Roles and Teams Hedonic Game

Matthew Spradling, Judy Goldsmith, Xudong Liu, Chandrima Dadi, and 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

University of Kentucky

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 and show easiness of finding individually stable partitions. We introduce a heuristic optimizer for coalition formation in this setting. We evaluate results of the heuristic optimizer and the results of local search for individually stable partitions with respect to brute-force MaxSum and MaxMin solvers.

Keywords: coalition formation, computational complexity, hedonic games, optimization

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 [10], with 70 million registered users and an average of 12 million daily active players [15]. 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 an avatar (called a "champion") from a finite set. Each team then plays against another team, competing for Elo improvement.

The game experience could be enhanced if teams were matched on the basis of strategic combinations of champions. This is not only our hypothesis but also the observation of Riot Games. A senior user research employee for Riot Games, user name *davin*, recently commented that *"we don't have a single way of playing the game. ... So when you match people together, you'd need some way of pairing together players who have agreed on a particular strategy or want to play in a certain way."*[16] There are two criteria upon which players might express preferences: the combination of champions on which they would like to play, and the individual champion they would prefer to play on a given composition. These could be expressed separately or conditionally.

Matching players by their preferences on their own teams is a hedonic coalition formation game [11]. 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 [11, 4, 6, 14, 1], etc. It is known that finding certain stable coalitions for hedonic games is NP-hard (see, for instance, [8, 2]). 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 bi-criteria problem of maximizing both stability and social utility [13]. We provide hardness results for the decision problems related to Perfect, MaxSum and MaxMin partitions in RTHG. We define Nash stability and individual stability in this setting and show that individually stable partitions can always be found in time polynomial in the size of the input. We introduce a quadratic time greedy heuristic optimizer for coalition formation and compare to brute-force MaxSum and MaxMin solvers and the results of local search for individually stable partitions.

2 Roles and Teams Hedonic Games

Definition 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.

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 [11], 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 [17], communications networks [18], and vehicular networks [19], among others.

In anonymous hedonic games [5], 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 [9]. 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) [3] 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 [15], 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. Huxley may wish to join Clover'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 *MaxSum* partition would, in a utilitarian fashion, optimize the sum total utility of the resulting coalitions. A *MaxMin* partition would take an egalitarian approach. It is unclear which metric (MaxSum or MaxMin) would best raise teaching evaluations in capstone computer science courses.

4 Evaluation of Solutions

Perfect partitions for general hedonic games have been defined such that each agent is in one of her most preferred coalitions [1].

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 compositions and roles. Therefore these preferences are not necessarily strict.

Definition 2. *A* perfect partition *is a partition of agents to coalitions so that, for each $p \in P$, $u_p(r, t) = min\{u_p(r, t) : \ r \in R \wedge t \in C\}$.*

A perfect partition is impossible for some RTHG instances. Consider an RTHG instance where $m = 2$ and $P = \{\text{Alice}, \text{Bob}\}$. Both Alice and Bob strictly prefer the team composition of $\langle$Mage, Assassin$\rangle$ with the role Assassin to all other $\langle r, t\rangle$ pairs. No perfect partition is possible.

We consider the following notions of utility optimization.

Definition 3. *Given an instance I of RTHG, a* MaxSum partition *is one that achieves the maximum value of $\Sigma_{i<|P|} u_{p_i}$.*

MaxSum is a utilitarian optimality criterion.

Definition 4. *Given an instance I of RTHG, a* MaxMin partition *is one that achieves the maximum value of* $\min_{p \in P} u_p$.

MaxMin is an egalitarian optimality criterion.

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 [1]. 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 *position* (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. Note that should some $X \subseteq P$ collaboratively change positions, this permutation would not change the utilities of the compositions for the agents in $\overline{X}$. All existing compositions remain intact.

Definition 5. *A partition π is* Nash team stable (NTS) *iff no set $X \subseteq P$ of agents can improve the sum of their utilities by a new permutation of their positions in their coalitions.*

A partition π is individually team stable (ITS) *iff no set $X \subseteq P$ 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 X.*

There will always be a NTS partition π_{NTS} where all agents select the same role. In this case, no agent can improve her utility by changing positions since the new position would be identical to her previous position. Some RTHG instances may lack a *non-uniform* NTS partition $\pi_{NTS_{NU}}$, where $r_{p_i} \neq r_{p_j}$ for at least one pair of agents p_i, p_j. Consider the following RTHG instance:

Table 2. RTHG instance with $|P| = 2, m = 2, |R| = 2$ where no $\pi_{NTS_{NU}}$ exists

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

No $\pi_{NTS_{NU}}$ exists in this instance. Consider each of the two possible non-uniform partitions:

- π_0, where $r_{p_0} = B$ and $r_{p_1} = A$. p_0 prefers to swap positions. Not NTS.

- π_1, where $r_{p_0} = A$ and $r_{p_1} = B$. p_1 prefers to swap positions. Not NTS.

To construct an individually team stable partition π_{ITS}, start with any partition π of I and iteratively improve it until no improvements are possible. At that point, the resulting partition will be ITS. To find an improvement, if one exists, construct a graph $G = \langle V, E_b \cup E_r \rangle$, where the vertices correspond to players, and there is an edge in E_b from p_i to p_j iff $u_{p_i}(r_i, t_i) = u_{p_i}(r_j, t_j)$, and there is an edge in E_r from p_i to p_j iff $u_{p_i}(r_i, t_i) > u_{p_i}(r_j, t_j)$. If there exists a cycle in the graph containing as least one edge $e_r \in E_r$, then the partition is not ITS.

Another movement 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.

Definition 6. *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'.*

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.*

Some RTHG instances may lack a NRS partition π_{NRS}. Consider the following RTHG instance:

Table 3. RTHG instance with $|P| = 2, m = 2, |R| = 2$ where no π_{NRS} exists

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

No π_{NRS} exists in this instance. Consider each of the four possible partitions:

- π_0, where $r_{p_0} = A$ and $r_{p_1} = B$. p_1 prefers to switch to role A. Not NRS.
- π_1, where $r_{p_0} = B$ and $r_{p_1} = A$. p_1 prefers to switch to role B. Not NRS.
- π_2, where $r_{p_0} = A$ and $r_{p_1} = A$. p_0 prefers to switch to role B. Not NRS.
- π_3, where $r_{p_0} = B$ and $r_{p_1} = B$. p_0 prefers to switch to role A. Not NRS.

An IRS partition π_{IRS} of an RTHG instance I can be found in time polynomial in $|I|$. Given any partition π of I, perform a local search where the neighborhood is one individual in one coalition changing her role and improvement is evaluated in terms of changes to that coalition's utility. Since this improves the overall utility of the partition, there are a limited number of possible improvements; when no improvements are possible, the partition is IRS.

Definition 7. *A partition π is* Nash stable (NS) *iff it is both NTS and NRS.*

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

A NS partition π_{NS} may not always exist for some RTHG instances, given that a partition π_{NRS} may not exist.

An IS partition π_{IS} of an RTHG instance I can be found in time polynomial in $|I|$. Given any partition π of I, alternatively perform IRS local search and ITS local search until neither finds an improvement. The resulting partition π' will be IS.

Theorem 1. *Every instance of RTHG has an IS partition. Not every instance of RTHG has a NS partition.*

5 Complexity

Definition 8. *An instance of Special RTHG is an instance of RTHG such that for each agent $p \in P$, each $t \in C$, and each $r \in t$; $u_p(t, r) \to \{0, 1\}$ and $u_p(t, r) = 1$ only if t is* uniform, *namely it consists of m copies of a single role r.*

In other words, each agent finds some non-empty set of single-role team compositions acceptable (utility 1), and no other types of team compositions acceptable.

Definition 9. *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.*

In Special RTHG instances, the question of a perfect partition reduces to the problem of finding a MaxMin partition, or the decision problem of whether there's a partition with MaxMin value m.

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 $\{1, ..., r\}$?

Exact Cover is NP-complete [12].

Theorem 2. 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 MaxMin value. To compute the MaxMin value, compute the utility of each of the $|P|/m$ coalitions (time $\mathcal{O}(mt)$ 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 0, 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 $E = \langle r, S \rangle$. For each set in S, R_E will have a role and a corresponding team composition. $P = \{1, ..., r\}$. 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 1. *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 10. *The language* MaxSum RTHG *consists of pairs $\langle G, k \rangle$, where G is an instance of RTHG, k is an integer, and the MaxSum value of G is $\leq k$;* MaxSum Special RTHG *consists of those instances of Special RTHG for which the MaxSum value is $|P|$.*

Definition 11. *The language* MaxMin RTHG *consists of pairs $\langle G, k \rangle$, where G is an instance of RTHG, k is an integer, and the MaxMin value is $\leq k$;* MaxMin Special RTHG *consists of those instances of Special RTHG for which the MaxMin value is m.*

Theorem 3. MaxMin RTHG *and* MaxSum RTHG *are both NP-hard.*

Proof. A Special RTHG partition π for G is perfect iff $\sum_{p \in P} u_p(\pi) = |P|$ iff $\text{MaxMin}(\pi) = m$ iff $\langle G, |P| \rangle \in$ MaxSum RTHG iff $\text{MaxSum}(\pi) = |P|$ iff $\langle G, m \rangle \in$ MaxMin RTHG. Therefore MaxMin RTHG and MaxSum 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 12. *[7] 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:

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

```
for $|C|$ compositions $c_0 \rightarrow c_{|C|-1}$ do
    for $m$ positions $r_0 \rightarrow r_{m-1} \in c_i$ do
        calculate the sum of agent votes on $\langle c_i, r_j \rangle$. $O(|P|)$
    end for
end for
for $|P|/m$ coalitions $t_0 \rightarrow t_{|P|/m-1}$ to assign to $\pi$ do
    find the set of compositions $C_{max}$ for which the sum of total votes is maximized.
    $O(|C| \cdot m)$
    select one composition $c_i$ uniformly at random from within the set.
    for $m$ positions $r_0 \rightarrow r_{m-1} \in c_i$ do
        find the set of agents $P_{max}(c_i, r_j)$ for whom the individual agent's vote for
        $\langle c_i, r_j \rangle$ is maximized. This takes time $O(|P|/m)$, given that the population
        shrinks by $m$ agents as each team is formed and removed.
        select one agent $p_j$ uniformly at random from within the set.
        add agent $p_j$ to the coalition $t_k$.
        for $|C|$ compositions $c_0 \rightarrow c_{|C|-1}$ do
            for $m$ positions $r_0 \rightarrow r_{m-1} \in c_i$ do
                remove agent $p_j$'s vote from the population, decrementing the sum total
                vote on $\langle c_i, r_j \rangle$.
            end for
        end for
    end for
    append team $t_k$ to the partition $\pi$.
end for
```

Observation 4 *The time complexity of GreedyRTHGPartiton is $O(|P|^2/m)$, or $O(|P| \cdot |C| \cdot m)$ if $|P| < |C| \cdot m^2$.*

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.

Two hundred and forty 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.

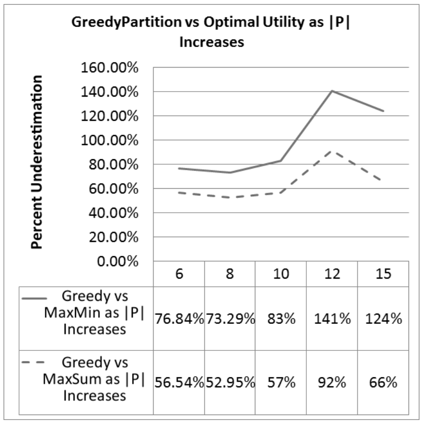
We began with $|P| = 6$, $|R| = 3$, and $m = 3$ in the minimal case. Ten random preference matrices were generated with these arguments. We then incremented $|R|$ by 1 and generated ten new random preference matrices, up to $|R| = 6$. This process was repeated for $\langle m, |P|\rangle = \langle 4, 8\rangle, \langle 5, 10\rangle, \langle 3, 12\rangle, \langle 4, 12\rangle, \langle 5, 15\rangle$. These upper bounds were chosen because larger inputs dramatically increased the time required for the brute force solvers to process the data.

Optimal results were calculated for each of these instances by MaxSum and MaxMin brute force implementations we developed. There are

$$O(|P|! \cdot (|C| + |P|/m)^{|P|/m})$$

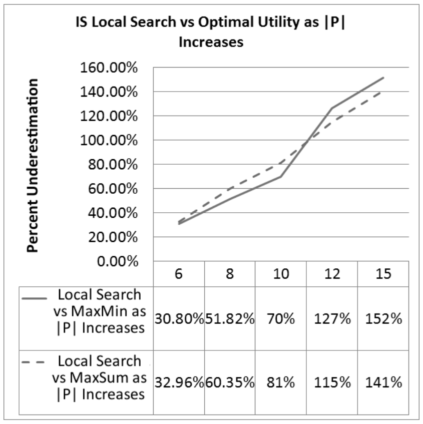
possible partitions in an instance of RTHG. We generate all of them and find the MaxSum and MaxMin values for each instance considered. Our implementation of *GreedyRTHGPartiton* ran each instance 500 times, in order to limit random error. For the same instances, IS solutions were constructed by our implementation of *ISLocalSearch*. Fifty initial partitions were selected uniformly at random for each instance as starting points for *ISLocalSearch*. We compared the mean utilities of partitions generated by *GreedyRTHGPartiton* and *ISLocalSearch* to the optimal results as $|P|$ increased.

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



	6	8	10	12	15
Greedy vs MaxMin as |P| Increases	76.84%	73.29%	83%	141%	124%
Greedy vs MaxSum as |P| Increases	56.54%	52.95%	57%	92%	66%

Fig. 1. Percent underestimate of optimal MaxSum and MaxMin using *GreedyRTHGPartiton* as $|P|$ increases



	6	8	10	12	15
Local Search vs MaxMin as |P| Increases	30.80%	51.82%	70%	127%	152%
Local Search vs MaxSum as |P| Increases	32.96%	60.35%	81%	115%	141%

Fig. 2. Percent underestimate of optimal MaxSum and MaxMin using *ISLocalSearch* as $|P|$ increases

Results are presented in Figures 1 and 2. We show the percentages by which *GreedyRTHGPartiton* and *ISLocalSearch* overestimate optimal MaxSum and MaxMin for each test case. The lower the percent overestimation the better. Each figure shows the mean overestimation as $|P|$ increases.

GreedyRTHGPartiton produces consistently better results for estimating MaxSum compared to MaxMin. The greedy heuristic may leave a very poor coalition at the end, lowering MaxMin performance. 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 $CDCE, CF$, 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. It may transpire that CD is the next team to be formed, *even if EF happens to be the worst coalition of all.*

Total utility is balanced out by strong selections made at the beginning, raising the performance against MaxSum. In our experiments, *GreedyRTHGPartiton* underestimates MaxSum utility by 68.38% and MaxMin by 105.76% on average.

ISLocalSearch performance against MaxMin and MaxSum optimal solutions is close. There is an approximately linear increase in overestimation as $|P|$ increases, because there are increasingly many local optima as the size of the input increases. In our experiments, *ISLocalSearch* underestimates the MaxSum optimal utility by 93.62% and the MaxMin optimal by 90.88% on average.

To test the stability of *GreedyRTHGPartition* solutions, we ran each of its outputs as input to the *ISLocalSearch* algorithm. We included an additional 80 inputs with $|R| = 5$ and $m = 5$, with $|P|$ increased for every 10 inputs. The results are shown in Figures 3 and 4. Defining Q as the number of iterations required for *ISLocalSearch* to form an IS partition from *GreedyRTHGPartition*, Q increases as $|P|$ increases. The speed with which Q grows relative to $|P|$ is the ratio $Q/|P|$. This is the number of local search iterations required per unit population. This ratio decreases at $|P|$ increases, suggesting that fewer local searches per unit population are required as $|P|$ grows.

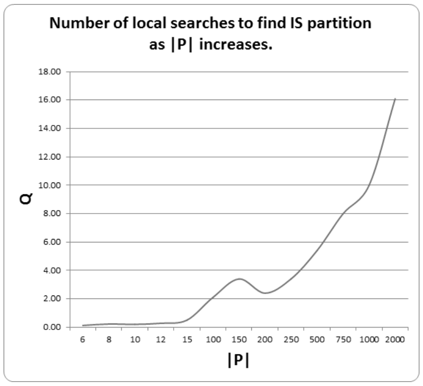


Fig. 3. Number of local searches to find IS partition as $|P|$ increases.

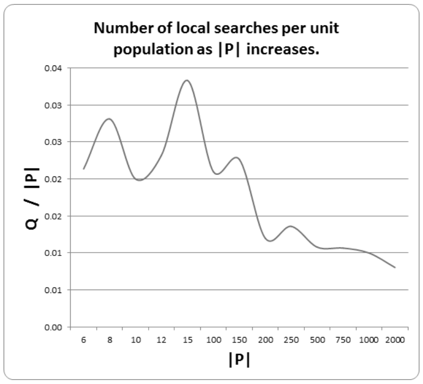


Fig. 4. Number of local searches *per unit population* as $|P|$ increases.

References

1. 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.
2. 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.
3. Haris Aziz, Felix Brandt, and Hans Georg Seedig. Computing desirable partitions in additively separable hedonic games. *Artificial Intelligence*, 2012.
4. S. Banerjee, H. Konishi, and T. Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18:135–153, 2001.
5. Suryapratim Banerjee, Hideo Konishi, and Tayfun Sönmez. Core in a simple coalition formation game. *Social Choice and Welfare*, 18(1):135–153, 2001.
6. A. Bogomolnaia and M.O. Jackson. The stability of hedonic coalition structures. *Games and Economic Behavior*, 38(2):201–230, 2002.
7. 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.
8. Vincent Conitzer and Tuomas Sandholm. Complexity of constructing solutions in the core based on synergies among coalitions. *Artificial Intelligence*, 170:607–619, 2006.
9. Andreas Darmann, Edith Elkind, Sascha Kurz, Jérôme Lang, Joachim Schauer, and Gerhard Woeginger. Group activity selection problem. In *Workshop Notes of COMSOC-2012*, 2012.
10. DFC. League of Legends Most Played PC Game. DFC Intelligence, 2012. `http://www.dfcint.com/wp/?p=343`.
11. J.H. Drèze and J. Greenberg. Hedonic coalitions: Optimality and stability. *Econometrica*, 48(4):987–?, 1980.
12. Oded Goldreich. *Computational Complexity, A Conceptual Perspective.* Cambridge University Press, 2008.
13. 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.
14. Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Stability in matching problems with weighted preferences. *Proc. ICAART11*, 2011.
15. 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`.
16. Riot. Let's talk about Champ Select - Page 5 - League of Legends. Riot Games Inc, 2013. `http://na.leagueoflegends.com/board/showthread.php?p=35559688#35559688`.
17. 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.
18. 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.

19. 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.