Global Planning from Local Perspective: An Implementation of Observation-based Plan Coordination in RoboCup Simulation Games

Cai Yunpeng, Chen Jiang, Yao Jinyi, and Li Shi

State Key Lab of Intelligent Technology and System,
Deparment of Computer Science and Technology,
Tsinghua University , Beijing, 100084, P.R.China
{Caiyp01, Chenjiang97, yjy01}
@mails.tsinghua.edu.cn
lishi@s1000e.cs.tsinghua.edu.cn

Abstract. This paper presents a method of implementing distributed planning in partially observable, low communication bandwidth environments such as RoboCup simulation games, namely, the method of global planning from local perspective. By concentrating planning on comprehending and maximizing global utility, the method solved the problem that individual effort might diverge against team performance even in cooperative agent groups. The homogeny of the agent decision architecture and the internal goal helps to create privities among agents, thus make the method applicable. This paper also introduces the application of this method in the defense system of Tsinghuaolus, the champion of RoboCup 2001 Robot Soccer World Cup Simulation League.

1 Introduction

Cooperation is an essential topic in the study of multi-agent systems, among which Distributed Planning is a challenging issue. In a multi-agent system, distributed plans can sometimes be formulated in a centralized manner by appointing a single agent as the planner. The planning agent executes the process of task decomposition, selection and assignment, and then distributes subtasks to each given member of the team. But centralized planning requires that at least one agent has the ability to possess accurate information of the global environment, and that the communication bandwidth is sufficient to accomplish task distribution in time. If the above condition is not satisfied, distributed planning is required.

The key to achieving cooperation is to establish agreement among the group of agents. This can be done either through communication or privities. Communication is the usual way to maintain coherence because it is more reliable and interpretable. Many methods such as Contract Net, Blackboard System and KQML [1] have been applied for cooperation protocols. And PGP (Partial Global

Planning) [1] has proven useful in dealing with distributed planning. However, in a real-time environment such as RoboCup simulation, sometimes the communication ability of an agent is not adeque to perform conversations before they reach an agreement. One way to coordinate without explicit communication is to allow agents to infer each others plans based on observations [1]. This is what we do in our simulation team.

Undoubtedly in circumstances like robot soccer where teamwork is so critical, agents must be designed to be altruist. Agents do not have any interest other than those relative to the common goal. But two main obstacles still exist preventing the group from acting as a unified entity. The first is the cognition difference: Agents' different observation on environment leads to different judgment about current mission of the team. The second is the divergence of individual effort against team performance. The maximization of individual contribution to the team does not certainly lead to best team performance.

In the design of the Tsinghuaeolus RoboCup simulation team, we introduced a method called Global Planning from Local Perspective to solve the second problem and to make an attempt to reduce the influence of the first problem to an acceptable degree. In this paper, we chose planning of defense as an example to demonstrate the implementation of this method in our RoboCup simulation team.

Defense plays an indispensable part both in real soccer game and in RoboCup. The defense task in RoboCup simulation game is obviously a process of distributed planning and dynamic coordination. The diversity of opponent attack strategies makes it impossible to set up a fixed defense strategy beforehand. Due to the limitation of communication bandwidth in RoboCup simulation environment, it is impossible to exchange enough information to perform centralized planning. Players cannot even tell the team what he himself is planning to do because the situation might change so fast that communication cannot catch up. In a word, a player has to decide what to do by himself according to his knowledge about the current state of the environment, but the team must keep a high level of consistency to create a solid defense system.

2 Main Points of Global Planning from Local Perspective

The essence of Global Planning from Local Perspective is to abstract the concept of global utility, then concentrate the purpose of planning on maximizing it. Sometimes global utility can be represented by a simple variable that can be noticed by all members, e.g., total output. But generally it is a quantity that exceeds the ability of a single agent to grasp directly. In RoboCup system, we cannot even have an objective description on what global utility is.

In fact, we model global utility instead of trying to describe it as it is. Since the decision of planning will eventually turn into behaviors of team members, global utility can be considered as the integration of influences each individual behavior contributes to the environment, which can be considered as individual utilities. Decompounding global utility into individual ones will attain great benefit because the latter are far easier to estimate.

Obtaining global utility from individual utilities is a process of evaluating a set of single agent behaviors and synthesizing them. During the procedure of synthesis, a set of operators has to be introduced to model the joint effect of behaviors. There are two basic kinds of synthesizers. Mutexs means that two behaviors should not be executed together, which can be expressed in mathematics as leading to a negative infinite compound utility. Interferences means that the compound utility P (AB) of action A with utility P (A) and action B with utility P (B) can be formulated as P (AB) = P (A) + P (B) + I (A, B), where I (A, B), may be either positive or negative, is a term that measure the influences of putting A and B together. I (A, B) is zero if A and B are independent.

Having defined individual and global utilities, we divide the process of planning into five steps:

- Step 1, Task Decomposition [1]: A series of subtasks that can be executed by a single agent are generated. Note that although fulfilling all subtasks means the accomplishment of the overall task, it is not necessary to do this all the time.
- Step 2, Generation of subtask-executor pairs: All subtasks are linked to each agent that is able to carry them out. Each pair is called an arrangement.
- Step 3, Evaluation of arrangement: Some evaluate functions are given to measure the performance of the subtask if a given arrangement is applied (that is, the executor really takes up the task). The value represents the individual utility of an arrangement.
- Step 4, Generation of scheme: A selection of arrangement (the execution set) is made based on the evaluation functions. The execution set includes all key subtasks that are required to achieve the global goal, and is considered the best plan to finish the task. The execution set is the one with the highest global utility among all feasible plans. In order to figure out the execution set, we use the algorithm of branch and bound. Mutexs, as a standard synthesizing operator, are used prior to interference operator so as to prevent redundancy or conflicts in combination of arrangements, thus speeding up searching.
- Step 5, Task assignment and execution: Agents pick out their own share of tasks from the execution set and carry it out.

The above procedure looks similar to centralized planning. The difference is that in centralized planning the procedure is carried out by only one planner, and in distributed planning it is carried out by all agents, each agent having a distinct viewpoint. In centralized planning, any inaccuracy of information may lead to fault in results. But in distributed planning, an agent only needs to guarantee that assignment concerning itself is proper, and the diversity of viewpoint provided more opportunity not to have all members making the same mistake. So, in partially observable environments, distributed planning might even achieve more reliability.

3 Reducing the Influence of Cognition Difference

So far we have not addressed the problem of cognition difference, which exists in every partially observable environment. The idea of global utility maximization involves assumptions on teammate behaviors, which indicates the risk of misunderstanding. When an agent abandons the action that can bring maximal individual utility, it assumes that its teammates will have corresponding actions to make up this loss, which is doubtful. The entire plan might be broken even if simply one agent acts singularly because of its distinctive viewpoint. This should be prevented. Indeed, we do not propose a radical solution to this problem. But we found that through proper choice of opponent modeling and good construction of evaluation functions, we can make our system less sensitive to slight differences in observation, hence increasing coherency in plan execution.

For convenience, in this section we call the agent executing the procedure of task planning the planning agent. But note that it is different from the one in centralized procedure, for all agents are planning agents.

As mentioned above, we made use of privities among agents in observation-based planning. Privities are knowledge of something secret shared between individuals, which is very important in cooperation among people. For example, in a human soccer team, many times team members may know how a player will pass the ball without any notification. To reach privities, some conditions must be satisfied:

- I, Agents have mutual beliefs [2] about the environment.
- II, All agents know the behavior model of others, so that they can to some degree predict teammates' future actions.
- III, Some public rules are set up as norms [4] to clarify roles of a given member in a given situation.

The latter two conditions can be realized in the structure of decision framework. What needs to be thoroughly considered is the first one.

Mutual beliefs can be represented in the following form:

If A and B mutually believe that p, then:

(a) A and B believe that p,

(b) A and B believe that (a). [2]

From this representation we found that the confirmation of mutual beliefs involving inferring others' internal states. For low bandwidth communication systems, to exchange parameters of internal states is not realistic. So, the ordinary manner of interpreting others' intention is through observation. A set of action styles is defined and characteristic behavior patterns are assigned to them. Through information collection an agent watches the activities of others and captures characteristic behaviors so as to comprehend their purpose. Thus mutual belief can be formed and coherence can be reached.

But the procedure of pattern recognition might be time consuming since agents' behaviors are not always distinct enough. Scarcity of information about teammate's intention might frequently emerge when an agent tries to make a decision. Four choices are offered to deal with this case.

The first one is to exclude all those whose intention is not clearly known from the cooperation scheme. This is advisable if the result of failed collaboration will be much more severe than working alone. But in environments where cooperation is emphasized, the choice is not adaptable.

In RoboCup simulation games, players perform decision-making almost every simulation cycle. Batches of new schemes come out replacing the old ones. So in most situations the fault of a single cycle won't have severe consequences. That means the environment is fault-tolerant. And the nature of soccer game made teamwork – even with poor coordination - more essential than individual practice. So agents must try to collaborate even with communication difficulties.

The second one is to choose the action with the largest probability (according to the agents expressive action) as the intention of that agent. The third one is to describe its intention in the form of a probability distribution, then to make a Bayesian decision. These two ways are popular in behavior modeling and do have great benefits. But both have to face the fact that the action inferred in this way is the one agents performed in the past instead of the one they are going to carry out in the future. In dynamic systems where agent might reconsider at any time, modeling in these two ways will cause bluntness in environment changes.

In global planning from local perspective, we chose the last approach - assumption of mutual knowledge. That is, the planning agent never tries to assume what teammates are intended to do before the entire plan is completed. Instead, it treats its own view of the situation as the global view, and assumes that teammates will perform those actions that the planning agent considers to achieve global utility maximization.

The introduction of assumed mutual knowledge exploits more collaboration ability from agents, but also raises more risk of cognition mistakes. But the homogeny of agent decision architecture and internal goal help to reduce it. As is mentioned, "just the fact that the agents share membership in a community provides a wealth of evidence upon which to infer mutual belief and common knowledge"[2]. Although in a distributed system with local information acquisition, agents have diversified partial knowledge of the external world, the fact that they share the same goal and use the same goal-directed information acquisition function makes it easy to have all the team members mutually grasp the key parameters of the current situations.

So, during the phase of planning the planning agent does not collect evidence of collaborative behaviors; instead, it collects those that shows that some agent failed to collaborate. Agents obviously acting inconsistently with the rest of the team will be wiped off from the candidate list of task executors. And new scheme will be brought out as a remedy. This can usually be implemented by adding some factor in the evaluation of subtasks.

Looking at partial knowledge as the picture of the entire environment is of course dangerous. But it is not always infeasible. Notice that partial knowledge acquired by an agent via its local information collecting mechanism is the true (although inaccurate and incomplete) mapping of the real world, so partial views differ from each other but the difference is bounded. If (and only if) the evaluation of individual and global utilities will not magnify these quantitative differences to qualitative ones, team coherency will be maintained.

Plainly speaking, the design of evaluation functions must take into account the bound of quantitative difference a variable may reach when observed by different agents. The mathematical characters of such functions should be continuous and smoothly varied. Further more, often when evaluation can proceed from various starting points, it is important to choose parameters that are easy to observe for all agents, as inputs to the function, instead of choosing hidden or trivial ones.

4 Application to Defense Play in RoboCup Simulation

Defense can be interpreted as a matter of positioning. Tsinghuaeolus uses a kind of Situation Based Strategy Positioning policy [3]. One of the most important positioning is the basic formation, which determines a player's position by standard role, ball position and ball controlling state. In the defense problem, basic formation serves as a standard, namely, a common rule, for role assignment and task evaluation.

Another important factor is the defense position sensitivity, which measure how dangerous it is if an opponent is staying in a given point. Defense position sensitivity for any point in on the field is known beforehand by all players of the team.

Four types of defense actions are defined:

- 1, Block: intersecting an opponent possessing the ball in the outer side of our goal, preventing him from pushing forward
- 2, Press: running after an opponent possessing the ball who is nearer to our goal, keeping threat on him
- 3, Mark: keeping up with an opponent without ball so that his teammates cannot pass the ball to him
- 4, Point Defend: staying at the basic formation position, this will benefit when a nearby teammate fails in 1 vs. 1 defense or when the team regains control of ball.

To simplify the problem, we set the rule that one defender cannot defend against two opponents at the same time and two defenders should not defend against the same opponent. The mutex operator is applied on both of these circumstances. The Interference operator is not used here.

There is an alternative where block and press are set as interference instead of mutex, providing a different defense strategy. Since either is ok, we do not discuss it further.

After decomposing a defense task into defending against each one of the attackers and staying at the basic formation point, arrangements are generated by linking every defender with every subtask. Then, the procedure goes to the evaluation of individual utilities.

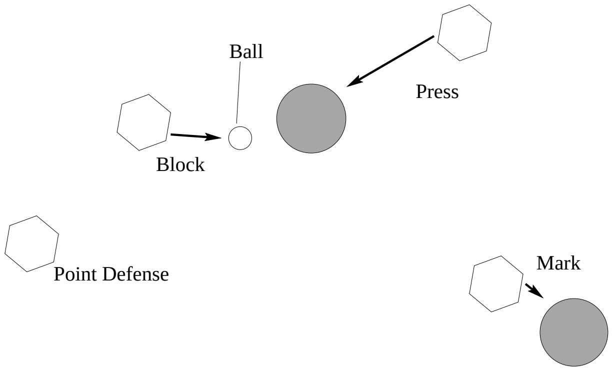

Fig. 1. Defend Actions

We defined one evaluation function for each type of action. Three variables are used as the input of each evaluation function: distance, measuring the interval from the player's current position to the defend point; deviation, measuring the distance from the defend point to the player's basic formation position; threat, namely the defend position sensitivity of the opponent's current position. The function output increases with threat and decreases with distance and deviation. For point defend, only threat is applied as input.

To attain the actual value of each function, we set some typical scenes, extract input values from them, and endow output values to the functions. The values are carefully tuned to keep mild variation. In this way a list of input and output data is created.

We use BP neural networks for encoding evaluation functions. The list generated above is used as the training set of the neural networks. After training, networks are tested to ensure that they fit requirements mentioned in the previous section, thus evaluation functions are defined.

The rest of the planning goes as the standard procedure presented above and no further discussion is needed.

One thing left is the generation of remedy scheme. Sometimes the team does fail to maintain coherency. For example, when an opponent is about equally far away from two defenders' current position and, at the same time, formation point, two arrangements will have similar evaluation and the two defenders may be in conflict. Our design takes advantage of the positive feedback that exists in the system. Since the ball will keep moving during the game, and opponents must move forward to perform attacking, either the basic formation points or the defend point will change in the next cycle. There is little chance that the two arrangements still keep the same evaluation. Once the difference between the two arrangements is obvious enough for both defenders to recognize, the prior defender will execute the defend action, moving nearer to the opponent, and the other one will move in another direction. So, the contrast is increased and the system is drawn to a status apart from the foregone dilemma, thus coherence is achieved again.

5 Experimental Results

We performed an experiment in order to discover to what extent agents coordinate internally and externally. We played a game against FC Portugal 2000 (December binary) in 3000 cycles. During the game all defense players (7 players altogether) recorded their plans and actions, then the log files were analyzed. The statistical result is below:

Table 1. Statistic data on defend planning

Arrangements after merge (1):	13830
Arrangements executed:	10056
Num of cycles the team is in defend state:	1569
Num of cycles the team have dissidence in arrangement:	1219
Num of executed arrangement per cycle:	6.4
Num of dissidence per cycle:	2.4
Num of cycles the team conflict in executions (2):	70

(1) Arrangements generated at the same time by different planners and with the same decision are considered the same one

(2) Including both defending by too many players and missing a attacker that need being defended against

From the data we can conclude that for most of the time agents have dissidence in planning, but they have reached agreement on the major part of the plan (at least 4 arrangements in 6.4 is agreed by all planners in average). Another point to recognize is that most of time dissidence in planning does not lead to incoherency in execution. Most of the time dissidence is about whether a player should mark an attacker in the position near a turning point or stay at the formation point, which is hard to judge but scarcely affects the performance. For over 95 percent of the time the team acts uniformly despites of dissidence in 77 percent of the time, which shows that our method have perfectly reached its design goal.

In RoboCup 2001, Tsinghuaeolus won the championship, scoring 90 goals, while conceding only 1 goal. The defense faced the challenge of diversified strong attack styles from various teams and functioned well, leaving few chances for opponents, while not involving too many players in defense.

6 Relative Works

In paper [5], Lesser cast a survey on recent progress of multiagent research and put forward some principles that are useful to build a multiagent system. During explaining his viewpoint he proposed to move the focus from the performance of individual agents to the properties and character of the aggregate behavior of agents. The basis of Global Planning from Local Perspective is highly corresponding to this proposal.

In RoboCup fields, one of the hot topics relative to coordination is to achieve cooperation by learning (e.g.[6],[7],[8]).And many other teams tried to coordinate via communication. Paper [9] presented an implementation of this approach in Middle Sized Robot Games.

Peter Stone proposed the TPOT-RL algorithm and the Flexible Teamwork architecture [6][7]. In his architecture teammates are considered as part of dynamic environments, and coordination is achieved with the success of agents adapting the environments through reinforcement learning. Our method differs from this approach in terms that we do not design an algorithm to learn a decision structure; instead, we provide a fixed but general framework beforehand. The flexibility of the structure lies in its ability to incorporate varied knowledge (that is, evaluation function of different forms). Both human knowledge and machine learning results can become elements of the decision module.

7 Conclusions

Practice results show that merely by introducing the concept of global utility can we achieve a significant enhancement on system performance. The design of evaluation functions enhances performance mainly by decreasing the probability of cognition fallacies and dead locks, hence making performance more stable. Global Planning from Local Perspective is designed for fault-tolerant systems. It contributes mainly by speeding up the period of planning to adapt fiercely changing environments, and exploiting the potential hidden in agents' reactions.

Further more, the evaluation functions has to be contrived to fit specific environments. More skills other than those described above must be employed. It might be worthwhile to explore the possibility to obtain a good evaluation function by online learning.

References

1. Gerhard Weiss . Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. pp.83-111, pp.139-156. The MIT Press,1999
2. Stephen A.Long and Albert C.Esterline . Fuzzy BDI Architecture for social agents. North Carolina A&T Univ. Southeastcon 2000. Proceedings of the IEEE , 2000 pp.68-74
3. Lus Paulo Reis and Nuno Lau. FC Portugal Team Description: RoboCup 2000 Simulation League Champion.Univ. of Porto & Univ. of Aveiro. In Stone P., Balch

T., and Kaetzschmar G., editors, RoboCup 2000:Robot Soccer World Cup IV. pp.29-41,Springer Verlag, Berlin, 2001

4. Towards socially sophisticated BDI agents .Dignum, F.; Morley, D.; Sonenberg, E.A.; Cavedon, L. MultiAgent Systems, 2000. Proceedings. Fourth International Conference on , 2000 pp.111-118
5. Lesser, V.R. Cooperative multiagent systems: a personal view of the state of the art. Knowledge and Data Engineering, IEEE Transactions on, Volume: 11 Issue: 1,Jan.-Feb. 1999 pp.133-142
6. Peter Stone. Layered Learning in Multi-Agent Systems. PhD Thesis, Computer Science Dep.,Carnegie Mellon University, December 1998
7. Peter Stone and Manuela Veloso. Layered Learning and Flexible Teamwork in RoboCup Simulation Agents. CMU. In Veloso M., Pagello E., and Kitano H., editors, RoboCup-99: Robot Soccer World Cup III. pp.495-508. Springer Verlag, Berlin, 2000
8. Masayuki Ohta. Learning in Cooperative Behaviors in RoboCup Agents. Tokyo Institute of Technology. In Hiroaki Kitano, editor, RoboCup-97: Robot Soccer World Cup I. pp.412-419. Springer Verlag, Berlin, 1998
9. K. Yokota, K.Ozaki, et al. Cooperative Team Play Based on Communication. Utsunomiya Univ. and Toyo Univ. In Minoru Asada, editor, RoboCup-98: Robot Soccer World Cup II. Proceedings of the second RoboCup Workshop. pp.491-496. Paris, 1998.