In *Proceedings of the Plan, Activity, and Intent Recognition Workshop at the Twenty-Fifth Conference on Artificial Intelligence* (PAIR-11), San Francisco, CA, August 2011.

Role-Based Ad Hoc Teamwork

Katie Genter and **Noa Agmon** and **Peter Stone**
Department of Computer Science
The University of Texas at Austin
{katie, agmon,pstone}@cs.utexas.edu

Abstract

An ad hoc team setting is one in which teammates must work together to obtain a common goal, but without any prior agreement regarding how to work together. In this paper we present a *role-based approach* for ad hoc teamwork, in which each teammate is inferred to be following a specialized role that accomplishes a specific task or exhibits a particular behavior. In such cases, the role an ad hoc agent should select depends both on its own capabilities and on the roles currently selected by the other team members. We formally define methods for evaluating the influence of the ad hoc agent's role selection on the team's utility, leading to an efficient calculation of the role that yields maximal team utility. In simple teamwork settings, we demonstrate that the optimal role assignment can be easily determined. However, in complex environments, where it is not trivial to determine the optimal role assignment, we examine empirically the best suited method for role assignment. Finally, we show that the methods we describe have a predictive nature. As such, once an appropriate assignment method is determined for a domain, it can be used successfully in new tasks that the team has not encountered before and for which only limited prior experience is available.

1 Introduction

Ad hoc teamwork is a relatively new research area (Bowling and McCracken 2005; Jones et al. 2006)—and the subject of a AAAI challenge paper (Stone et al. 2010)—that examines how an agent ought to act when placed on a team with other agents such that there was no prior opportunity to coordinate behaviors. In some team domains, such as search and rescue missions and many team sports, the team behavior can be broken down into *roles*. In such domains, an ad hoc teamwork agent's main task is to decide which role to assume, such that the team's performance is maximized.

The decision of which role an ad hoc team member should assume is situation-specific: it depends on the task the team is to perform, on the environment in which it will operate, and on the capabilities of the team members. One trivial approach to the problem is for an ad hoc team member to assume the role at which it is most *individually* capable. However, the choice of optimal role—one that results in highest *team* utility—rarely depends only on the ad hoc team member, but also relies on the behavior of the other team members. We therefore examine the contribution of an ad hoc team member to the team by the measure of *marginal utility*, which is the increase (or decrease) in a team's utility when an ad hoc agent is added to the team and assumes a particular role. An *optimal mapping* of an ad hoc team member to a role is, therefore, one that maximizes the marginal utility, hence maximizing the contribution of the ad hoc agent to the team's utility. In this paper we describe several methods for modeling the marginal utility of a role selection as a function of the number of teammates performing the various roles.

As an example that we will return to throughout the paper, consider the well-studied problem of multi-robot foraging in which a team of robots is required to travel inside a given area, detecting targets and returning them to a predefined station (Mataric 1994). The goal of each robot is for the team to collect as many targets as possible as quickly as possible. Consider a special case of this problem, in which the targets are divided into two groups: red targets to the North and blue targets to the South, where the blue targets are worth twice as much to the team as the red targets. In this problem, as demonstrated in this paper, determining the optimal task the ad hoc agent should perform (corresponding to the type of target it should collect) is easily computable.

In more complex environments, deciding what functional form the marginal utility estimate should take becomes more challenging. For such cases, an empirical evaluation is required. We study a capture-the-flag style variant of Pacman (DeNero and Klein 2010) to investigate the implication of different possible role selections by the ad hoc agent on the marginal utility.

We show in three different tasks within this domain that it is possible to model the marginal utility of a role selection using a particular functional form. We then fit the parameters of this function with a limited amount of data from a new task, and use the new fitted function as a *predictive model* to determine how the ad hoc agent should behave in the new task in situations that it has not previously encountered.

The main contributions of this paper are i) a formalism of role-based ad hoc teamwork scenarios, ii) a classification of types of tasks according to the patterns they exhibit in terms of marginal utilities for role mappings, and iii) detailed experiments in a new role-based ad hoc teamwork domain.

2 Problem Definition

An ad hoc teamwork problem is one in which several agents find themselves in a situation where they all have perfectly aligned goals, yet they have had no previous opportunity to coordinate their teamwork (Stone et al. 2010). This problem arises quite often for humans, who tend to solve the problem quite naturally. However, autonomous agents — such as robots and software agents — do not currently handle this problem as gracefully.

Plenty of progress and success has been achieved using pre-coordinated teams of robots and agents. On these teams, which can be found in warehouses (Wurman, D'Andrea, and Mountz 2007), RoboCup soccer fields (Stone and Veloso 1999), and disaster recovery areas (Sugiyama, Tsujioka, and Murata 2008), the agents are often all programmed by the same group and all follow a pre-specified coordination protocol or rely on the ability to communicate. However, as agents become more prevalent in the world and are designed by many different companies and universities, there will become a need for these agents to adapt to one another and work together on the fly without pre-coordination and without relying on communication. Consider the case of a team of disaster recovery robots that are pre-coordinated to work together to rescue victims. If a robot becomes disabled and the team does not have an adequate replacement on site, the recovery process might significantly slow or halt until the robot could be repaired or a replacement could be shipped in. However, an ad hoc agent could step into the team and quickly adapt its behaviors to cooperate with the current teammates to allow the rescue process to continue.

In this work we study the *role-based* ad hoc teamwork problem, which is one that requires or benefits from dividing the task at hand into roles. Throughout this paper we will refer to the agents that make up the team as either *ad hoc agents* or *teammates*. The ad hoc agents are the agents that we designed and whose behavior we can control, while the teammates are the agents that were programmed by other groups or at different times such that future collaboration with our agents was unforeseeable. The teammates may not believe they are performing a role, but nevertheless it is useful for the ad hoc agent to classify their behaviors into roles.

In many ad hoc team settings the roles of an agent's teammates will be readily apparent. For example, in a pickup soccer game, each player's intended role is usually apparent from where the player is positioned at the beginning of the game (e.g. the goalie stands near the goal). Likewise, in a rescue scenario a teammate's role might be obvious based on the sensors and manipulators attached to the robot. However, in other cases it may take more extended observations to determine the intended roles of the teammates. Such a case might occur in the foraging domain mentioned above. In this domain, the role of each teammate can be determined with increasing certainty as observations are made about what color targets the agent collects and returns to the base.

We assume that different roles have different values to the team, and each agent has some ability to perform each role. As such, an ad hoc agent must take into account both the needs of the team and its own abilities when determining what role to adopt. A team receives a score when it performs a task, therefore the goal of the ad hoc agents is to choose roles that maximize the team score, and hence maximize the marginal utility of adding the ad hoc agents to the team.

Formally, let a task d be drawn from domain D, where task d has m roles $R(d) = \{r_0, ..., r_{m-1}\}$. Each role r_i has an associated relative importance value v_i, where r_x is more critical to team utility than r_y if $v_x > v_y$. Let $\mathbf{A} = \{a_0, ..., a_{n-1}\}$ be the set of ad hoc agents and $\mathbf{B} = \{b_0, ..., b_{k-1}\}$ be the set of teammates such that $T = A \cup B$ is the team that is to perform task d. Each agent $t_j \in T$ has a utility $u(t_j, r_i) \geq 0$ for performing each role $r_i \in R(d)$, which is a quantitative representation of the agent's ability to perform that role.

Let mapping $\mathbf{P} : B \rightarrow R(d)$ be the mapping of the teammates in B to roles $\{r_0, ..., r_{m-1}\}$ such that the teammates associated with role r_i are $B_i^P = \{b_{i_0}, ..., b_{i_{m_i^P - 1}}\}$, where $|B_i^P| = m_i^P$ and $B_0^P \oplus B_1^P \oplus ... \oplus B_{m-1}^P = B$. Remember that we can not command the teammates to perform particular roles. As such, mapping P may be given either fully or probabilistically, or it may need to be inferred via observation, in which case it is the ad hoc agents' assessment of which role is being performed by each teammate. For simplicity, we will write throughout this paper as if the teammates are performing the roles that the ad hoc agents believe they are performing. Now let mapping $\mathbf{S} : A \rightarrow R(d)$ be the mapping of the ad hoc agents in A to roles $\{r_0, ..., r_{m-1}\}$ such that the ad hoc agents performing role r_i are $A_i^S = \{a_{i_0}, ..., a_{i_{m_i^S - 1}}\}$, where $|A_i^S| = m_i^S$ and $A_0^S \oplus A_1^S \oplus ... \oplus A_{m-1}^S = A$. Finally, let mapping $SP : T \rightarrow R(d)$ be the combination of mappings S and P. As such, agents $T_i^{SP} = B_i^P \cup A_i^S$ are performing role r_i and $T_0^{SP} \oplus T_1^{SP} \oplus ... \oplus T_{m-1}^{SP} = T$. In other words, mapping SP is the association of *all* team members to the particular roles they are performing. Without loss of generality, the agents in each B_i^P are numbered such that $u(b_j, r_i) \geq u(b_{j+1}, r_i)$.

Notationally, let $T_i^W[k]$ denote the k agents that are performing role r_i according to mapping W with the k highest utilities for role r_i. Likewise, let $posB(a_j, r_i)$ denote the 0-indexed position in T_i^{SP} that the ad hoc agent a_j occupies. Finally, let $T_i^W(num)$ denote the agent that is performing role r_i under mapping W with the num highest utility on role r_i. For example, if agents A, B, C, and D are performing role R under mapping Y with the following utilities for role R: $A = 1$, $B = 2$, $C = 3$, and $D = 4$, then $T_R^T(0) = D$ and $T_R^T(1) = C$.

A team score $U(W, d, T)$ results when the set of agents T perform a task d, with each $t_j \in T$ fulfilling some role $r_i \in R(d)$ under mapping W. Team score U is a function of individual agent utilities, but its precise definition is tied to the particular domain D and specifically task $d \in D$. The marginal utility $MU(S, P)$ obtained by mapping S, assuming P is the mapping of the teammates in B to roles, is the score improvement obtained when each ad hoc agent $a_j \in A$ chooses role $r_S(a_j)$ under mapping S. Assuming that either teammates B can perform the task or that $U(P, d, B) = 0$ when B can not complete the task, marginal utility $MU(S, P) = U(SP, d, T) - U(P, d, B)$ [1].

Given that mapping P is fixed, the role-based ad hoc team problem is to find a mapping S that maximizes marginal utility. The problem definition and notation provided above are valid for any number of ad hoc team agents. Hence, although for the remainder of this paper we focus our attention on the case where there is only one ad hoc agent such that $A = \{a_0\}$, our general theoretical contributions can still be applied in teams to which multiple ad hoc agents are added. For example, multiple ad hoc agents could coordinate and

[1] MU is a function of d, B, T, P and S, however throughout the paper we use the implicit notation

work together as a single 'agent' under the theoretical contributions presented below.

3 Choosing a Role—Proposed Models

The ground truth way for an ad hoc agent to determine the marginal utility from selecting a particular role, and hence determine its optimal role, is to determine $U(SP, d, T)$ for each possible role it could adopt. However, in practice, the ad hoc agent must *predict* its marginal utility for all possible roles and then select just *one* role to adopt. Here we lay out five possible models with which the ad hoc agent could do this prediction based on the roles its teammates are currently filling, where each model is appropriate for a different class of role-based tasks.

For all of the models except the Unlimited Role Mapping model we assume that the ad hoc agent a_0 knows the utilities $u(b_j, r_i), \forall b_j \in B, r_i \in R(d)$ and the mapping $P : B \rightarrow R(d)$. Additionally, when considering the following five models, note that the marginal utility of agent a_j choosing to fulfill role r_i under mapping S is often given by an algorithm MU-X. In these cases, $\text{MU-X}(a_j, r_i, P) = U(SP, d, T_i^{SP}) - U(P, d, B_i^P)$.

Unlimited Role Mapping Model:

Recall the multi-robot foraging example from Section 1. If the number of targets is unlimited, the size of the area is unbounded, and each target can be acquired by a single robot, then the optimal role for the ad hoc agent would be one that acquires the highest value target that it is capable of collecting. In tasks such as this one, the benefit the team receives for an agent performing a role does not depend on the roles fulfilled by its teammates.

In such cases, the contribution to the team of an agent t_j performing role r_i is simply the agent's utility $u(t_j, r_i)$ at role r_i multiplied by the value of the role v_i. As such, the team utility can be modeled as $U(SP, d, T) = \sum_{i=0}^{m-1} rs_i * v_i$, where $rs_i = \sum_{t_j \in T_i^{SP}} u(t_j, r_i)$. Note that in this model, agent utility $u(t_j, r_i)$ for performing each role r_i and the importance v_i of each role r_i are parameters that can be tuned to match the characteristics of a particular task. Theorem 1 describes the optimal role mapping under this model.

Theorem 1. *In Unlimited Role Mapping tasks, mapping S, under which a_0 chooses the role r_i that obtains $\operatorname{argmax}_{0 \le i \le m-1} u(a_0, r_i) * v_i$, maximizes marginal utility such that $\forall S' \neq S \; MU(S', P) \leq MU(S, P)$.*

Limited Role Mapping Model:

Returning to the multi-robot foraging example, assume as before that the number of targets is unlimited and the size of the area is unbounded—but now also assume that each target needs at least three, but no more than six, agents in order to be successfully acquired. In tasks such as this, each role r_i has an associated r_i^{min} value and r_i^{max} value that represent the minimum and maximum number of agents that should perform role r_i. For all i, let $0 \leq r_i^{min} \leq r_i^{max} \leq n$.

If the number of agents performing role r_i is less than r_i^{min}, then the team gains no score from their actions. On the other hand, if the number of agents performing role r_i is greater than r_i^{max}, then only the r_i^{max} agents with highest utility, $T_i^{SP}[r_i^{max}]$, will be taken into account when calculating the team score. As such, the team utility for Limited Role Mapping tasks can be modeled as $U(SP, d, T) = \sum_{i=0}^{m-1} rs_i * v_i$, where

$$rs_i = \begin{cases} \sum_{t_j \in T_i^{SP}} u(t_j, r_i) & \text{if } r_i^{min} \leq m_i^{SP} \leq r_i^{max} \\ \sum_{t_k \in T_i^{SP}[r_i^{max}]} u(t_k, r_i) & \text{if } m_i^{SP} > r_i^{max} \\ 0 & \text{if } m_i^{SP} < r_i^{min} \end{cases}$$

The function $\text{MU-1}(a_j, r_i, P)$ displayed in Algorithm 1 gives the marginal utility obtained from the ad hoc agent a_j choosing to perform role r_i, where the current mapping of teammates to roles is described by P. In this model, agent utility $u(t_j, r_i)$ for performing each role r_i, the importance v_i of each role r_i, and the minimum and maximum number of agents that should perform each role r_i are all tunable model parameters. Theorem 2 describes the optimal role mapping for the ad hoc agent under this model.

Algorithm 1 $\text{MU-1}(a_j, r_i, P)$

if $m_i^P + 1 < r_i^{min}$ **then**
  return 0
else
  if $m_i^P + 1 = r_i^{min}$ **then**
    return $\sum_{t_j \in T_i^{SP}} u(t_j, r_i) * v_i$
  else
    if $r_i^{max} < m_i^P + 1$ **then**
      if $posB(a_j, r_i) \leq r_i^{max}$ **then**
        return $u(a_j, r_i) * v_i - u(T_i^P(r_i^{max}), r_i) * v_i$
      else
        return 0
    else
      return $u(a_j, r_i) * v_i$

Theorem 2. *In Limited Role Mapping tasks, mapping S, under which a_0 chooses the role r_i that obtains $\operatorname{argmax}_{0 \le i \le m-1} \text{MU-1}(a_0, r_i, P)$, maximizes marginal utility such that $\forall S' \neq S \; MU(S', P) \leq MU(S, P)$.*

Incremental Value Models:

Continuing the multi-robot foraging example, consider the case where the size of the South area containing the blue targets is bounded. In such a bounded area, adding robots hinders the speed with which all of the robots in the area can acquire targets and return them to a station. Therefore the optimal role might be one that collects a less valuable target in the less congested North area or it might be one that collects a valuable target in the congested South area if the value of the target offsets the penalty felt by all agents in the area due to increased congestion.

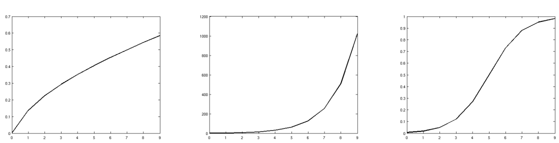
In tasks such as this one, the value added by agents performing a role may not be linearly correlated with the number of agents performing that role. As such, the team utility in incremental value tasks can be modeled as $U(SP, d, T) = \sum_{i=0}^{m-1} rs_i * v_i$, where $rs_i = \sum_{t_j \in T_i^{SP}} u(t_j, r_i) * F(i, j)$.

In particular, we consider the following three functions F — each with two parameters that can be tuned to match the characteristics of a particular task — that describe how the value added to the team by each subsequent agent performing a role incrementally increases or decreases as more agents perform that role. Example curves obtained using an example set of parameters can be seen in Figure 1. As has been convention so far in this paper, subscript i is used to number roles and subscript j is used to number agents, where agents with higher utilities for performing a role are represented by lower js.

Logarithmic Function $F(i, j) = \log_{j+1}(x_i) + k_i$, where k_i represents the amount added to the role score rs_i for each agent performing role r_i and x_i sets the pace at which the function decays for agents performing role r_i.

Exponential Function $F(i, j) = b_i^{(j/n_i)}$, where b_i is the growth factor and n_i is the time required for the value to decrease by a factor of b_i — both for each agent performing role r_i.

Sigmoidal Function $F(i, j) = \frac{1}{1+e^{n_i*(m_i^{SP}+b_i)}}$ where n_i determines the sharpness of the curve and b_i dictates the x-offset of the sigmoid from the origin for each agent performing role r_i.



(a) Logarithmic (b) Exponential (c) Sigmoidal

Figure 1: Example curves for each of the three incremental value functions. The x-axis is the number of agents performing a particular role and the y-axis is the team utility. All curves use $u(t_j, r_i) = 1$ and $v_i = 1$. The logarithmic curve uses $x_i = 1.1$ and $k_i = 0$, the exponential curve uses $b_i = 2$ and $n_i = 0.9999$, and the sigmoidal curve uses $n_i = -0.99$ and $b_i = -5$.

As may be noted in Figure 1, the sigmoidal function can perform closely to the exponential function on one side of its curve and closely to the logarithmic function on the other side. Parameter b_i can be tuned to effectively turn the sigmoidal curve into an exponential curve or a logarithmic curve. However, we still consider the logarithmic and exponential functions because they might be better models than the entire sigmoid function in some cases.

The function MU-2(a_j, r_i, P) displayed in Algorithm 2 gives the marginal utility obtained from the ad hoc agent a_j choosing to perform role r_i, where the current mapping of teammates to roles is described by P. In this model, agent utility $u(t_j, r_i)$ for performing each role r_i, the importance v_i of each role r_i, and the parameters used in function F are all tunable parameters. Line 7 in function MU-2(a_j, r_i, P) handles the case where the ad hoc agent a_j is not the only agent on the team performing role r_i (handled by line 2) and not the agent with the lowest utility for performing role r_i that is performing r_i (handled by line 5). As such, line 7 accounts for the benefit obtained by the team from agent a_j performing role r_j and then subtracts small incremental amounts for each agent performing role r_i with lower utility than a_j. We do this because under the incremental models, agents performing some role r_i provide benefit to the team at different levels based on their utility for performing r_i. Hence, when an agent with a higher utility for performing r_i joins the team and performs r_i, the benefit obtained by each of the agents performing r_i with lower utility decreases slightly. Note that although in this paragraph we assumed benefit was obtained as additional agents joined the team, our models can also handle the case where additional agents add penalty as they join the team.

Algorithm 2 MU-2(a_j, r_i, P)

1: **if** $m_i^P = 0$ **then**
2: **return** $v_i * (u(a_j, r_i) * F(j, 1))$
3: **else**
4: **if** $posB(a_j, r_i) = m_i^P$ **then**
5: **return** $v_i * u(a_j, r_i) * F(j, m_i^P + 1)$
6: **else**
7: **return** $v_i * u(a_j, r_i) * F(j, posB(a_j, r_i) + 1) - \sum_{y=b_{posB(a_j,r_i)}}^{m_i^P - 1} (u(b_y, r_i) * v_i * F(j, posB(b_y, r_i) + 1) - u(b_y, r_i) * v_i * F(j, posB(b_y, r_i) + 2))$

Theorem 3 describes the optimal role mapping for the ad hoc agent under this model.

Theorem 3. *In Incremental Value tasks, mapping S, under which a_0 chooses the role r_i that obtains* $\operatorname{argmax}_{0 \le i \le m-1}$ MU-2(a_0, r_i, P)*, maximizes marginal utility such that $\forall S' \neq S\ MU(S', P) \leq MU(S, P)$.*

4 Model Evaluation

In Section 3, we examined the contribution of an ad hoc agent to its team by providing several methods for modeling the marginal utility of a role selection as a function of the number of teammates currently performing each role. Each model is appropriate for some tasks. But given a task in a particular environment, how should the correct model be selected? Additionally, once a model is selected, how should we determine reasonable parameters for the model given limited ground truth data? In the second half of this paper we examine both of these questions in the Pacman Capture-the-Flag environment.

The Pacman Capture-the-Flag Environment

We empirically examine each of the five models described above in a capture-the-flag style variant of Pacman designed by John DeNero and Dan Klein (DeNero and Klein 2010). The Pacman map is divided into two halves and two teams compete by attempting to eat the food on the opponent's side of the map while defending the food on their home side. A team wins by eating all but two of the food pellets on the opponent's side or by eating more pellets than the opponent before three thousand moves have been made. When a player is captured in opponent territory, it restarts at the team's starting point.

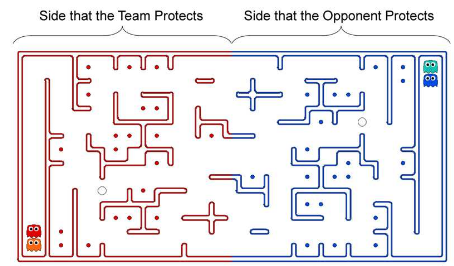
The result of each game is the difference between the number of pellets protected successfully by the team and the number of pellets successfully protected by the opponent—we refer to this result as the *score differential*. Wins, losses and ties result in positive, negative and zero score differentials (respectively). More importantly, high positive score differentials indicate that the team dominated the opponent, while score differentials closer to zero indicate that the two teams were well matched. We therefore input the score differential from each game into the following sigmoid function $1/1 + e^{-0.13 * scoreDifferential}$ to obtain *ground truth data* for the team's utility when different numbers of agents fill each role. We examined different values for the multiplicand and found that 0.13 yielded the most representative score differential spreads in the three tasks that we used to select the most appropriate model. Note that we transform the score differentials using a sigmoid function because this emphasizes differences in score differentials close to zero. For example, score differentials of -15, -10, -5, and 0 become 0.125, 0.214, 0.343, and 0.5 after being transformed by the sigmoid function. This is desirable because we mainly care whether we win or lose — and hence we want the difference between score differentials -15 and -10 after being transformed by the sigmoid function to be less than the difference between score differentials -5 and 0 after being transformed by the sigmoid function.

In each experiment we consider two roles that could be performed: $R = \{\text{offense, defense}\}$. Offensive players always move toward the closest food on the opponent's side, making no effort to avoid being captured by defenders or to capture opponents while in their own territory. On the other hand, defensive players wander on their own side and chase down any invaders they see. These offensive and defensive behaviors are deliberately suboptimal, as we focus solely on role decisions given whatever behaviors the agents execute when performing their roles.

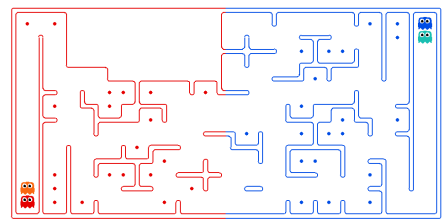
We consider the opponents and map to be fixed and part of the environment for each experiment. All of the agents — opponents, teammates, and the ad hoc agent—run either the offensive or defensive behavior just described. Additionally, half of the opponents perform defensive behaviors and half perform offensive behaviors.

Choosing a Model

We use three tasks to determine which of the models best represents the marginal utility of a role selection for the Pacman Capture-the-Flag environment. In particular, a task is defined by the number of opponents and the map. The first task "vs-2" is against two opponents on the "Basic" map shown in Figure 2(a), the second task "vs-6" is against six opponents on the "Basic" map, and the third task "vs-2-SmallDefense" is against two opponents on the "SmallDefense" map shown in Figure 2(b). The "Basic" map is substantially different from the "SmallDefense" map in that the defensive agents for the team on the left side of the map have about 33% less territory to defend, and hence intuitively should derive less benefit from adding additional defensive agents after some point.



(a) "Basic" Map



(b) "SmallDefense" Map

Figure 2: Maps used to determine the most representative model.

In order to decide which of the models is most representative of the marginal utility of a role selection in the Pacman Capture-the-Flag environment, we first gather full sets of *ground truth data* for the three tasks presented above. In particular, in each task we gather scores over one thousand games for teams of zero to six offensive agents and zero to six defensive agents (i.e., for forty-nine teams) and then calculate the ground truth performance over the one thousand runs for the forty-nine teams. As mentioned above, we calculate the ground truth data for each team by putting the score differential from each of the one thousand games though the sigmoidal function given above and then averaging the results. The ground truth data from the "vs-2" environment is shown in Table 1. Note that 0.09 is the worst possible ground truth performance, and corresponds to obtaining 0 pellets and losing all 18 pellets to the opponent.

We then use the ground truth data to determine the *ground truth decision* of whether an ad hoc agent should perform an offensive role or a defensive role on any team composed of zero to five offensive agents and zero to five defensive agents in each of the three tasks. For example, to determine the ground truth decision of whether it is better for the ad hoc agent to perform an offensive or defensive role when added to a team with three offensive agents and two defensive agents, we look at whether the ground truth data is higher for a team with three offensive agents and three defensive agents or for a team with four offensive agents and two defensive agents. If the former (latter) is true, then the ground truth decision is that it is best for the ad hoc agent to perform a defensive (offensive) behavior if added to this team. We determine whether a ground truth decision is statistically significant by running a two-tailed Student's t-Test assuming two-sample unequal variance on the score differentials obtained if an offensive agent is added to the current team and the score differentials obtained if a defensive agent is added to the current team.

	0d	**1d**	**2d**	**3d**	**4d**	**5d**	**6d**
0o	0.09 (+o)	0.09 (+o)	0.09 (+o)	0.13 (+o)	0.23 (+o)	0.31 (+o)	0.36
1o	0.29 (+d)	0.49 (X)	0.64 (+o)	0.74 (+o)	0.79 (+o)	0.81 (+o)	0.82
2o	0.42 (+d)	0.63 (+d)	0.75 (+d)	0.81 (+d)	0.83 (X)	0.85 (X)	0.86
3o	0.54 (+d)	0.71 (+d)	0.80 (+d)	0.83 (+d)	0.85 (X)	0.85 (X)	0.86
4o	0.56 (+d)	0.74 (+d)	0.81 (+d)	0.84 (+d)	0.85 (+d)	0.87 (X)	0.87
5o	0.61 (+d)	0.75 (+d)	0.83 (+d)	0.84 (+d)	0.86 (X)	0.87 (+d)	0.88
6o	0.64	0.79	0.83	0.86	0.87	0.88	0.88

Table 1: Ground truth data and decisions from the "vs-2" environment, rounded to two decimal points. The rows represent the 0...6 agents performing an offensive role, while the '0d...6d' columns represent the 0...6 agents performing a defensive role. A '+o' ('+d') decision means that the ad hoc agent should adopt an offensive (defensive) role if added to a team with teammates performing the roles indicated by the row and column. An 'X' decision means that the decision of which role to perform was not statistically significant at $p = 0.05$.

With the ground truth decisions for the ad hoc agent in each of the three tasks, we can determine which of the five models best captures the actual marginal utility of role selection in each of the three tasks. First, we input the ground truth data and the model function into Matlab's lsqcurvefit algorithm (which uses the trust region reflexive least squares curve fitting algorithm) and obtain *fitted parameters* for the model function. The fitted parameters vary in type and number for each of the five models, but often include the role importance value v_i, the agent's utility $u(a_j, r_i)$ at performing role v_i, and parameters of the model function — both for each role $r_i \in R(d)$. We use the fitted parameters to calculate *fitted results* for teams of zero to six offensive agents and zero to six defensive agents (i.e., forty-nine teams). Last, we translate these fitted results into *fitted decisions* using the same methodology used to translate the ground truth score differentials into ground truth decisions.

Now that we have ground truth decisions for each of the three tasks and fitted decisions for all five models in the three tasks, we compare the number of times the ground truth decision (for example, '+o') is statistically significant but does not match the fitted decision for a particular team arrangement (for example, '+d')—in other words, the number of times the model made an *incorrect decision*.

Model	**vs-2**	**vs-6**	**vs-2-Small Defense**
Unlimited Role Mapping	19	8	14
Limited Role Mapping	13	11	11
Logarithmic Incremental Value	3	2	1
Exponential Incremental Value	1	1	1
Sigmoidal Incremental Value	**1**	**0**	**0**

Table 2: The number of statistically significant incorrect decisions made by each model in the three tasks.

As is apparent from Table 2 the sigmoidal model makes the fewest incorrect decisions in all three tasks. Additionally, as can be seen in Table 3, the sigmoidal model also obtains the least model error. Although model error is not as strong of an indicator as the number of incorrect decisions — since the actual decision is what matters, not the amount by which the ground truth data differs from the fitted results — it is still encouraging to see that the sigmoidal model has the least error.

Model	**vs-2**	**vs-6**	**vs-2-Small Defense**
Unlimited Role Mapping	1.7650	0.2371	1.6692
Limited Role Mapping	1.5695	0.2645	1.4422
Logarithmic Incremental Value	0.1602	0.1335	0.1061
Exponential Incremental Value	0.1239	0.1284	0.0833
Sigmoidal Incremental Value	**0.1161**	**0.1051**	**0.0768**

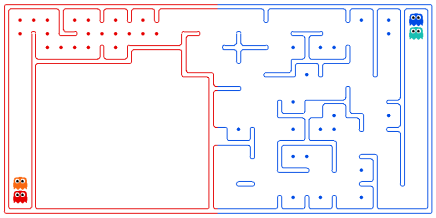
Table 3: The error of each model in the three tasks. The error is calculated to be the value of the squared 2-norm of the difference between the fitted results and the ground truth data.

From the results presented in this section, we conclude that in the Pacman Capture-the-Flag domain, at least on the maps and opponents we studied, the sigmoidal incremental model most accurately models team utility. However, to conclude this we generated a full set of ground truth data for each of the three tasks, amounting to 49,000 games per task and used this data to fit the parameters of the model. Next we consider how to use the sigmoidal model for predictive modeling when substantially less ground truth data is available.

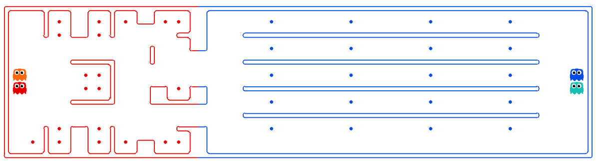
Predictive Modeling

Once a model type has been selected for a domain, the ad hoc agent can use this model to predict the marginal utility of role selection on new tasks in this domain for which we have limited ground truth data. Essentially we want to be able to determine how the ad hoc agent should behave in a new task—including never seen before situations—without the expense of gathering substantial amounts of ground truth data for every scenario. We do this by choosing fitted parameters — also referred to as 'fitting the model' — for the new task based on the data that is available. Below we evaluate the accuracy of the chosen model in our Pacman domain on multiple new tasks in which only a limited amount of randomly selected data is available for fitting the models. We also consider how much ground truth data is enough for the ad hoc agent to make acceptable role selection decisions on a new task. Note that we use the sigmoidal model in this section because the experiments in the previous section indicate that it is the most appropriate model for this domain. However, as we will discuss later in this paper, we find that the model parameters need to be tuned for each new task encountered.

We use two new tasks in this section. The first task "vs-2-alley" is against two opponents on the "AlleyDefense" map shown in Figure 3(a) and the second task "33%Defense" is against two opponents on the "33%Defense" map shown in 3(b). Both the "AlleyDefense" and "33%Defense" maps include a smaller defensive area than offensive area for the team that the ad hoc agent is added to, but the alley in "AlleyDefense" calls for the ad hoc agent to behave very differently than in the "33%Defense" map where the opponent's food pellets are relatively easy for the ad hoc agent's team to capture. Specifically, in the "33%Defense" map it is desirable—up to a certain threshold—to add an offensive agent as long as there is at least one defensive agent, whereas in the "AlleyDefense" map it is desirable to have substantially more defensive agents than offensive agents as long as there is at least one offensive agent.



(a) "AlleyDefense" Map



(b) "33%Defense" Map

Figure 3: The maps used for the tasks in this section.

Consider the case in which the ad hoc team agent is given, either through experience or observation, randomly selected data points that represent some sparse experience in the domain, where a data point consists of the number of agents fulfilling each role and the average ground truth data calculated over just twenty-five games. In this experiment, we evaluate the predictive model's fit from one to forty-nine randomly selected data points, measuring accuracy by the number of incorrect decisions made by each model. Note that if only one data point is used to fit the model, then score differentials from only 25 games are required. Likewise, if ten data points are used, 250 (10*25) games are required. Even if all forty-nine data points are used, only 1,225 (49*25) games are required. To put these game numbers in perspective, we can easily run 250 games on our high-throughput computing cluster in under 5 minutes when the cluster is not saturated.

Figure 4 shows the accuracy of the sigmoidal model on the "vs-2-alley" and "33%Defense" tasks when given varying amounts of randomly selected data points calculated from twenty-five games. As the figure shows, the accuracy of the model improves steadily in both tasks as additional data points are used to fit the model.

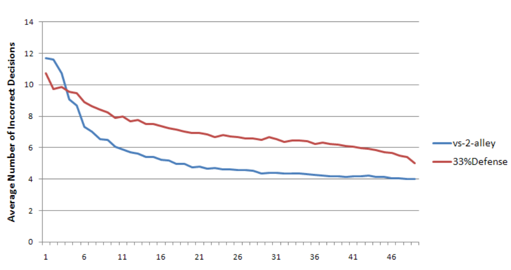


Figure 4: Accuracy of the sigmoidal predictive model (averaged over 1000 trials) using various amounts of randomly selected data points from twenty-five games in two different tasks.

Finally, we consider how much ground truth data is actually necessary for the ad hoc agent to make acceptable role selection decisions. The results given so far in this section were obtained from predictive models using parameters that were fit using average ground truth data gathered over twenty-five games. We determined that twenty-five games were enough to make adequate role selection decisions by considering the accuracy of the "vs-2-alley" predictive model when using parameters that were fit on ground truth data for 25 and 49 random data points gathered over one to one hundred games. As can be seen in Table 4, twenty-five games does indeed seem to be an appropriate trade-off between the time required to collect ground truth data and the value of minimizing incorrect decisions by the model.

Number of Games	**Incorrect Predictions when using 25 Points**	**Incorrect Predictions when using 49 Points**
1	12.8	11
5	9.6	9
10	7.1	7
15	5.1	6
20	7	7
25	**4.6**	**4**
30	2.5	2
35	1.3	1
40	3.2	2
45	3.2	3
50	3.3	3
100	2.4	2

Table 4: The average number of incorrect predictions obtained on the "vs-2-alley" task using the sigmoidal predictive model.

Determining the Fitted Parameters

In the previous section we presented the idea that once an ad hoc agent has chosen a model for a particular domain, it can use this chosen model to predict the marginal utility of role selection on new tasks in the same domain by using limited ground truth data to determine new fitted parameters for the model — or in other words, fit the model. Remember that fitted parameters can be obtained by inputting the ground truth data and the sigmoidal model function into Matlab's lsqcurvefit algorithm, as this will fit the sigmoidal model to the limited ground truth data using a least squares curve fitting algorithm. Then these fitted parameters can be used to calculate fitted results and fitted decisions (as described in the "Choosing a Model" section), which represent the decisions chosen by the model given each possible set of teammates.

However, how important is it to use (limited) ground truth data to determine appropriate fitted parameters for the chosen model function in a new task? We found experimentally that if parameters fit on one task are used on another task, the results can be quite poor. For example, although fitted parameters for the "vs-6" task perform well on that task (the sigmoidal models yield 0 incorrect decisions), attempting to use these same fitted parameters on the "33%Defense" task yields an abysmal 15 incorrect decisions and only 16 correct decisions. Not surprisingly, we found that the similarity of the tasks and the similarity of their ground truth fitted decisions played a sizable role in how well the model did without finding new fitted parameters. Therefore, if we can determine close similarity of a task to one for which we already have a well-fitted model, then it may be preferable to use the fitted model as is — especially if we only have extremely limited ground truth data in the new task. However, in most cases we found that it is important and worthwhile to find new fitted parameters when a new task is encountered and there is opportunity to obtain *any* ground truth data.

5 Related Work

This paper contributes towards answering the ad hoc teamwork challenge, which calls for teammates to work together without any prior coordination (Stone et al. 2010). Most prior research on multi-agent teamwork requires explicit coordination protocols, languages, and/or shared assumptions (e.g. (Grosz and Kraus 1996; Tambe 1997)). Some multi-robot teams are even designed to work specifically with their teammates in pre-defined ways, such as via "locker-room agreements" (Stone and Veloso 1999). Other multi-robot teams, although non-communicating, are designed assuming that all team members are reactive and homogenous (Lerman et al. 2001).

Bowling and McCracken (2005) examined the concept of "pick-up" teams in simulated robot soccer. Similarly to us, they propose coordination techniques for a single agent that wants to join a previously unknown team of existing agents. However, they take a different approach to the problem in that they provide the single agent with a play book from which it selects the play most similar to the current behaviors of its teammates. The agent then selects a role to perform in the presumed current play.

Jones *et al.* (2006) perform an empirical study of dynamically formed teams of heterogeneous robots in a multi-robot treasure hunt domain. They assume that all of the robots know they are working as a team and that all of the robots can communicate with one another, whereas in our work we do not assume that the teammates realize they are working on a team with the ad hoc agent.

There has been previous research on multi-agent teamwork in both the Capture-the-Flag domain and the foraging domain (e.g. Sadilek and Kautz (2010), Lerman *et al.* (2006)). However, most of this work focuses on coordination between all teammates instead of coordination of one or more ad hoc agents with existing teammates, and hence does not truly address the ad hoc teamwork problem.

6 Conclusions and Future Work

This paper presented a formalization of role-based ad hoc teamwork settings and introduced several methods for modeling the marginal utility of an ad hoc agent's role selection as a function of the number of teammates currently performing each role. We assume in this work that the roles of the teammates are known and that we know how well some team configurations (i.e., the number of teammates fulfilling each role) do in a particular task. However, we do not know how much an agent could help the team if added to each role. As such, we showed that it is possible to use a particular functional form to model the marginal utility of a role selection in a variety of tasks. Additionally, only a limited amount of data is needed on a new task in order to be able to fit the function such that it can be used as a predictive model to determine how an ad hoc agent should behave in situations of a new task that it has not previously encountered.

This research is among the first to study role-based ad hoc teams. As such, there are many potential directions for future work. We provide both theoretical and empirical contributions in this paper, and although our empirical results consider the minimal team size in which our contributions apply, the theoretical formulation is fully general to larger teams and applicable in domains with additional roles. We do plan on expanding our work into more interesting and complicated environments with more than two potential roles to fulfill and more than one ad hoc agent. Additionally, we wish to consider the case in which the ad hoc agents encounter teammates that are running unfamiliar behaviors, forcing the ad hoc agents to model their teammates in order to classify their behaviors into known roles and successfully collaborate. Finally, another possible direction is to assume that teammates may be modifying their behavior in response to the actions of the ad hoc agents, making teamwork more difficult.

7 Acknowledgements

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-0917122), ONR (N00014-09-1-0658), and the FHWA (DTFH61-07-H-00030).

References

Bowling, M., and McCracken, P. 2005. Coordination and adaptation in impromptu teams. In *AAAI'05*, 53–58.

DeNero, J., and Klein, D. 2010. Teaching introductory artificial intelligence with pac-man. In *Proceedings of the Symposium on Educational Advances in Artificial Intelligence*.

Grosz, B. J., and Kraus, S. 1996. Collaborative plans for complex group action. *AIJ* 86(2):269 – 357.

Jones, E.; Browning, B.; Dias, M. B.; Argall, B.; Veloso, M. M.; and Stentz, A. T. 2006. Dynamically formed heterogeneous robot teams performing tightly-coordinated tasks. In *ICRA'06*, 570 – 575.

Lerman, K.; Galstyan, A.; Martinoli, A.; and Ijspeert, A. J. 2001. A macroscopic analytical model of collaboration in distributed robotic systems. *Artificial Life* 7:375–393.

Lerman, K.; Jones, C.; Galstyan, A.; and Mataric, M. 2006. Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research* 25:225–242.

Mataric, M. 1994. Learning to behave socially. In *From Animals to Animats: International Conference on Simulation of Adaptive Behavior*, 453–462. MIT Press.

Sadilek, A., and Kautz, H. 2010. Modeling and reasoning about success, failure, and intent of multi-agent activities. In *Ubiquitous Computing: Mobile Context-Awareness*.

Stone, P., and Veloso, M. 1999. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *AIJ* 110(2):241–273.

Stone, P.; Kaminka, G. A.; Kraus, S.; and Rosenschein, J. S. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *AAAI'10*.

Sugiyama, H.; Tsujioka, T.; and Murata, M. 2008. Coordination of rescue robots for real-time exploration over disaster areas. In *IEEE Symposium on Object Oriented Real-Time Distributed Computing*, 170–177. IEEE Computer Society.

Tambe, M. 1997. Towards flexible teamwork. *JAIR* 7:83–124.

Wurman, P. R.; D'Andrea, R.; and Mountz, M. 2007. Coordinating hundreds of cooperative, autonomous vehicles in warehouses. In *IAAI'07*, 1752–1759. AAAI Press.