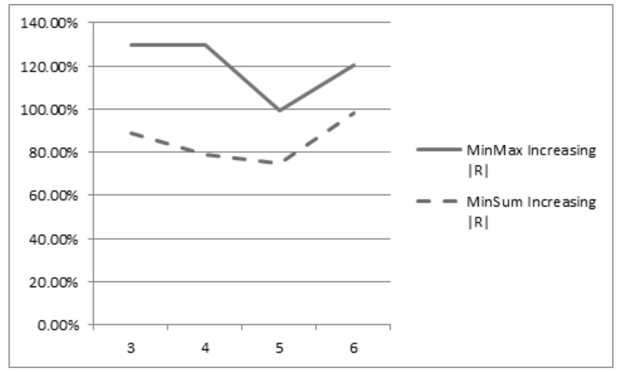

Figure 2: Percent overestimate of MinSum and MinMax using *GreedyRTHGPartiton* as $|R|$ increases.

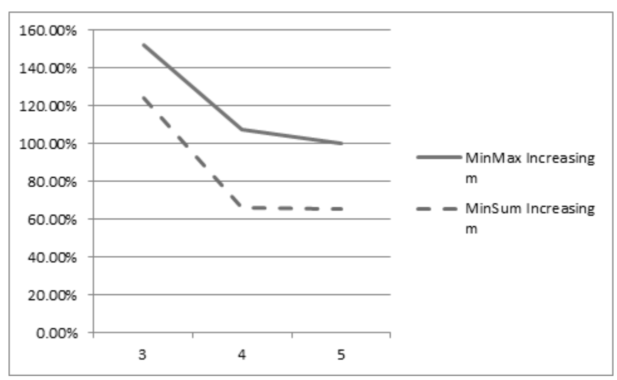

Figure 3: Percent overestimate of MinSum and MinMax using *GreedyRTHGPartiton* as m increases.

References

[Aziz and Brandl, 2012] Haris Aziz and Florian Brandl. Existence of stability in hedonic coalition formation games. In *Proc. Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012),*, 2012.

[Aziz *et al.*, 2011] Haris Aziz, Felix Brandt, and Paul Harrenstein. Pareto optimality in coalition formation. In G. Persiano, editor, *Proc. International Symposium on Algorithmic Game Theory (SAGT)*, Lecture Notes in Computer Science. Springer, 2011.

[Aziz *et al.*, 2012] Haris Aziz, Felix Brandt, and Hans Georg Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 2012.

[Banerjee *et al.*, 2001a] S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18:135–153, 2001.

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

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

[Conitzer and Sandholm, 2005] Vincent Conitzer and Tuomas Sandholm. Communication complexity of common voting rules. In *Proceedings of the 6th ACM conference on Electronic commerce*, EC '05, pages 78–87, New York, NY, USA, 2005. ACM.

[Conitzer and Sandholm, 2006] Vincent Conitzer and Tuomas Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170:607–619, 2006.

[Darmann *et al.*, 2012] Andreas Darmann, Edith Elkind, Sascha Kurz, Jérôme Lang, Joachim Schauer, and Gerhard Woeginger. Group activity selection problem. In *Workshop Notes of COMSOC-2012*, 2012.

[DFC, 2012] DFC. League of Legends Most Played PC Game. DFC Intelligence, 2012. `http://www.dfcint.com/wp/?p=343`.

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

[Goldreich, 2008] Oded Goldreich. *Computational Complexity, A Conceptual Perspective*. Cambridge University Press, 2008.

[Ismaili *et al.*, 2013] Anisse Ismaili, Evripidis Bampis, Nicolas Maudet, and Patrice Perny. A study on the stability and efficiency of graphical games with unbounded treewidth. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2013)*, 2013. To appear.

[Pini *et al.*, 2011] Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Stability in matching problems with weighted preferences. *Proc. ICAART11*, 2011.

[Riot, 2012] Riot. League of Legends' Growth Spells Bad News for Teemo — Riot Games. Riot Games Inc, 2012. `http://www.riotgames.com/articles/20121015/138/league-legends-growth-spells-bad-news-teemo`.

[Saad *et al.*, 2010] Walid Saad, Zhu Han, Tamer Basar, Are Hjorungnes, and Ju Bin Song. Hedonic coalition formation games for secondary base station cooperation in cognitive radio networks. In *Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2010.

[Saad *et al.*, 2011a] Walid Saad, Zhu Han, Tamer Basar, M. Debbah, and Are Hjorungnes. Hedonic coalition formation for distributed task allocation among wireless agents. In *Proc. IEEE Transactions on Mobile Computing*, 2011.

[Saad *et al.*, 2011b] Walid Saad, Zhu Han, Are Hjørungnes, Dusit Niyato, and Ekram Hossain. Coalition formation games for distributed cooperation among roadside units in vehicular networks. *IEEE Journal on Selected Areas in Communications (JSAC), Special issue on Vehicular Communications and Networks*, 2011.