Coordination and Adaptation in Impromptu Teams

Michael Bowling and **Peter McCracken**[*]
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada
{bowling,peterm}@cs.ualberta.ca

Abstract

Coordinating a team of autonomous agents is one of the major challenges in building effective multiagent systems. Many techniques have been devised for this problem, and coordinated teamwork has been demonstrated even in highly dynamic and adversarial environments. A key assumption of these techniques, though, is that the team members are developed together as a whole. In many multiagent scenarios, this assumption is violated. We study the problem of coordination in *impromptu teams*, where a team is composed of independent agents each unknown to the others. The team members have their own skills, models, strategies, and coordination mechanisms, and no external organization is imposed upon them. In particular, we propose two techniques, one adaptive and one predictive, for coordinating a single agent that joins an unknown team of existing agents. We experimentally evaluate these mechanisms in the robot soccer domain, while introducing useful baselines for evaluating the performance of impromptu teams. We show some encouraging success while demonstrating this is a very fertile area of research.

Introduction

Coordinating a team of agents to complete a joint task is a challenging problem. Despite the many difficulties, a large body of literature attests to the many successful techniques for building effective teams of agents. These techniques range from methods with very strong theoretical underpinnings (Grosz & Kraus 1996; Cohen & Levesque 1991) to practical heuristic methods (Parker 1998), as well as combinations of the two (Tambe 1997). Even robot soccer, where the dynamic and adversarial environment further complicates the problem, has seen many effective mechanisms proposed and implemented. Just within the RoboCup Small-Size League we see much variety (Tews & Wyeth 1999; Yoshimura *et al.* 2003; Bowling, Browning, & Veloso 2004).

A common assumption in these techniques is that the members of the team are all under the agent designer's control. Therefore agents can be designed knowing that their teammates will share their models of the world, decomposition hierarchies, arbitration mechanisms, and communication languages and protocols. In robot soccer, this is codified in Stone's "locker-room agreement": an implied set of conventions, protocols, strategies, and plans that each member adheres to, and assumes its teammates will as well (Stone & Veloso 1999). Some techniques have focused on issues of robustness where coordination can continue despite agent failures or even rogue team members. However, few techniques have looked at dropping this assumption entirely.

Another very active area of research is learning team behavior, particularly in the reinforcement learning paradigm (Sen, Sekaran, & Hale 1994; Claus & Boutilier 1998; Wang & Sandholm 2002). A common feature in this work is to focus on situations of "self-play", where the teammates are all using identical learning algorithms. This assumption is essentially a "locker-room agreement". Case-studies on particular games (Sandholm & Crites 1996; Stimpson & Goodrich 2003), such as iterated prisoner's dilemma, have sometimes dropped this assumption. It's not clear how to generalize these results beyond small strategic game settings, nor is it clear that the learning timescales are practical for real applications.

In many realistic settings, the agents in a team can come from multiple sources and therefore be designed independently. For example, robots participating in search and rescue operations are no more likely to come from a single organization than the human rescuers (*e.g.*, EMS, fire, police, military services). Some settings involve teams of agents only collaborating temporarily on smaller tasks, such as autonomous personal assistants scheduling meetings. The collaboration may be entirely impromptu, and the agents or agent designers may not have agreed on models, decompositions, subgoals, coordination mechanisms, or even a communication protocol. This is analogous to humans playing a friendly "pickup game", where players who may never have played together are divided into teams in an impromptu fashion. We will call these *impromptu teams* or *pickup teams*. Humans handle these situations with little difficulty; they often coordinate implicitly while also adapting to both the new teammates and new opponents. Most techniques for coordinating autonomous agents are not nearly so flexible.

In this paper we examine this problem of coordination in impromptu teams. We focus on the problem of a single agent, a *pickup player*, joining an existing unknown team of

[*]This research was supported by NSERC and the Alberta Ingenuity Fund.

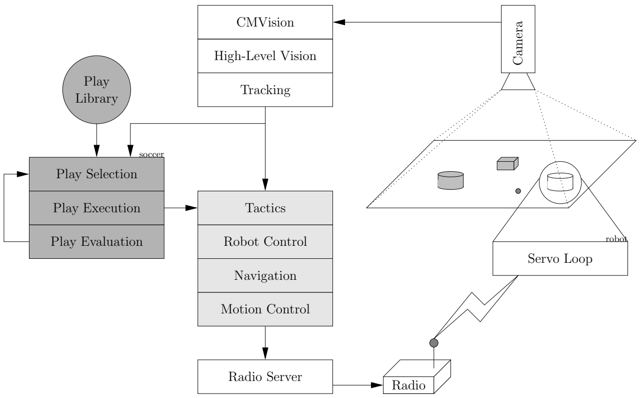


Figure 1: The CMDragons' team architecture.

agents, and present two techniques for coordination. Both techniques were implemented and evaluated in the context of robot soccer, building upon the CMDragons' team architecture (Browning *et al.* 2005). Since the techniques rely only on the generic concept of a team plan, or play, to provide impromptu coordination, this work is quite general. In fact, it is directly applicable to almost any domain where possible plans for coordinated action can be enumerated.

The rest of the paper is organized as follows. First we summarize the CMDragons' robot soccer team and play system, which forms the foundation for this work. We then describe how the system is adapted for coordinating an impromptu team. We evaluate our proposed techniques in terms of resulting team performance. We then conclude.

Play-Based Coordination

Our techniques for the coordination of impromptu teams are based upon and implemented in the CMDragons' team architecture. CMDragons is a robot soccer team that has competed in RoboCup's Small-Size League (SSL) from 2001 through 2004. An SSL team consists of five robots, each no larger than 18cm in diameter, playing soccer with a golf ball on a 4.9m by 3.4m field. The game is very fast; robots reach speeds as high as 2m/s and the ball exceeds speeds of 5m/s. This highly dynamic and adversarial environment makes for a very challenging team coordination problem. The CMDragons have been very successful in competition, reaching the semi-finals the past two years. An overview of the CMDragons'03 team architecture is shown in Figure 1. Full details of the various components are described in (Browning *et al.* 2005).

We are primarily interested in the darkly shaded components of Figure 1, called the *strategy* module. The strategy is responsible for coordinating the five robots, through the specification of *tactics*. Tactics are simple individual robot goals, which are implemented by the lightly shaded components using various shared robot skills, a fast path planning technique, and localized motion control algorithms. The strategy module's goals, then, are to define tactics for all five robots over time to bring about coordinated team play. In this section we overview the key components of this module, as it is the basis for coordination in our techniques for impromptu teams.

```
PLAY Two Attackers, Pass from Corner

APPLICABLE offense in_their_corner
DONE aborted !offense

ROLE 1
  pass 3
  mark 0 from_shot
ROLE 2
  block 320 900 -1
ROLE 3
  position_for_pass { R { B 1000 0 } ...
  receive_pass
  shoot A
ROLE 4
  defend_line { -1400 1150 } ...
```

Table 1: A complex play in the CMDragons' play language.

Plays. Plays are essentially team plans, defined in a simple text format. An example play is given in Table 1, which defines a team plan for crossing the ball from an opponent's corner to create a shot on goal. We will use this example throughout this section. The three key parts of a play are applicability conditions, termination conditions, and roles.

Applicability conditions, labeled by the keyword `APPLICABLE`, are like preconditions of operators in classical planning. They specify a conjunction of high-level predicates that must hold true before the play is selected for execution. This allows plays to specify whether they should be used when on offense or defense, or even more narrow situations such as a kick-off or if the ball is in a particular region of the field. In our example play, the play can only be executed when we are on offense and the ball is already in one of their corners (`offense` and `in_their_corner` are high-level predicates).

Termination conditions, labeled by the keyword `DONE`, specify possible outcomes of a play. Each termination condition is associated with a conjunction of predicates. The predicates are preceded by a result that identifies whether the play `succeeded`, `failed`, was `completed`, or `aborted`. If the conjunction of predicates is true, the play is terminated, and the play is tagged with the condition's result. These results affect the future selection of plays. In our example, the play is stopped and the result is considered `aborted` if at any time the team is no longer on offense. All plays contain implied termination conditions with appropriate results for the calls of the referee, such as goals and fouls.

The roles are the action component of the play, defining the behavior of the four non-goalie robots. A role consists of a list of tactics, which a robot will perform in sequence. This sequence is synchronized so that the entire team follows through their sequence of tactics simultaneously. Since tactics are heavily parameterized, many variations of individual and team behavior are possible. In the example play, the primary role involves passing the ball to the robot fulfilling role three, and then marking an opponent. Role three involves positioning itself to be open in a particular area of the field, receiving the pass, and then shooting toward the goal. The other two roles are defensive.

Play Execution. Instantiating a play into actual robot behavior involves some key executive decisions. Primarily, this involves role assignment, and possibly role switching during the play. Roles are expected to be ordered by priority and are filled in this order. Every tactic has an evaluation function that measures the suitability of a robot for achieving the goal based on the current state (*e.g.*, the robot's position). This function is used to select the most suitable robot for the first role. The second role is filled by choosing from the remaining team members, and so on. Role switching uses a similar mechanism with bias toward the currently assigned robot. Other details, such as synchronizing the team in the sequence, taking advantage of short-lived opportunities, and timeouts also must be handled by the play executor.

Play Selection. The last detail is how to determine which play to select. The team's human "coaches" can provide a team with a library, or playbook, of possible plays to be used during the game. At a restart or when a play terminates, a new play must be selected for execution. Applicability conditions can narrow down the choice, but there can still be more than one play available. We use this decision as an opportunity to adapt our team's behavior to our unknown adversary. Using the results of previously selected plays, *i.e.*, based on their termination conditions, the probability of selecting those plays in the future can be adjusted. This is done through a weight update algorithm based on techniques from the online learning literature. The full details are available in (Bowling, Browning, & Veloso 2004). Essentially, plays that result in the team scoring, receiving a penalty kick, or just completing are more likely to be selected again. Plays that result in the team being scored upon, or penalized, or aborted are less likely. This allows the team to focus on particular styles of play that are most successful for the current opponent.

In summary, CMDragons' play-based strategy system enables coordinated and adaptive team behavior. We now examine the situation of a pickup team and present a method for using plays as the basis for impromptu coordination.

Impromptu Teams

There are many possible ways to form an impromptu team. In this investigation we focus on teams where a single agent in a coordinated team is replaced by an independent teammate. All but one of the teammates, referred to as the *core team*, are able to coordinate and act as a normal team. The remaining agent, referred to as the *pickup player*, acts as a separate entity and is unable to communicate with its teammates. The core team and the pickup player are together referred to as the *impromptu team* or *pickup team*. Our focus will be on examining different ways in which the pickup player can participate on the team.

In order to more directly focus on the behavior of the pickup player, the core team was made oblivious to the independence of the pickup agent. We modified the CMDragons'03 team described above to control the pickup team. The core team views the pickup player as a normal member of the team and includes the player in its play and role selection. This means that this independent player is implicitly assigned a role in the current play, although this role is never communicated to the player. The core team, therefore, considers the pickup player as an option for passes, allows it to handle the ball, and even positions to receive passes from the player.

The pickup player, though, has a more complicated problem. Without any knowledge of how its own team is coordinating, it must choose to act in a way that both complements its own unknown team and plays well against its unknown opponent. We use plays as the primary, and very general, mechanism for determining how to act. The challenge for this player, then, is to (1) choose a play, and (2) choose a role on this play, both in a way that complements its own team's actions. Ideally, the pickup player will choose a play similar to the core team's own strategy, and then choose a role that matches the role the core team expects it to perform. In reality, the pickup player's playbook is not identical to the core team's playbook, or the core team may not even be using plays for coordination. To be effective, however, the pickup player needs only to choose a role that complements its teammates' behavior, whether they are using plays or not. In order to better focus on this more realistic situation in our experiments, we made sure that the plays available to the pickup players were not the same as those available to the core team.

Play Selection. For the pickup player, we tested two different implementations for play selection: an adaptive version and a predictive version. The adaptive pickup player uses learning to discover which plays work well with the core team. It uses the weight update play selection algorithm described in (Bowling, Browning, & Veloso 2004), and increases the likelihood of choosing plays that resulted in the pickup team doing well. After each context switch (possession of the ball changing, goal scored, *etc.*), the adaptive pickup player selects from a weighted list of applicable plays. Once the play is over, the player increases or decreases the weight of the play it chose based on the outcome of the play. In this way, the pickup player will be more likely to select plays that work well with whatever style the rest of its team uses.

The predictive pickup player attempts to explicitly predict which of its plays matches its team's style based on the current positions and trajectories of its teammates. After each context switch it is assumed that the core team may change its style of play. The predictive pickup player waits for a short period after each context switch, and then considers all of the applicable plays in its playbook. For each play, it computes a matching score based on how well its teammates match the roles of the play. The score component for each role is computed greedily; the highest priority role is paired with the player that most closely matches that role, based on the player's field location and current trajectory. That player cannot then be matched to any other roles. Exactly how a matching score is calculated for a player and role depends on the specific tactics involved in the role. The overall matching score for a play is the sum of the matching scores of each of its roles. The play with the best score is chosen by the pickup player as the play it believes best coordinates with the core team. For these experiments, a delay of two seconds was used between a context switch and when the predictive player selects its next play. For the duration of these two seconds, the predictive player chooses a play using the same mechanism as the adaptive player.

Role Selection. Once the pickup player has decided on the play it will execute, it must choose a role from that play. Role selection is performed in the same manner as described in the previous section. The pickup player implicitly assigns roles to the players on the pickup team, and in this process also chooses a role for itself. We make the assumption that the pickup player is never the goal keeper and is only deciding among the field roles.

Experiments

The purpose of our experiments in the pickup soccer domain is to show the effect that a single non-integrated player can make on a team's performance. We compare the pickup teams' performance to several interesting baseline tests.

Experimental Setup

The effectiveness of the pickup players was tested in simulation using the Ubersim simulator from Carnegie Mellon University (Browning & Tryzelaar 2003). Using a simulator allows games to be played in parallel on different computers, and with no human supervision, so more tests could be run in total. An automated referee was used to declare goals, throw-ins, and kick-offs. Each of the experiments consisted of a single game of soccer, between an impromptu team and an opponent team. The opponent team was a standard coordinated team, with no pickup player or imposed disadvantage. Its playbook contained a variety of plays so that it could adapt to different styles of play used by the impromptu team.

Our experiments test the effectiveness of the pickup and baseline players under different team conditions. We varied the style of pickup player, the playbook used by the core team, and the number of players on each team. For each combination, 100 games were simulated, each lasting 15 minutes of simulated game time. After each game, the positions of the robots were reset and the simulation was restarted, so that adaptation did not occur between games.

Different Playbooks. Ideally, a single pickup player should be able to integrate well with different teams, even if these teams have different styles of play. To simulate this we ran experiments using three different playbooks for the pickup team. Each of the playbooks used by the pickup team contained only two plays[1]: one applicable when attempting to score a goal, and one applicable when guarding against a goal. This means that in any given situation, the core team had only one play available to it. This simulates the most common situations seen in actual RoboCup play, where a team employs a single style of play throughout the game.

We selected three archetypes of play strategy for the core team. One of the playbooks contained only highly guarded plays (defensive playbook), one contained only aggressive plays (offensive playbook), and the third contained plays that were more balanced and less extreme than the other two playbooks (balanced playbook). For instance, in the defensive playbook, the goal-scoring play had a single player attempting to score a goal, and the remaining three field players attempting to guard their end. In contrast, the goal-scoring play from the offensive playbook had only a single player on defense, with the other three players either shooting or trying to positioning to catch a deflected ball. By emphasizing three very different styles of play, we hope to see how the play selection methods used by the pickup player can adapt to perform well with very different teammates.

The adaptive and predictive pickup agents, and also the opponent team, used a more complete playbook than was available to the core team. This playbook contained a variety of plays for both offensive and defensive situations, and did not necessarily contain plays similar to the ones used by the pickup team. Having multiple plays for different situations gives the pickup player options in how to coordinate with its unknown team.

Types of Pickup Player. In order to provide a baseline against which to compare the results of the teams with the two types of pickup players, we also tested three other team situations. The first and most rudimentary baseline we investigated was a team in which the core team are functional players, but the pickup player is an immobile "brick" that does not respond to any commands. This brick player can be detrimental to the core team, as it will be relied on to perform tactics and accept passes, yet it never reacts. Based on intuition, a pickup team should be able to outperform this baseline, as remaining motionless could always be an option for the pickup player. A second baseline is simply to have the pickup player be non-existent. It may seem at first that this baseline would be trivially outperformed by any reasonable pickup player. However, because the core team includes the pickup player in their role assignment, a pickup player that operates far differently than how the core team expects could degrade the team's performance. When the pickup player is missing, though, the core team has no expectations for the player. The final baseline used was a regular full-player team. This shows how well a pickup player could perform if it was perfectly coordinated with the rest of the team, being controlled by the same central strategy mechanisms and sharing the same playbook. Intuitively, a successful impromptu team would not be able to outperform a completely coordinated full-player team, but may achieve close performance. The five types of pickup teams used in our experiments are summarized in Table 2.

Number of Players. Normal SSL RoboCup teams have five players. However, in a normal five player game, the contribution of an individual player to a team is small. The small field also makes it difficult for the fifth team member to be effective, without being "in the way". This is supported

[1] In addition, all of the playbooks used in our experiments contained a standard set of plays for special situations, like kick-offs and penalty kicks.

Type	Description
regular	The ‘pickup’ player is coordinated with the core team during play and and shares the same playbook. This is a normal soccer team.
missing	The pickup player is not present on the field. This is equivalent to a ‘regular’ team with one fewer player.
brick	The pickup player is present on the field, but remains immobile and does not respond to any commands.
adapt	The pickup player does not communicate with its team and has a different playbook; it attempts to adapt which plays it chooses based on which plays worked well in the past.
predict	The pickup player does not communicate with its team and has a different playbook; it predicts which play its teammates are performing based on their locations and trajectories.

Table 2: A summary of the types of pickup player.

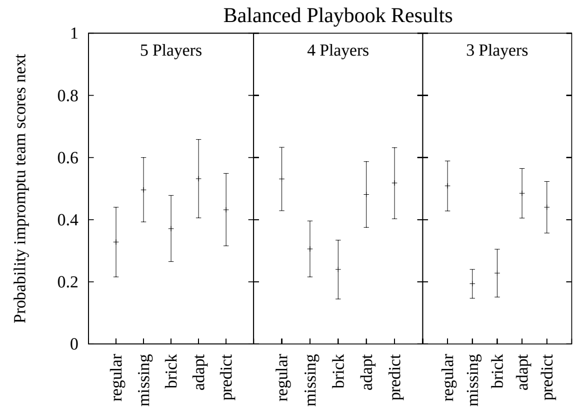
by the strong results of the missing team, as seen in Figure 2. In order to get a better understanding of the effect a pickup player has on the impromptu team’s performance, we ran experiments in which the number of players on each team was reduced. In games with four or three players on each team, coordination between teammates becomes increasingly important.

Team Performance

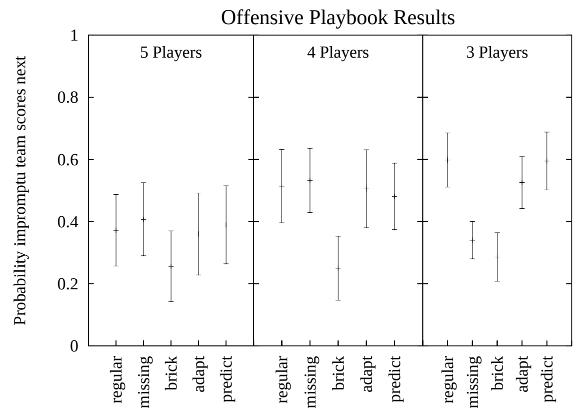
Figure 2 shows the results of the simulated games. The results are presented as the probability that the next goal in a game will be scored by the impromptu team. Each of the charts contains the data for using a particular playbook, and is partitioned according to the number of players on the teams. The error bars give a 95% confidence interval around the probabilities. Due to the stochastic nature of soccer and the fact that goals are relatively infrequent, the given probabilities possess a large degree of variance. Nevertheless, we believe that the general trend of the results is apparent and significant.

Performance of the Baseline Teams. The performance of the regular team is used as the comparison level for the other teams, since it shows the most “pure” performance of the playbook. The reason the regular team did not have a probability to score of 0.5 against the opponent team was due to the difference in playbooks between the impromptu team and the opponent team. The defensive playbook turned out to be quite strong against the opponent’s playbook, but the balanced and offensive playbooks did not perform as well.

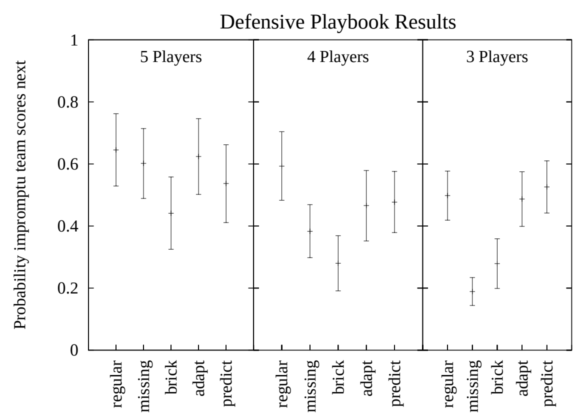
The first comparison to notice is the poor performance of the brick team compared to the regular coordinated team. In almost every case, the brick team performs statistically worse than the regular team (the exceptions are the five player games using the balanced and offensive playbook). The drastic difference suggests that the brick player must have been assigned crucial roles by the team, and its failure



(a) Using the balanced playbook.



(b) Using the offensive playbook.



(c) Using the defensive playbook.

Figure 2: Results from the simulation pickup games.

to perform those roles led to losses. This result suggests that coordinating roles on a team is critical. When a teammate is present on the field but does not do what is expected of it, the effect on team performance can be disastrous.

The performance of the missing team is also interesting. In five player games (*i.e.* games in which the missing team has four players), the missing team actually performed the same as, or better than, the regular team. It seems that the extra player in these circumstances must have been assigned to roles that actually hindered the performance of the team. For some strategies, the additional field congestion created by the extra player may be detrimental. As the number of players on the teams decreased, the effect of a missing player became more pronounced. In three player games, the missing team performed essentially the same as the brick team.

Performance of the Pickup Teams. Both the adaptive and the predictive pickup players performed similarly, and were not statistically different in any case. As discussed, a reasonable pickup player should be able to improve upon the performance of the baseline brick and missing strategies. We found this to be generally true of our pickup strategies. In almost all cases, both the adaptive and the predictive pickup players significantly outperformed the brick team, and in most cases also outperformed the missing team, especially when the number of players on each team was reduced. In the five player cases, notice that in the situations where the pickup teams did not match the performance of the missing team, the missing team itself had little drop in performance from the regular team. This suggests that the extra player in these situations is simply detrimental to the team, coordinated or not.

The regular team was included in our experiments as an upper bound baseline; it was meant to show how well a team could perform if it had complete coordination and a shared playbook. We expected the performance of the pickup teams to fall somewhere between the performance of the regular team and the baselines, brick and missing. We were surprised to find that the performance of the pickup teams using the adaptive and predictive players was competitive with the regular team. Even as the number of players on the team decreased, and thus the importance of each individual player increased, the adapt and predict teams still performed as well as the regular team. In the balanced playbook situation, the adaptive pickup team actually outperformed the coordinated five player team. This indicates an interesting potential of pickup players: to improve the performance of an existing team by adding knowledge in the form of new plays and roles.

Conclusion

In this paper we examined the problem of impromptu teams, where an agent's teammates are unknown and independent. We introduced two techniques for coordination in this setting, one adaptive and one predictive. We demonstrated that both techniques can make substantial improvements to the team's overall performance. The improvements were measured with respect to two baselines introduced for evaluating impromptu teams: a team with an immobile player and a team with an absent player. The performance of the pickup teams was even competitive with an unhindered full-player team. Our results demonstrate that impromptu coordination is a challenging problem, while giving encouraging evidence that such teamwork is possible.

References

Bowling, M.; Browning, B.; and Veloso, M. 2004. Plays as effective multiagent plans enabling opponent-adaptive play selection. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling*, 376–383.

Browning, B., and Tryzelaar, E. 2003. Ubersim: A multi-robot simulator for robot soccer. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, 948–949.

Browning, B.; Bruce, J.; Bowling, M.; and Veloso, M. 2005. STP: Skills, tactics and plays for multi-robot control in adversarial environments. *IEEE Journal of Control and Systems Engineering*. In Press.

Claus, C., and Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 746–752. Menlo Park, CA: AAAI Press.

Cohen, P., and Levesque, H. 1991. Teamwork. *Nous* 25:11–24.

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

Parker, L. E. 1998. ALLIANCE: An architecture for fault tolerant multi-robot cooperation. *IEEE Transactions on Robotics and Automation* 14(2):220–240.

Sandholm, T., and Crites, R. 1996. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* 37:147–166.

Sen, S.; Sekaran, M.; and Hale, J. 1994. Learning to coordinate without sharing information. In *Proceedings of the 13th National Conference on Artificial Intelligence*.

Stimpson, J. R., and Goodrich, M. A. 2003. Learning to cooperate in a social dilemma: A satisficing approach to bargaining. In *Proceedings of the Twentieth International Conference on Machine Learning*.

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

Tambe, M. 1997. Towards flexible teamwork. *Journal of Artificial Intelligence Research* 7:83–124.

Tews, A., and Wyeth, G. 1999. MAPS: A system for multi-agent coordination. *The International Journal of the Robotics Society of Japan* 14(1):37–50.

Wang, X., and Sandholm, T. 2002. Reinforcement learning to play an optimal nash equilibrium in team markov games. In *Advances in Neural Information Processing Systems 14*. MIT Press.

Yoshimura, K.; Barnes, N.; Rönnquist, R.; and Sonenberg, L. 2003. Towards real-time strategic teamwork: A robocup case study. In *RoboCup 2002: Robot Soccer World Cup VI*, volume 2752 of *Lecture Notes in Computer Science*. Springer. 342–350.