Roles and Teams Hedonic Game

Matthew Spradling, Judy Goldsmith, **Xudong Liu**, Chandrima Dadi and Zhiyu Li

Department of Computer Science
University of Kentucky
Lexington, KY, USA

1. Introduction
   - League of Legends®
   - Roles and Teams Hedonic Game (RTHG)
   - Coalition Formation Games (CFG)
2. RTHG Solutions
   - Optimization
   - Stability
3. Acknowledgments

League of Legends®

By hours played per month, the most popular online game in the world.

- Players are matched to small teams of 3 or 5.
- Players select their avatars, called **champions**.
- Each team plays against another team.

Figure: A few **champions** from **League of Legends**®.

Conflicts in Matchmaking

- Champions excel in different **roles** (Support, Attack, Defend...)
- Each player wants to play her preferred role.

Figure: Players may argue over their role selections.

Conflicts in Matchmaking

- A set of **roles** is a **team composition**.
- Each player wants her preferred team composition.

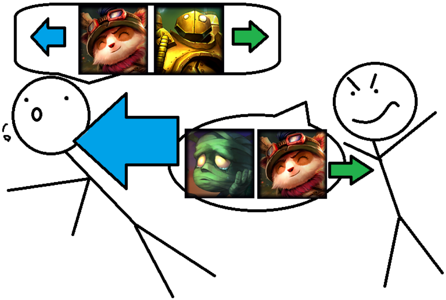

Figure: Players may argue over appropriate team compositions.

RTHG Model

- P: a population of agents;
- m: a team size
- R: a set of available roles
- C: a set of available team compositions. $t \in C$: a set of m not necessarily unique roles in R, where ordering doesn't matter.
- U: a utility function vector $\langle u_0, ..., u_{|P|-1} \rangle$ where for each agent $p \in P$, for each composition $t \in C$, and for each role $r \in t$, there is a utility function $u_p(r, t)$.

A solution is a partition of agents into teams of size m.

RTHG Instance

Table: 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
$\langle A, AA\rangle$	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
$\langle B, BB\rangle$	1	1	1	1

Coalition Formation Games (CFG)

- Players are to be divided into a partition of coalitions.
- Utility of a partition is determined by the agents in the coalitions.

Hedonic CFG

Utility of a partition depends only on each player's valuation of **her own coalition**, not other coalitions.

Other Hedonic CFG models:

- Group Activity Selection Problem
- Additively Separable Hedonic Games

Solution Concepts

Optimal solutions:

- Perfect partition
- Utilitarian partition (MaxSum)
- Egalitarian partition (MaxMin)

Stable solutions:

- Nash stable partition
- Individually stable partition

Perfect RTHG

Definition

A partition of agents to coalitions is **perfect** if each player p plays (r, t) and $u_p(r, t) = max\{u_p(r', t') : \ r' \in R \wedge t' \in C\}$. The language PERFECT RTHG consists of those instances of RTHG for which a perfect partition exists.

Theorem

PERFECT RTHG *is NP-hard.*

Proof Sketch.

We show that EXACT COVER $\leq_m^P$ a special case of PERFECT RTHG. $\square$

MaxSum and MaxMin RTHG

Definition

Given an instance I of RTHG, a **MaxSum partition** is one that achieves the maximum value of $\Sigma_{i<|P|} u_{p_i}$. A **MaxMin partition** is one that achieves the maximum value of $\min_{p \in P} u_p$.

Theorem

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

Proof Sketch.

We show that Special Perfect RTHG $\leq_m^P$ MaxSum RTHG.
We show that Special Perfect RTHG $\leq_m^P$ MaxMin RTHG. $\square$

Greedy Heuristic Partitioning

- Based on scoring voting.
- Agent's utility function vector is a ballot.

Table: Initialization Step: Determine votes upon each composition.

$\langle r, t \rangle$	$u_{p_0}(r, t)$	$u_{p_1}(r, t)$	$u_{p_2}(r, t)$	$u_{p_3}(r, t)$	**Total Vote**
$\langle A, AA \rangle$	3	3	0	0	**12 for AA**
$\langle A, AA \rangle$	3	3	0	0	
$\langle A, AB \rangle$	0	2	3	2	**14 for AB**
$\langle B, AB \rangle$	2	0	2	3	
$\langle B, BB \rangle$	1	1	1	1	**8 for BB**
$\langle B, BB \rangle$	1	1	1	1	

Greedy Heuristic Partitioning

Table: Select a composition with a maximum total vote.

$\langle r, t \rangle$	$u_{p_0}(r,t)$	$u_{p_1}(r,t)$	$u_{p_2}(r,t)$	$u_{p_3}(r,t)$	**Total Vote**
$\langle A, AA \rangle$	3	3	0	0	12 for AA
$\langle A, AA \rangle$	3	3	0	0	
$\langle A, AB \rangle$	0	2	3	2	**14 for AB**
$\langle B, AB \rangle$	2	0	2	3	
$\langle B, BB \rangle$	1	1	1	1	8 for BB
$\langle B, BB \rangle$	1	1	1	1	

Greedy Heuristic Partitioning

For each team being formed:

Table: Select m agents with the largest votes on the selected composition.

$\langle r, t\rangle$	$u_{p_0}(r,t)$	$u_{p_1}(r,t)$	$u_{p_2}(r,t)$	$u_{p_3}(r,t)$	**Total Vote**
$\langle A, AA\rangle$	3	3	0	0	12 for AA
$\langle A, AA\rangle$	3	3	0	0	
$\langle A, AB\rangle$	0	2	**3**	**2**	14 for AB
$\langle B, AB\rangle$	2	0	**2**	**3**	
$\langle B, BB\rangle$	1	1	1	1	8 for BB
$\langle B, BB\rangle$	1	1	1	1	

Greedy Heuristic Partitioning

For each team being formed:

Table: Match selected agents to preferred roles in this composition.

$\langle r, t\rangle$	$u_{p_0}(r,t)$	$u_{p_1}(r,t)$	$u_{p_2}(r,t)$	$u_{p_3}(r,t)$	**Total Vote**
$\langle A, AA\rangle$	3	3	0	0	12 for AA
$\langle A, AA\rangle$	3	3	0	0	
$\langle A, AB\rangle$	0	2	**3**	2	14 for AB
$\langle B, AB\rangle$	2	0	2	**3**	
$\langle B, BB\rangle$	1	1	1	1	8 for BB
$\langle B, BB\rangle$	1	1	1	1	

Greedy Heuristic Partitioning

For each team being formed:

Table: Remove selected agents from the population (team has been formed).

$\langle r, t \rangle$	$u_{p_0}(r,t)$	$u_{p_1}(r,t)$	**Total Vote**
$\langle A, AA \rangle$	3	3	**12 for AA**
$\langle A, AA \rangle$	3	3	
$\langle A, AB \rangle$	0	2	**4 for AB**
$\langle B, AB \rangle$	2	0	
$\langle B, BB \rangle$	1	1	**4 for BB**
$\langle B, BB \rangle$	1	1	

Continue until each player has been matched to a team.

Heuristic Testing

Observation

The time complexity of GreedyRTHGPartiton is $O(|P|^2/m)$*.*

- Experiment: Compared heuristic results to brute-force computed MaxSum and MaxMin on small populations up to $|P| = 15$.

Observation

The time complexity of brute force MaxSum or MaxMin calculation is $O(|P|! \cdot (|C| + |P|/m)^{|P|/m})$

Heuristic Results

- True MaxSum underestimated by 85.22% on average.
- True MaxMin underestimated by 119.79% on average.
- Consistently better at estimating MaxSum compared to MaxMin.

Role Change

Given a partition, a player may want to change to a prefered role.

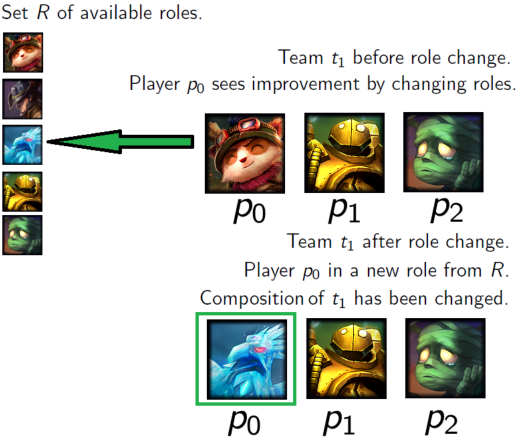

Figure: Player p_0 changes **roles**.

Position Swap

Given a partition, players may prefer to swap positions.

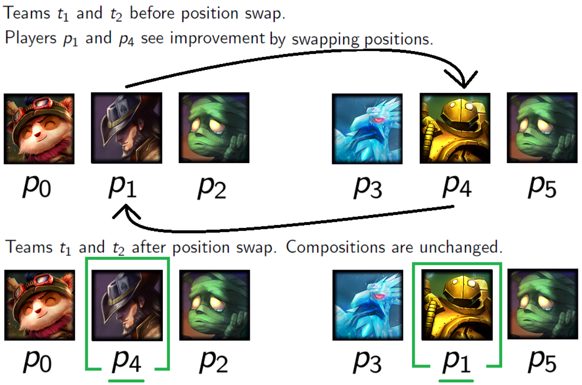

Figure: Players p_1 and p_4 swap **positions**.

Stable Partitions in RTHG

Definition

A partition is **Nash stable** iff no agent can improve her own utility by means of a role change or position swap.

Definition

A partition is **individually stable** iff no agent can improve her own utility by means of a role change or position swap without lowering the utility of any other agent.

Finding Stable Partitions

Theorem

Given an instance I of RTHG, an **individually stable** *solution can always be found by* **local search** *in time polynomial in $|I|$.*

Theorem

Given an instance I of RTHG, a **Nash stable** *solution may not always exist.*

For more details on these results, please see our paper in ADT'13!

Acknowledgments

- **League of Legends**®, **champions** and their images owned by **Riot Games**®, a subsidiary of **Tencent Holdings**®. Used under Fair Use for educational purposes.
- Xudong partially supported by NSF grant IIS-0913459.
- Thanks to Dr. Goldsmith's Comparative Decision Making course at the University of Kentucky, where this work first began!
- Slides prepared by Matthew Spradling.