An Efficient Approach to Do Multi-agent Planning

Lin Zeng[1, 2]
[1]School of Information Science and Technology
Sun Yat-sen University
Guangzhou, China
zsuzenglin@163.com

Lei Li[1]
[2]College of Computer Science and Information Technology
Hezhou University
Hezhou, China

***Abstract*—Multi-agent planning is a difficult, yet under investigated class of planning problems. In the cooperative assumption, agents plan to achieve their individual goals independently, communicate with each other by continually sending and receiving messages from other agents. However, finding a globally consistent plan requires agents to interact with each other, which is time-consuming. In this paper, we describe a general framework that extracts the graph structure from causal constraints between agents in solving multi-agent planning problems to reduce the effort of repeatedly searching agents' consistent local plans and we theoretical analytically show that our approach can efficiently solve multi-agent planning problems with very complicated coupling level.**

Keywords—AI Planning; Multi-agent System; Graph Structure

I. INTRODUCTION

Many real-world systems are comprised of multiple agents, and even the case closest to the cooperative multi-agent (MA) systems, differs substantially from the single agent setting. The difficult aspect of multi-agent planning problems is to coordinate actions of individual agents such that goals are achieved efficiently [1][2]. Agents in a multi-agent system are expected to have rich internal structure, providing them with a large degree of operational autonomy. On the other hand, the agents comprising a multi-agent system may be unwilling to expose their internals to the others in the system, in the case of short-term cooperation, for example. Thus, planning for such systems, these properties indicate that it is often impractical to ignore this expected independence between agents.

There have been many approaches designed to solve the multi-agent planning problems. For example, use planning graph to handle the possible relationships between agents' individual goals [3]. Dimopoulos and Moraitis dealed with coordination and cooperation in the multiple agents planning [4]. The work in [5] combines graph planning with distributed constraint satisfaction techniques to solve the multi-agent replanning problems. Brafman and Domshlak proposed a novel notion of multi-agent Planning as CSP+Planning [6], which was later extended to a fully distributed multi-agent planning approach called Planning-First(PF) [7].

Despite the success of previous systems, either only contour algorithm description, or finding a consistent global plan is time-consuming since agents have to repeatedly interact with each other through sending and receiving messages. The communication in the MA planning is to ensure that whether the public action sequences of each agent can be locally consistent.

In this paper, we present a novel approach, called to solving multi-agent planning problems based on conducting iterative backward search. Unlike other multi-agent planning technologies, our approach is capable of solving very tightly coupled multi-agent planning problem efficiently, which often makes multi-agent system become exponentially harder than loosely coupled systems

II. PROBLEM FORMULATION

We assume that the cooperative multi-agent systems is based on a formalization which is minimally extends STRIPS-style language model under the problem of deterministic domains and fully-observable information. In the following we briefly introduce the definitions and notation in multi-agent planning problem that we will rely on in the rest of this paper.

A. Problem Formulation

We consider planning for multi-agent system where agents must coordinate their efforts to the set of global goals. Specifically, we restrict our attention to multi-agent's STRIPS formalism which slightly extended to STRIPS representation of classical planning language. Intuitively, the concept of multi-agent planning is such that there can be more than one independent agent in a multi-agent planning problem which each of these agents must make local plan independently and coordinate their individual plans to achieve global goals and given a specification of their initial state and own actions. In MA-STRIPS model, states of the world are described by a subset of fluents and action is defined as tuples $a = < pre(a), eff(a) >$ where $pre(a)$ and $eff(a)$ denote the precondition and the effect (add and delete effect) of the action, respectively. We say that there exists an action a can be executed state s iff $pre(a) \subseteq s$. Formally, our multi-agent planning formalism is heavily based on MA-STRIPS model which were first introduced by [6], and the notations are as follows.

Definition 1 An MA-STRIPS problem $\prod$ for a set of all multiple planning agents $\Phi = \{\varphi_i\}_{i=1}^{k}$ is given by a quadruple $\Pi = < F, \{A_i\}_{i=1}^{k}, I, G >$, where

- F is a finite set of fluents(or propositions), $I \subseteq F$ is the initial state of the world, and $G \subseteq F$ is the specification of the goal conditions.

We gratefully acknowledge funding from the Scientific and Technological Research Projects of Universities in Guangxi(2013YB240).

- For $1 \le i \le k, A_i$ is the finite set of actions available to agent φ_i. Each action $a \in A = \cup A_i$ is given by the notation above that has the well known STRIPS syntax and semantics.

Before describing our characterization for multi-agent planning problem, we need to introduce a brief description of related definition such that the notions of internal and public actions of agents. The dependencies between local planning for agents in a MA-STRIPS problem induce several key characteristics. Firstly, on the basis of the above definition for multi-agent system model, we can use the set of fluents $F_i = \cup_{a \in A}(pre(a) \cup \text{eff(a)})$ to denote which part of fluents relate to agent φ_i . Then, the above mentioned fluents $F_i^{\text{int}} = F_i \setminus \cup_{\varphi_j \in \Phi \setminus \{\varphi_i\}} F_j$ can be further partitioned into the subsets $F_i^{pub} = F_i \setminus F_i^{\text{int}}$ and which represent its internal and public fluents, respectively. It induces directly from this notion of agent's internal fluents that agent φ_i's actions A_i can be partitioned into A_i^{int} and A_i^{pub} as its internal and public actions.

In general, the individual planning for a multi-agent planning problem can be viewed as a sequence of actions in the local partial plans. In what follows, we formalize this idea by defining the relationship between the local partial plans in the single agents.

Definition 2 A plan for multi-agent planning problem $\prod$ is a triple $P =< \overrightarrow{A}, O, \text{C} >$ where

- $\vec{a} =< a_1, \cdots, a_n >$ is a sequence of actions occurring in the plan of multi-agent system where $\vec{a} \in \overrightarrow{A}$ is the set of an ordered sequence of actions that transforms the initial state to the goal conditions .
- O is a set of ordering constraints between the actions in which denoted as a partial order(i.e., $a_i \in A, a_j \in A, (a_i \prec a_j) \in O$).
- C is a set of causal links which represent action a_i supplies the precondition pre required by action a_j that it should not be destroy between a_i and a_j.
- A local partial plan $P_i =< a_1, \cdots, a_k >\in P$ for multi-agent planning problem is a sequence of joint actions that can be can be executed by agent φ_i , and moreover, In order to facilitate solution, we limit the number of public fluents F_i^{pub} for P_i no more than two.

Using this characteristic of agents' local partial plans, we can further define the partition $G = G_1 \cup \cdots \cup G_k$ of the goal conditions into agents' individual goal conditions, respectively, where agent φ_i's local partial plan $P_i \in P$ can achieve its individual goal condition $g_i \in G_i$.

Definition 3 Given a local partial plan P_i for agent φ_i, there could exist a prefix action and a suffix action, respectively, where

- $prefix(\text{P}_i, \text{C}) = \{\text{a}_i \mid a_i, a_j \in A_i^{pub} \wedge \nexists(a_j \prec a_i) \in C\}$ is the public action in A_i^{pub} that the ordered execution of all other public actions in A_i^{pub} must be executed following it.
- Similarly to prefix of the local partial plan P_i , $suffix(\text{P}_i, \text{C}) = \{\text{a}_i \mid a_i, a_j \in A_i \wedge \nexists(a_j \prec a_i) \in C\}$ denote that the action in A_i^{pub} should not be executed preceding the ordered execution of all other public actions in A_i^{pub}.

B. *The Structure of multi-agent Planning Graph*

We now introduce the notation and definitions of a partial order graph for multi-agent planning that play a key role in our algorithm.

Definition 4 A directed layered graph for multi-agent planning problem $\prod$ is a tuple $G =< V, E >$, in which $V = \{\text{v}_i\}$ is non-empty finite set of vertices(also called nodes) in the graph and each vertices corresponds to the action $a_i \in A$ that agent $\varphi_i \in \Phi$ plan to perform. $E \subseteq V \times V$ is a finite set of directed arcs or edges, an arc or edge $e = [v_i, v_j]$ indicates directed edge from v_i to v_j which is considered to be a causal link between them.

Notice that a sequence of actions $a_i, \cdots, \text{a}_k$ can be represented by a local partial plan P_i , therefore, we formalize our view of planning steps as a directed layered graph. Furthermore, we ignore the internal actions of agents that their pre-conditions can always be satisfied by individual agent's local partial plan, focusing instead on the relations of cooperation between agents. Adopting the notion of multi-agent planning's directed layered graph above, we refer to local partial plans as nodes, and to causal links as arcs. Thus, it is straightforward to extend such representation to a directed layered graph, and most convenient to explore natural structure for multi-agent planning problem. What is more, we will record more detail information by adding an additional layer of states. That is, at each level in the graph structure contains two alternating layers of states and actions. In order to discuss and analyse the graphical structure for multi-agent planning, we are using a block and a fragment of the graph structure to represent a local partial plan and all local partial plans that belong to a particular agent, respectively, and they are generated and kept by a single agent(i.e, Agent 1, Agent 2, and Agent 3 constructs one block, two blocks, and one block in Figure 1, respectively).

A directed layered graph structure built by our multi-agent planning algorithm is shown in Figure 2, there exist three agents that each of them participates in building a fragment in the graph structure, and agent 1, agent 2, and agent 3 generates 1 block, 2 blocks, and 1 block, respectively. Each block has a sequence of internal actions of individual agent φ_i that possibly includes no-op actions. The no-op action for every fluents(or propositions) $p \in F$ in classical planning problem is defined as $pre(a_p) = eff(a_p) = p$. Notice that the internal actions of agent φ_i in the block affect only its internal state, and the overlap between the blocks in a graph structure is coordination points for multi-agent planning problem. A graph structure is said to be incomplete if there exist preconditions of a block that are not yet supported through a causal link, and, the goal-regression process will propose local partial plans for the purpose of achieving such preconditions(also called subgoals). Thus, we introduce causal link representation for conveniently handing relation between these blocks: a causal link for blocks is of the form $\mathcal{B}_i \xrightarrow{pre} \mathcal{B}_j$ iff $\mathcal{B}_i$ and $\mathcal{B}_j$ are blocks in graph structure, such that there exists a local partial plan P_i can be performed by $\mathcal{B}_i$ which can supply the precondition pre to the block $\mathcal{B}_j$.

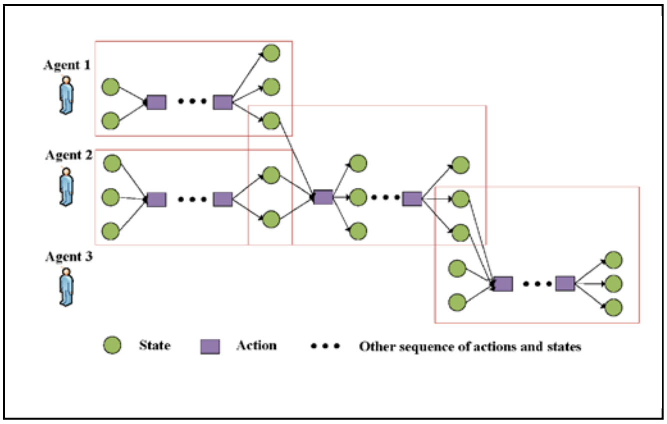

Fig. 1. A directed layered graph structure

An alternative subgoal g_i in a multi-agent planning graph structure is a precondition pre of block $\mathcal{B}_j$ that has not yet supported by a causal link $\mathcal{B}_i \xrightarrow{pre} \mathcal{B}_j$. A goal-regression process needs to generate all the alternative subgoals $subgoals(\prod)$ that supply the preconditions for the block that has already in the graph structure. we refer to $subgoals(\prod)$ as the set of alternative subgoals in $\prod$. A graph structure for multi-agent planning problem is complete if no such unhandled subgoal exists. Otherwise, we say that it is a incomplete graph structure.

A valid plan P for multi-agent planning problem $\prod = < F, \{A_i\}_{i=1}^k, I, G >$ is a sequences of local partial plans(blocks) which must be locally consistent such that the internal preconditions of every action in each block are satisfied, and all global goals are achieved as well at the end of these partial plans execution, that is, $\nexists \mathcal{B}_i, \mathcal{B}_j \in P$, there exists a threat between the two blocks $\mathcal{B}_i, \mathcal{B}_j$, or mutually exclusive in the concurrent execution.

III. Multi-Agent Planning Algorithm

In this section, we present our multi-agent planning Algorithm and then briefly introduce the main process of realization.

First, we introduce the goal-regression operator, defined $R(G_i, a) = (G_i \setminus eff(a)) \cup \text{pre}(a)$, is the particular preconditions can be regressed backwards from G_i through an action a with $eff(a)$, where $pre(a)$ and $eff(a)$ indicate the precondition and the effect of an action a, respectively. Otherwise, we say $R(G_i, a)$ is undefined if the particular preconditions cannot be regressed backwards from G_i through an action a with $eff(a)$. Obviously, we can use goal-regression operator repeatedly to generate the particular situation which is closer to the initial state I than the goal condition G.

As we already mentioned above, current multi-agent planning algorithms become very inefficient and time consuming when solving tightly coupled multi-agent planning problem that there has large number of coordination points per agent. To find a plan for multi-agent planning problem, we exploit the information of relationship between coordination points rather than exhaustive searching the consistent execution time points with respect to possible coordination sequences. We summary the main procedures for our multi-agent planning algorithm that given a multi-agent planning problem obtains a set of local partial plans. The high-level skeleton of our algorithm LGS(leaning graph structure) is depicted in Algorithm 1.

Algorithm 1 An overview of our multi-agent planning algorithm framework

input: a multi-agent planning problem $\prod$ with a set of agents $\Phi = \{\varphi_i\}_{i=1}^k$

output: a valid plan P that is a solution to $\prod$ or failure

1. initial multi-agent planning problem $\prod$
2. WHILE $\exists g_i \in subgoals(\Pi)$ do
3. learning graph structure
4. check and resolve threats or mutually exclusive relationships
5. agent φ_i communicates with the other agents
6. if achieve all global goals G or no solution
7. break
8. $P \leftarrow$ use graph structure and topological order technique
9. Coordination plan according to Graphical structure and knowledge
10. Return P

In order to use goal-regression, it is typical and convenient to add a dummy agent which has only one action producing completion status of solution process, and moreover the preconditions of this action are the global goals. To start with, given the multi-agent planning problem of multiple agents, the algorithm performs the initialization process including creates the set of public and internal fluents according to above mentioned conception so as to be further used in the following planning process. Furthermore, agents start to carry out the local planning process by iterative deepening goal-regression with exchange of partial local information until solving the overall planning problem, and keep moving the graph structure expanding forward if the local partial plans are consistent. Finally, after this iterative planning process terminates, we use topological order technique to extract an ordered sequence of the actions from the planning graph structure, so as to obtain a valid plan in the end. Once a globally consistent solution has found for multi-agent planning task is found, planning process of multi-agent stops, and the result is return. Otherwise, there is no solution plan can be found in the multi-agent planning problem.

We consider multi-agent planning problem in which multiple agents must plan to achieve the global goals with need for coordination, and each of them has its own capabilities(actions can be executed by the agent) and its own local information that cannot be used by someone else. After the set of all possible fluents are created, each agent keeps partial information of multi-agent planning problem and shares its limited knowledge with the other agents during planning process. That is, none of these agents has the complete search information about the multi-agent planning task. The knowledge and structure for the multi-agent planning problem, especially the causal links between agents' actions, are very crucial which show greatly affect the efficiency of the multi-agent planning search.

IV. Conclusion and Future Works

When multi-agent planning problem is solved using Planning-First algorithm, it looks like it should easy to tackle by determining the order of the public actions of different agents. However, to find the consistency of coordination points and internal-planning become so difficult that often he execution sequence of agents' public actions cannot be easily determine in advance. As has become tightly coupled from the multi-agent system, two fundamental problems underlie the multi-agent planning algorithms: Firstly, the complexity of the multi-agent planning algorithm to solve multi-agent planning problem grows with the problem size, which is exponential in the degree of interaction between agents and the minimax number of public actions are executed by each agent. And secondly, finding a globally consistent plan achieving all the agents' sub-goals is often infeasible, which lead to agents' backtrace procedures(that is because the execution sequence of the public actions by different agents are often inconsistent), thus causing multi-agent planning algorithm to be inefficient. Therefore, it is not surprising that PF algorithm for multi-agent become exponentially harder than centralized planning in the tightly coupled multi-agent system. In this paper, we present a novel approach to improve the efficiency of solving multi-agent planning problems in MA-STRIPS domains with a fully distributed setting. This algorithm can be easily deal with various types of multi-agent planning problem including those with very complex interactions between agents. Coordination points are defined through MA-STRIPS model, which is very useful to build distributed constraints in the multi-agent planning process. These constraints can be adapted to ensure that the local partial plans work together into a consistent global multi-agent plan. On the other hand, all the interrelated agents interact with other agents by exchange information which needs to be shared during the search process.

Our approach is based on the goal-regression procedure that is concurrently being integrated with building multi-agent planning graph structure iteratively and use agents' local planers to formulate local partial solutions until a consistent global plan is found. More importantly, we combine and adapt traditional planning technologies into multi-agent planning algorithm. That is, we use relevant partial-order causal link methodology to construct constraints between different agents, and exploit reachability analysis to check and resolve implicit inconsistencies as early as possibly. We also suggested that exploiting the graph structure of multi-agent planning problem might be appropriate for agents' communication and coordination with each other. Finally, we further use topological order technique through this graph structure model to determine actions' execution timepoints. We theoretical analytically show that our algorithm is superior to one of the state-of-the-art multi-agent planning algorithms, neither in terms of performance nor the number of message passing.

Currently, we assume that multiple agents are cooperative in the deterministic planning domains. In the future, we would like to extend our approach to non-cooperative and noderministic scenario.

References

[1] C. Boutilier, Planning, learning and coordination in multiagent decision processes, in: In Proceedings of the Sixth Conference on Theoretical Aspects of Rationality and Knowledge (TARK96), 1996,pp.195–210.

[2] M. de Weerdt, B. Clement, Introduction to planning in multiagent systems, Multiagent Grid Syst. 5 (4) (2009) pp.345–355.

[3] T. D. Bui, W. Jamroga, Multi-Agent Planning with Planning Graph, in: EUNITE 2003, pp.558–564.

[4] Y. Dimopoulos, P. Moraitis, Multi-Agent Coordination and Cooperation through Classical Planning, in: IAT, pp.398–402, 2006.

[5] J. F. Zhang, X. T. Nguyen, R. Kowalczyk, Graph-based multiagent replanning algorithm, in: AAMAS, 122, 2007.

[6] R. Brafman, C. Domshlak, From One to Many: Planning for Loosely Coupled Multi-Agent Systems, in: The International Conference on Automated Planning and Scheduling, American Association for Artificial Intelligence, Austin, Texas, 28–35, 2008.

[7] R. Nissim, R. I. Brafman, C. Domshlak, A General, Fully Distributed Multi-Agent Planning Algorithm, in: 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), International Foundation for Autonomous Agents and Multiagent Systems,2010