Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)

# Team Formation with Heterogeneous Agents in Computer Games

**Robert G. Price, Scott D. Goodwin**

University of Windsor
401 Sunset Ave.
Windsor, Ontario, Canada
price9, sgoodwin@uwindsor.ca
http://cs.uwindsor.ca/~price9/teams/

**Abstract**

Forming teams using heterogeneous agents that perform well together to accomplish a task in a game can be a challenging problem. There can often be an enormous amount of combinations to look through, and having an agent that is really good at a particular task is no guarantee that agent will perform well on a team with members with different abilities. Picking a good team is important, as changing teams is often not allowed midway through a task.

## Using Roles to Create Teams

Roles can help simplify the search for the best team. As an example, a hockey coach might not put his best six players on the ice, especially if none of his top six players is a goalie. The concept of a goalie simplifies the search since it limits the criteria for the selection of that player to the abilities that are important for someone who is playing in that position.

The concept of roles is explained in (Lu 2005), where agents are classified by the capabilities they can offer. To help simplify a selection process, roles will be used to guide search; first the roles that appear in the best teams will be determined, and then best character types to fill those roles will be found. Some character types are more effective than others at certain roles, and some can play more roles than others. The ‘Defender’, ‘Striker’, ‘Leader’, and ‘Controller’ roles (D, S, L, and C) are described in (Schwalb 2009) and explain the roles different characters are best at in the Neverwinter Nights (NWN) game. NWN is a role-playing game, but has also been used as a teaching tool. NWN allows a player to choose from many options when making each character, allowing millions of combinations of unique teams, making it a good tool for this research. The characters have different abilities, and based on their strengths we can classify them in different roles. Characters in the ‘Defender’ role are tough and can take a lot of physical damage. ‘Strikers’ are good at dishing out damage, even from a distance. ‘Leaders’ can make their teammates more effective. ‘Controllers’ can damage many enemies simultaneously.

## Experimental Design

For these experiments, we attempted to come up with a procedure to find the most effective group based on roles, then determined which team-mates should fill the roles.

### 1) Get an initial indication of how the roles interact.

There were 4 members on each team, and each member was picked from one of the 4 roles (D, S, L, C) used to partition all of the character classes. There are 35 possible combinations of teams of 4 when picking from the 4 roles.

Originally we started with all 35 combinations of 4 roles and had them each fight the same series of benchmark teams several times, and recorded the winning percentage, the average fitness for each win, and the length of the fights. For the fitness, the percentage of remaining hit points was used– so a team that barely won a fight would have a positive score near 0, and a team that decisively beat its opponents without getting hurt had a score of 1.

The individuals that were used in the teams to represent each class were identical – they had the same equipment, statistics, etc. In other words the Fighter in one team was the same as the Fighters in other teams. There is a built-in “toughness rating” in the toolset that ships with NWN, so we could ensure that each of the agents under consideration was roughly balanced. If one individual was significantly better than the others, they might have ended up being included in all teams simply because of their ability, not because of how well they complemented their team’s abilities. Having a superior member on a team might be desired, but this would detract from the purpose of these experiments which was finding roles that complemented each other.

**2) Create a graphical representation of the team;** the weights were based on the team-mates success in step 1. The nodes represented the roles of the agents on the team; the edges represented the likelihood of the corresponding agents being on the team together.

Example: After the first trials, the team that had one Controller and three Leaders (C,L,L,L) performed the worst of all the teams, losing to 5 benchmark teams and beating none, so we decreased the weights between those pairs of roles ({C, L}, and {L, L})

**3) Evaluate the teams.**

We had a series of fights between teams created using the graph in 2, and updated the weights after each fight. We took the best teams from step 1 and used them as new benchmark teams, along with a few control teams (balanced, classic teams common in role-playing games). We created new teams using the weights in the graph, where character types with higher weights between them were included more often. We had these teams compete against the benchmark teams and adjusted the weights depending on how well the team did. This varied from the first part of the experiments where the roles changed: the teams that were created here had their roles specified, but the individual agents that could fulfill the roles varied at each trial, and were picked randomly between the available choices that represented that role.

**Non-combat skills:** When determining which teams performed well, the non-combat skills were ignored. If there was a need to include non-combat roles as well, the tests could be modified as follows. Include or replace some benchmark opponents with some tests that include the non-combat skills to test, and include a metric to evaluate effectiveness. The teams could then be compared using a fitness function similar to the other experiments.

For example, if the skills 'picking locks' or 'disarming traps' were considered important skills that we would want in a team, instead of some benchmark opponents, there could be a different benchmarks where there are scenarios where a team will have to pick a lock or disarm a trap. We would need some metric for comparing successes based on how important we view lock picking and disarming traps compared to combat – what do we value more: a team that can defeat an opponent but not pick a lock, or a team that can pick a lock but not defeat a certain opponent? Once the relative importance is determined, the rest of these experiments can be carried out as described above.

## Results

Throughout these experiments, some items become apparent: some teams perform better by having at least one agent that can fulfill a task, but experience diminishing returns having multiple agents fulfill it. For example, teams that had at least one individual that ran forward when the fight starts to engage the enemy end up holding up the opponent so the spell casters that remain in the back can get several spells off. It did not matter if that agent is particularly tough or not, the absence on someone to perform that task allows the enemy to directly approach and end the fight quickly.

Another observation was that the strategies picked didn't seem to take the opponents into account. For example, when facing four magical opponents, a wizard would cast some defensive spells that would protect him from magic – an effective move. The same wizard would also cast spells to protect against magic when not facing any opponents that could use offensive magic – a complete waste.

Table 1 shows partial results of the trials from step 1.

| | Benchmark Teams | | | | | |
|---|---|---|---|---|---|---|
| Team #, roles | 1 | 2 | 3 | 4 | 5 | 6 |
| … | | | | | | |
| 18, CLLS | 1 | 0 | 19 | 5 | 9 | 0 |
| 19, CLSS | 1 | 0 | 19 | 6 | 10 | 0 |
| 20, CSSS | 0 | 0 | 16 | 6 | 3 | 0 |
| 21, DDDD | 19 | 2 | 20 | 20 | 20 | 15 |
| … | | | | | | |

**Table 1: Wins of different teams**

The first column lists of the teams that were used in step 1, as well as the roles that are covered by the four team members. The columns represent the benchmark teams. The numbers in the cells represent how many times the team defeated the benchmark team out of twenty trials. (The team with four defenders defeated the second benchmark team twice and lost 18 times).

## Conclusions

This research provided a way to create teams of heterogeneous agents that performed tasks well. For example, from the complete set of experiments (not shown here) we were able to identify that the teams that had multiple 'Defenders' performed the best. Even though there can be an intractable number of combinations available when forming a team from different agents, by using roles to guide the search this process will provide a systematic way to form effective teams.

## Acknowledgements

These experiments used Pieter Spronck's Neverwinter Nights module, version 3 (Spronck 2010).

## References

Hongen, L. 2005. Team Formation in Agent-Based Computer Games, In ACM International Conference Proceeding Series 123: 121-124.

Schwalb, R. 2009. Party Building: From Basics to Themes, *Dragon* 373: 24-34.

Spronck, P. 2010 Pieter Spronck's Neverwinter Nights Modules. http://ticc.uvt.nl/~pspronck/nwn.html