Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 46 (2015) 396 – 401

Procedia Computer Science

International Conference on Information and Communication Technologies (ICICT 2014)

Plan Restructuring in Multi Agent Planning

Satyendra Singh Chouhan[a,]*, Rajdeep Niyogi[a]

[a]*Computer Science and Engineering Department, IIT Roorkee, Roorkee-247667, INDIA*

Abstract

In this paper we consider a multi agent planning problem where given a set of plans of the individual agents that may lead to conflict during execution, the task is to find a conflict-free plan for the agents. For this we give a formal model of the multi agent planning problem (MAPP) and give a description of the multi agent path planning. We demonstrate this technique in a grid world domain. The experimental results show promise for our method.

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)

Keywords: Multi-agent planning; Plan restructuring; grid world domain;

1. Introduction

Planning is an important feature in the design of intelligent systems. Planning precede acting. In multi agent planning domains typically there are several agents that plan and act together to share resources, activities, and goals[1,2,3]. Multi-agent planning is a major issue for the distributed AI and multi-agent community. In this paper, we assume that a state is completely known, actions are deterministic, and the environment is static. But even in such a setting there may be a possibility of plan failure.

For example, consider a two agent robot domain[4] where both agents are in a hallway and they want to move into the same room through a single narrow doorway through which only one agent can pass at a time. The agents can perform the action "GO" to go through the doorway, but this action will only succeed if the other agent is not trying

* Corresponding author. Tel.: +91-895-422-1599.
E-mail address: satycdec@iitr.ernet.in

Peer-review under responsibility of organizing committee of the International Conference on Information and Communication Technologies (ICICT 2014)
doi:10.1016/j.procs.2015.02.036

to go through the doorway at the same time. To achieve the individual goals of both agents one of the agents has to wait till other agent passes through the door. Thus there should be coordination between their actions.

Multi agent planning is an interesting research area. In the literature different multi agent planning techniques have been suggested. Some authors[7,11] consider multi agent planning as a process of finding joint plan by coordination of different agents. In some works[5,17,18] multi agent plans are viewed as combination of individual plans of different agents. A recent work[10, 15] considers solving a multi agent planning problem by dividing the planning problem into sub problems that are assigned to multiple agents. These works use the inherent structure of planning domains to represent it as a multi agent planning problem. Our work focuses on the problems where multiple agents work concurrently to achieve the individual goals.

In section 2 we describe the framework for multi agent planning problem. Section 3 describes the plan restructuring approach to solve multi agent planning. Section 4 gives the implementation results. Section 5 describes related work and we finally conclude in section 6.

2. A Framework for Multi Agent planning problem

2.1 (Definition1) Multi agent planning domain[4]

A multi agent planning domain D is a tuple $\{P, n, A_{i=1 \text{ to } n}, R\}$, where P is a finite set of propositions and S is set of states, $S \subseteq 2^P$

n: number of agents

A_i: finite set of actions of agent i

$R \subseteq S \times A \times S$ is a non-deterministic transition relation where $A = A_1 \times A_2 \times \ldots\ldots \times A_n$

R satisfies the following condition:

If $(s, a, s') \in R$ and $(s, b, s'') \in R$ then $\forall i$ there exists $s''' \in S$, $(s, (a_1 \ldots a_{i-1}, b_i, a_{i+1} \ldots\ldots a_n), s''') \in R$

In addition, let $Act_i(s) \subseteq A_i$ be the set of actions that are executable in state 's', i.e.,

$Act_i(s) = \{a_i \in A_i \mid \exists\, (s, (\ldots.a_i\ldots), s') \in R\}$

2.2 (Definition 2) Multi agent planning problem[4]

For a multi agent planning domain $D = \{P, n, A_{i=1 \text{ to } n}, R\}$, a multi agent planning problem P is a tuple $\{D, I, G_{i=1 \text{ to } n}\}$, where I is the set of initial states and $I \subseteq S$ and $G_i \subseteq S$ is the set of goal states for agent i.

When agents execute their individual plans conflicts may arise. Thus we need to restructure the individual plans.

2.3 (Definition 3) Multi agent path planning problem is a multi-agent planning problem where the states are positions or coordinates in a given environment.

A path for a given plan consists of a sequence of states. In the multi agent environment, coordination is needed to achieve conflict free paths (plans) for all agents. Conflict free plans can be achieved by restructuring any one or more individual plans. For example, consider a grid world domain of size 5×5. Some cells are obstacles and an agent cannot occupy these cells. Each cell can be occupied by only one agent at any point of time. In multi agent system the knowledge of agents is limited and restricted to only local information. In the grid world domain, an agent knows information about at most four adjacent cells. Every agent can perform actions from the action set {move_left, move_right, move_down, move_up, wait}. Each cell is assigned a unique number that determines the state of an agent.

In the following we show two possible situations where optimal individual plans may lead to failure during execution.

A. (Case1) Point collision

Let agent x and y's initial and final states be (S_{11}, S_{15}) and (S_3, S_{23}) respectively. Optimal individual plans of agent x and y are $\pi_x = \{s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$ and $\pi_y = \{s_3, s_8, s_{13}, s_{18}, s_{23}\}$. In this case there is a conflict at state s_{13}.

Both agents will try to occupy the same cell at the same time (see Fig. 2.).

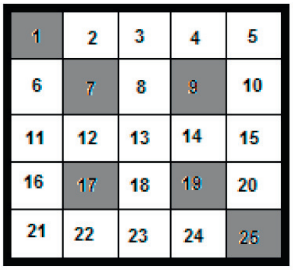

Fig. 1.Grid world domain

B. (*Case2*) *Plan obstruction*

Let agent x and y's initial and final states be (S_6, S_{15}) and (S_5, S_{18}) respectively. Optimal Individual plans of agents x and y are $\pi_x = \{s_6, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$ and $\pi_y = \{s_5, s_{10}, s_{15}, s_{14}, s_{13}, S_{18}\}$. In this case, both plans will fail due to the obstruction of other plan (See Fig. 2.).

Fig. 2. (a) Point collision; (b) Plan obstruction

3. Plan Restructuring Approach

In this section we suggest a method for plan restructuring for a multi agent path planning problem. Let $p=a_1,a_2,..a_k$ be a plan executed by agent 1 at the initial state s_0. This results in a sequence of states: $s_0,s_1,..,s_k$; the initial state is also considered in the sequence. Let $p' = b_1, b_2,..b_k$ be another plan executed by agent 2 at the initial state s_0. Let the resulting sequence of states be: $s'_0, s'_1,..,s'_k$. For the plans p,p' to be conflicting there is a common point in the sequences for which the states are same. Formally, $\exists j,\ 0 < j \leq k\ \ s_j = s_{j'}$. At this point ($s_j$) in the sequence of say plan p, we change the original action that was executed at s_{j-1}. We can also choose the point as $s_{j'}$ in the plan p' and make the change accordingly. As a result of this change, the sequence changes; so we need to iteratively change the sequence (and thus the plan).

Algorithm:

Input: π_x, π_y // individual plans of agent x & y.
Output: π'_x, π'_y // restructured plans
Joint_planner(π_x, π_y)
Pos ← **Conflict_check**(π_x, π_y);
If pos =null then output "no restructuring required "and exit;
{π'_x , π'_y} ← **Resolve**(π_x, π_y , pos);
If {π'_x , π'_y}=null then no solution exists and exit;
If there are still conflicts **Conflict_check** (π'_x, π'_y)! = null
then go to step 3.
Output restructured plan π'_x, π'_y

Algorithm's explanation:

There are two important methods, **Conflict_check**(π_x, π_y) and **Resolve**(π_x, π_y , pos)**. Conflict_check** (π_x, π_y) determines two types of conflicts, i.e. point collision and plan obstruction. Point collision is checked by comparing plans, such that if there is a common state after the same number of actions then collision will occur. If it is so, then conflict check returns the index of the same state variable in both plans. Plan obstruction is checked by determining a common sub-plan on which both are moving in opposite directions.

Conflict_check returns an index of state variable corresponding to the point where plan obstruction occurs. In **Resolve** (π_x, π_y, pos) method we randomly choose one of the plan and restructure the plan from (pos-1) such that the collision does not occur at next state. If it is not possible, then it selects another plan to restructure.

For example in point collision case let the optimal individual plan of agents x and y be $\pi_x = \{s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$ and $\pi_y = \{s_3, s_8, s_{13}, s_{18}, s_{23}\}$. These plans are forwarded to the planner by agents. The planner checks for possible conflicts and restructures the plans accordingly. In this case the conflict is at state s_{13}. The planner restructures the plan of any one of the agents (let's say x) to $\pi_x = \{s_{11}, s_{12}, s_{12}, s_{13}, s_{14}, s_{15}\}$ by assigning a wait action at s_{12}. Now it iteratively checks if there are any more conflicts. If no, it forwards the plan to the agents. Solution plans would be $\pi_x = \{s_{11}, s_{12}, s_{12}, s_{13}, s_{14}, s_{15}\}$ and $\pi_y = \{s_3, s_8, s_{13}, s_{18}, s_{23}\}$.

In plan obstruction case, let the optimal individual plan of agents x and y be $\pi_x = \{s_6, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$ and $\pi_y = \{s_5, s_{10}, s_{15}, s_{14}, s_{13}, S_{18}\}$. In this case planner iteratively re plans one of the agent's plan and check conflicts. In this situation there is a conflict due to state s_{13} and s_{14}. Planner iteratively restructures one of the plans. In this situation planner will restructure the plan of x like $\{s_6, s_{11}, s_{12}, s_{13}, s_8, s_3, s_4, s_5, s_{10}, s_{15}\}$. And planner again will check if there is any conflict in $\pi_x = \{s_{06}, s_{11}, s_{12}, s_{13}, s_{18}, s_{13}, s_{14}, s_{15}\}$ and $\pi_y = \{s_5, s_{10}, s_{15}, s_{14}, s_{13}, s_{18}\}$. If there is any conflict again, it will select one of plan and try to restructure till there is no conflicts. Solution would be $\pi_x = \{s_{06}, s_{11}, s_{12}, s_{13}, s_{18}, s_{13}, s_{14}, s_{15}\}$ and $\pi_y = \{s_5, s_{10}, s_{15}, s_{14}, s_{13}, S_{18}\}$.

4. Implementation results

The purpose of our experiment is to show the need of plan restructuring in multi agent planning for the grid world domain. We have implemented the above algorithm in JAVA. All the experiments are performed on 2.10 GHz Intel Pentium I3 with 3 GB RAM Windows 7 operating system. We have studied and observed the different cases with individual plan of agents where plan interaction might cause failure of individual plans. We have also shown the restructured plan generated by the planner with time taken to generate it.

Table1 shows the experimental results for the different cases. In table1 columns two and three show the individual plans of the agents. The fourth column shows if there is any conflict between the individual plans. The fifth column shows the restructured plan after resolving the conflict. Sixth column shows the time taken by the simulator program for conflict resolution.

5. Related work

In paper[8] the authors proposed a multi agent planning model based on plan reuse and an algorithm based on plan reuse. The task is to assign a subset of the public goals to each agent. Each agent iteratively solves its problem by receiving plans from other agent. In paper[12] the authors compare two strategies to come up with stable plan for agent. It suggests that plan stability can be achieved by re-planning (planning from scratch) or plan repairing (modifying an existing plan).

In paper[6] the authors proposed an approach of multi-agent (MA) plan repair (MA-REPAIR), based on multi-agent planning (MA-STRIPS) [10]. MA-STRIPS is an extension of the classical STRIPS-based planning approach to multi agent planning for teamwork and coordination. According to the MA-REPAIR approach, the multi-agent team computes a team plan using a fully decentralized MA-STRIPS planning algorithm, and subsequently executes the plan, while at the same time monitoring possible failures of plan execution. Upon an occurrence of such a failure, the team stops execution and invokes a plan repair algorithm and fixes the failed joint plan in order to reach a joint goal state from the state in which the failure occurred[13]. When agents work in dynamic environments there may be condition for which some states and goals of agents may be modified. So in such situations existing plans should be modified and re-planning is required[14].

In paper[9] a general model of single agent re-planning is presented. The paper describes three re-planning paradigms that are distinguished by the constraints that they are bound to satisfy during the re planning process. These are re-planning as restart, re-planning to reduce computation, and re-planning for multi-agent scenarios. Some work has been done in the area of continual planning[16]. This technique consists of interleaving planning with plan execution and execution monitoring.

6. Conclusion and Future Work

In this paper we have presented a plan restructuring approach for multi agent path planning. We have illustrated our approach on a grid world domain. The results are satisfactory. As part of future work we wish to apply our algorithm on robot path planning and also study the applicability of the approach for other complex domains.

Table1. Agent's individual plans and restructured plans

Sr. No.	Agent x's plan	Agent y's plan	Conflict	Restructured plan	Time (ms)
1	$\pi_x = \{s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$	$\pi_y = \{s_{03}, s_{08}, s_{13}, s_{18}, s_{23}\}$	Yes	$\pi_x = \{s_{11}, s_{12}, s_{12}, s_{13}, s_{14}, s_{15}\}$ $\pi_y = \{s_{03}, s_{08}, s_{13}, s_{18}, s_{23}\}$	0.437
2	$\pi_x = \{s_{11}, s_{16}, s_{21}, s_{22}, s_{23}, s_{24}\}$	$\pi_y = \{s_{03}, s_{08}, s_{13}, s_{18}, s_{23}\}$	Yes	$\pi_x = \{s_{11}, s_{16}, s_{21}, s_{22}, s_{23}, s_{24}\}$ $\pi_y = \{s_{03}, s_{08}, s_{13}, s_{18}, s_{18}, s_{23}\}$	0.396
3	$\pi_x = \{s_{06}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}\}$	$\pi_y = \{s_{05}, s_{10}, s_{15}, s_{14}, s_{13}, s_{18}\}$	Yes	$\pi_x = \{s_{06}, s_{11}, s_{12}, s_{13}, s_{18}, s_{13}, s_{14}, s_{15}\}$ $\pi_y = \{s_{05}, s_{10}, s_{15}, s_{14}, s_{13}, s_{18}\}$	0.584
4	$\pi_x = \{s_{02}, s_{03}, s_{08}, s_{13}, s_{18}, s_{23}\}$	$\pi_y = \{s_{21}, s_{22}, s_{23}, s_{18}, s_{13}, s_{14}\}$	Yes	$\pi_x = \{s_{02}, s_{03}, s_{08}, s_{13}, s_{12}, s_{13}, s_{18}, s_{23}\}$ $\pi_y = \{s_{21}, s_{22}, s_{23}, s_{18}, s_{13}, s_{14}\}$	0.421
5	$\pi_x = \{s_{16}, s_{21}, s_{22}, s_{23}, s_{24}\}$	$\pi_y = \{s_{14}, s_{13}, s_{18}, s_{23}, s_{24}\}$	Yes	$\pi_x = \{s_{16}, s_{21}, s_{22}, s_{22}, s_{23}, s_{24}\}$ $\pi_y = \{s_{14}, s_{13}, s_{18}, s_{23}, s_{24}\}$	0.365
6	$\pi_x = \{s_{08}, s_{13}, s_{14}, s_{15}, s_{20}\}$	$\pi_y = \{s_{15}, s_{14}, s_{13}, s_{18}, s_{23}\}$	Yes	$\pi_x = \{s_{08}, s_{13}, s_{1}, s_{13}, s_{14}, s_{15}, s_{20}\}$ $\pi_y = \{s_{15}, s_{14}, s_{13}, s_{18}, s_{23}\}$	0.788
7	$\pi_x = \{s_{10}, s_{05}, s_{04}, s_{03}, s_{08}\}$	$\pi_y = \{s_{03}, s_{04}, s_{05}, s_{10}\}$	Yes	$\pi_x = \{s_{10}, s_{05}, s_{04}, s_{03}, s_{08}\}$ $\pi_y = \{s_{03}, s_{02}, s_{02}, s_{02}\ s_{04}, s_{05}, s_{10}\}$	1.02
8	$\pi_x = \{s_{20}, s_{15}, s_{10}, s_{05}, s_{10}\}$	$\pi_y = \{s_{05}, s_{10}, s_{15}, s_{14}, s_{13}\}$	Yes	$\pi_x = \{s_{20}, s_{20}, s_{20}, s_{15}, s_{10}, s_{05}, s_{10}\}$ $\pi_y = \{s_{05}, s_{10}, s_{15}, s_{14}, s_{13}\}$	0.738
9	$\pi_x = \{s_{04}, s_{03}, s_{08}, s_{13}, s_{18}\}$	$\pi_y = \{s_{11}, s_{12}, s_{13}, s_{08}, s_{03}, s_{02}\}$	Yes	$\pi_x = \{s_{04}, s_{03}, s_{08}, s_{13}, s_{18}\}$ $\pi_y = \{s_{11}, s_{12}, s_{13}, s_{14}, s_{13}, s_{08}, s_{03}, s_{02}\}$	0.780
10	$\pi_x = \{s_{24}, s_{23}, s_{22}, s_{21}, s_{16}, s_{11}\}$	$\pi_y = \{s_{11}, s_{16}, s_{21}, s_{22}, s_{23}, s_{24}\}$	Yes	$\pi_x = \{s_{24}, s_{23}, s_{18}, s_{18}, s_{18}\ s_{23}, s_{22}, s_{21}, s_{16}, s_{11}\}$ $\pi_y = \{s_{11}, s_{16}, s_{21}, s_{22}, s_{23}, s_{24}\}$	1.037

References

1. Ghallab M, Nau D, Traverso P. Automated Planning: Theory and Practice. *Morgan Kaufmann, San Francisco* 2004.
2. Weerdt M, Clement B. Introduction to Planning in Multi-agent Systems. *In Journal of Multi agent Grid System;* 2009; 345-355.
3. Durfee EH. Distributed Problem Solving and Planning. *In Multi agent Systems: A Modern Approach to Distributed Artificial Intelligence Springer*; 1999; 121–164.
4. Bowling M, Jensen R, Veloso M. A Formalization of equilibria for multi agent planning. *In Proceedings of International Joint Conference on Artificial Intelligence* (IJCAI); 2003;1460–1462

5. Larbi RB, Konieczny S, P. Marquis. Extending classical planning to the multi-agent case: A game-theoretic approach. *In ECSQARU Lecture Notes in Computer Science, Springer*; 2007; 4724; 731–742.
6. Komenda A, Novák P. Decentralized multi-agent plan repair in dynamic environments. *In Proceedings of Autonomous agent multi-agent system* (AAMAS); 2012; 1239–1240.
7. Nissim R, Brafman R I, Domshlak C. A general fully distributed multi-agent planning algorithm. *In Proceedings of Autonomous agent multi-agent system (AAMAS)*; 2010; 1323–1330.
8. Borrajo D. Plan Sharing for Multi-Agent Planning. *In proceeding of distributed and Multi-Agent Planning workshop* (ICAPS-DMAP); 2013; 57-65.
9. Talamadupulay K, Smith D E, Cushingy W, Kambhampati S. A Theory of Intra-Agent Re-planning. *In proceeding of Distributed and Multi-Agent Planning workshop* (ICAPS DMAP); 2013; 48-56.
10. Brafman R I, Domshlak C. From one to many: Planning for loosely coupled multi agent systems. *In Proceedings of International Conference on Automated Planning and Scheduling* (ICAPS); 2008; 28–35.
11. Brenner M. Planning for multi-agent environments: From individual perceptions to coordinated execution. *In Workshop of Multi agent Planning and Scheduling, (ICAPS)*; 2005; 80-88.
12. Fox M, Gerevini A, Long D, Serina I. Plan stability: Re-planning versus plan repair. *In Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling* (ICAPS); 2006; 212–221.
13. Komenda A, Novák P, Michal P. Decentralized How to Repair Multi-agent Plans: Experimental Approach. *In proceeding of Distributed and Multi-Agent Planning workshop* (ICAPS DMAP); 2013; 66-74.
14. Cushing W, Kambhampati S. Re-planning: A New Perspective. *In Proceedings of the Sixteenth International Conference on Automated Planning and Scheduling* (ICAPS); 2005.
15. Nissim R, Brafman R. Cost-Optimal Planning by Self-Interested Agents. *In proceeding of Distributed and Multi-Agent Planning workshop* (ICAPS DMAP); 2013; 1-7.
16. Brenner M, Nebel B. Continual planning and acting in dynamic multi agent environments. *In Proceedings of Autonomous agent multi-agent system* (AAMAS); 2009; 297-331.
17. Durfee EH, Lesser V R. Partial global planning: A coordination framework for distributed hypothesis formation. *IEEE Transactions on Systems, Man, and Cybernetics;* 1991; 21, 5; 1167–1183.
18. Boutilier C, brafman R. Partial-order planning with concurrent interacting actions. *Journal of Artificial Intelligence Research*; 2001; 14; 105–136.