Multi-agent Planning Based on Causal Graphs: From Theory to Experiment

Lin Zeng, *Member, IAENG,* GuangZhi Chen, BiWan Yang

***Abstract*—Multi-agent planning is a challenging but largely understudied planning problem. In the cooperative assumption, past techniques either used a backtrack-based solution to build a globally consistent plan or created local plans first and subsequently merged them. Local planning amongst agents, on the other hand, is in most cases incompatible and contradictory, and developing a globally consistent plan in this scenario is time-consuming and laborious. Since exact planning is intractable with the agents' local planning, we present an approximate planning algorithm that initially ignores the internal nodes and considers solving only the external nodes. Theoretical analysis and experimental results show that compared to a state-of-the-art multi-agent planning algorithm, our approach can efficiently address the multi-agent planning problems with tight coupling.**

***Index Terms*—Multi-agent System, Intelligence Planning, Causal-Links, Graph Structure**

I. INTRODUCTION

MULTI-AGENT-RELATED problems are both interesting and meaningful in artificial intelligence, which has recently drawn increasing attention of both domestic and foreign researchers. Multi-agent-related technology is widely used a variety of applications in many industries including disaster relief scenarios[1], mobile robot autonomous exploration[2], public safety[3], unmanned surface vessels[4], cloud computing[5], social network[6] and automatic web service composition[7]. Moreover, there are many practical application scenarios for the cooperation of agents to complete tasks. Typically, these instances have certain characteristics; each agent, for instance, has a unique capacity to conduct and develop individual's plans while maintaining separation. To fulfill the global goals, the agents must plan individually and communicate with other agents. As a result, techniques for efficiently achieving global objective for multi-agent planning problems must be developed.

Many strategies have been developed to address multi-agent planning problems. Jamroga, for instance treated the various links between the agents' local planning using a planning graph[8]. Cheng proposed a Teaching-Learning-Interactive Learning-Based Optimization (TLILBO) to find a collision-free planning path of mobile robots[9]. Several research efforts have focused on developing the multi-agent plan coordination technique[10][11]. In the planning of many agents, Dimopoulos and Moraitis handled coordination and collaboration[12]. Other research looked at how to employ partial order planning in multi-agent planning[13][14]. To handle multi-agent replanning problems, Zhang et al. integrated graph planning with distributed constraint satisfaction approaches[15], or employed causal-links graphs for theoretical research only[16]. Brafman and Domshlak presented a revolutionary multi-agent planning concept called CSP+Planning[17], which was eventually evolved to a fully distributed multi-agent planning system called Planning-First[18]. Although the great progress achieved by multi-agent planning research efforts to date, most of the existing work has focused on separate individual planning and collaboration, and did not consider the causal links between these sub-goals and goals. In these studies, the consistency of local planning was determined through continual exchange of messages and backtrack-based solution. As a result, these efforts either implement a pseudo-distributed multi-agent planning or share information resources via memory.

In this work, we describe a unique strategy to address multi-agent planning problems that is based on an iterative backward search. Unlike existing multi-agent planning technologies, our technique can efficiently address the closely linked multi-agent planning problem while saving limited computing and communication resources. Our core approach is to first establish the causal-links for the multi-agent planning problem, and then enable agents to iteratively explore their realistic objectives throughout the backward planning search. We also determined the sequence of actions that can be performed consistently between agents and use the structure graph to achieve efficient planning. Furthermore, We enable agents to perform in a fully distributed environment in which other agents do not have access to their personal information. The main achievements, including contributions in this work are summarised as follows. First, we present a novel multi-agent planning algorithm based on the iterative learning graph structure through using a goal-oriented procedure. Second, we demonstrate that this technique can handle multi-agent planning problems with tight coupling effectively. Finally, our approach of exploring the inner structures and the causal links between these structures may also be beneficial for inspiring new concepts and methods for multi-agent community.

II. PROBLEM STATEMENT

The cooperative multi-agent model is built on the STRIPS-style language formalism, and has been used successfully in the problem of fully observable external environment and deterministic planning domains. The definitions and syntax in multi-agent planning problems that we will use in the rest of this work are briefly introduced in the following sections.

Manuscript received June 2, 2020; revised November 28, 2021. This work was funded by the Doctoral Scientific Research Foundation of Hezhou University (Grant No. HZUBS201709) and 2021 Guangxi Higher Education Teaching Reform Project(Grant No. 2021JGA336).

Lin Zeng is a senior engineer in the School of Artificial Intelligence, HeZhou University, HeZhou, 542899, China (e-mail: z-suzenglin@gmail.com).

GuangZhi Chen is a lecturer in the School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou, 450015, China (e-mail: chgzhi@zua.edu.cn).

BiWan Yang is a professor in the School of Culture and Media, HeZhou University, HeZhou, 542899, China (e-mail: yangbw80@163.com).

We explore planning for multi-agent system, in which agents must coordinate their actions in order to satisfy a set of global objectives. The multi-STRIPS formalism is the emphasis of this paper, although it may easily be extended to the STRIPS representation of the classical planning language. It was obvious that a multi-agent planning problem is made up of several(≥ 2) agents, with all agents collaborating to accomplish the global objective. Multi-agent planning problems, by definition, involve more than one autonomous agent, each of whom must design and coordinate their own local plans in order to achieve global objective within the constraints imposed by their starting state and actions. Moreover, each agent may only plan its own activities and for its immediate surroundings.

In the MA-STRIPS framework, the states information of the surrounding environment are defined by a limited subset of propositions and action is described as tuples $a = \langle eff(a), pre(a) \rangle$ where $eff(a)$ and $pre(a)$ express the effect(add and delete effect) and the precondition of the action, respectively. If $pre(a) \subseteq s$, then there exists an action a that can be performed given state s. Theoretically, The representation of our multi-agent programming algorithm is substantially based on Souliman and Shleyfman's MA-STRIPS framework, which was initially described by [13], and employs the following concepts and notation.

Definition 1 A quadruple $\Gamma = \langle \mathcal{F}, \{\mathcal{A}_i\}_{i=1}^{l}, \mathcal{I}, \mathcal{G} \rangle$ is used to represent an MA-STRIPS problem Γ with a set of planning agents $\{\varphi_i\}_{i=1}^{l}$.

- $\mathcal{F}$ is a limited subset of propositions, $\mathcal{I} \subseteq \mathcal{F}$ is the initial information of the surrounding environment, and $\mathcal{G} \subseteq \mathcal{F}$ is the specified goal situations,
- For $1 \leq i \leq l, \mathcal{A}_i$ is the limited subset of actions available for agent φ_i to perform. Each action $a \in \mathcal{A} = \cup \mathcal{A}_i$ is is represented by the symbol above, which follows the STRIPS semantics and syntax[19].

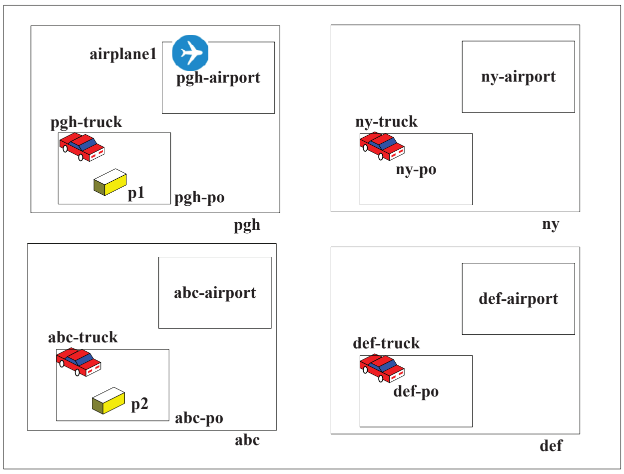

Fig. 1. An example of the multi-agent planning problem

When $l = 1$ and the individual agent holds private knowledge that no one else knows, the MA-STRIPS model becomes completely equal to the STRIPS framework. Prior to explaining our characterisation for the multi-agent planning issue, a quick introduction of relevant definitions, such as the basic concept of public and internal actions of agents, is required. In an MA-STRIPS scenario, the interdependence between local planning for agents provide numerous significant properties. To begin, we may use the set of propositions $\mathcal{F}_i = \cup_{a \in \mathcal{A}_i} pre(a) \cup eff(a)$ to designate which portion of propositions pertain to agent φ_i, assuming we already have the multi-agent planning model formulation. The above-mentioned propositions may then be subdivided into he subsets $\mathcal{F}_i^{\otimes} = \mathcal{F}_i \setminus \mathcal{F}_i^{\odot}$ and $\mathcal{F}_i^{\odot} = \mathcal{F}_i \setminus \cup_{\varphi_j \in \Phi \setminus \{\varphi_i\}} \mathcal{F}_j$ that represent its public propositions and internal propositions, respectively. Actions $\mathcal{A}_i$ of agent φ_i can be partitioned into $\mathcal{A}_i^{\otimes}$ and $\mathcal{A}_i^{\odot}$ as its public and internal actions respectively, as a direct result of the concept of agent's internal propositions.

We utilize employ a scenario from the well-known logistics industry as a sample example of a multi-agent planning problem to better comprehend the MA-STRIPS framework. The goal is to deliver a collection of goods from their starting points to their final destinations. The vehicles (trucks or airplanes) can transport a package from one area on the map to another. Each vehicle is limited to moving along a subset of the roadmap segments within or between cities. As illustrated in Fig.1, we assume that there are initially five agents in the roadmap, with four trucks(agents) capable of transporting goods across a roadmap in each city and one airplane(agent) capable of transporting items between two cities. The agents' final overall goals are to transport the goods p1 from the beginning location pgh-po to the target position def-po and the goods p2 from abc-po to ny-po. Each truck may do three distinct actions: load and unload the goods in the carriage, as well as travel through the city's roadmap. In this MA-STRIPS planning example, there are many public actions (e.g., load(p1, ny-truck, ny-airport), unload(p2, ny-truck, ny-airport), load(p1, def-truck, def-airport) and unload(p2, def-truck, def-airport)) that serve as coordination points. There are other actions that are only available to an agent, which including move(ny-truck, ny-po, ny-ariport) or fly(airplane1, abc-airport, def-ariport). It is worth noticing that the agents' public actions depend on the specific multi-agent planning problem. The Logistics planning problem poses significant difficulty for the multi-agent algorithm. It is tightly coupled and on average, more than half of the actions of each agent are public actions. In a multi-agent planning system, "tightly linked" indicates that the activities of the agents frequently need to interact with one another.

III. THE PROPOSED METHOD

As previously stated, we categorize the propositions and actions of the multi-agent planning issue into two types: private and public. The crucial milestones in the collaboration amongst the agents to attain the global objective are coordination points, which are public propositions or actions. In general, the state of propositions and the actions in the planning graph correlate one to one. As a result, In the strategy of solving multi-agent planning problems, we can obtain state propositions based on actions (or vice versa). We employ the actions in the causal-links graph to help with representation and analysis. The three agents' local plans include eleven internal points $\beta_1, \beta_2, \cdots, \beta_{11}$ and five coordination points $\alpha_1, \alpha_2, \cdots, \alpha_5$, as illustrated in Fig.2. These coordination points are in the order of $< \alpha_1, \alpha_2, \cdots, \alpha_5 >$ (for example, execute α_1 before α_2). All of the sub-plans comprise a solution $< \gamma_1, \gamma_2, \cdots, \gamma_n >$ for the multi-agent planning problem. A monotonic mapping function f exists that maps the action α_i in the public coordination points into $< \gamma_1, \gamma_2, \cdots, \gamma_n >$, such that $r_{f(i)} = a_i$.

A major challenge is determining the coordination points and their execution sequence. Furthermore, These elements are also beneficial in resolving the agents' local plans. In this case, we propose using a "goal-oriented" method for constructing the planning causal graph, which is essential for resolving the problem of agent collaboration in the multi-agent planning problem. To put it another way, the "goal-oriented" technique is to devise a strategy for achieving the specified goal state, and carrying out that strategy requires the relevant conditions.

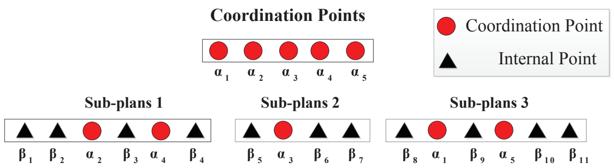

Fig. 2. The coordination points of actions

Prior to introducing the "goal-oriented" technique, it is first necessary to describe how the causal-links formal representation is used in this paper. We then use some of its useful derivations in the subsequent solution to the multi-agent planning problem. The simplest form of Causal-links is $sub\text{-}plan_1 \rightarrow sub\text{-}goal_1 \rightarrow goal_1 \rightarrow sub\text{-}plan_2 \rightarrow goal$. This causal link demonstrates that executing $sub\text{-}plan_1$ is utilized to reach $sub\text{-}goal_1$, the objective of which is to enable $sub\text{-}plan_2$ to obtain the $goal$. A finite execution sequence $sub\text{-}plan_1 \rightarrow sub\text{-}goal_1 \rightarrow goal_1 \rightarrow \cdots \rightarrow sub\text{-}goal_m \rightarrow sub\text{-}plan_2 \rightarrow sub\text{-}goal_2 \rightarrow goal_2 \rightarrow \cdots \rightarrow goal_n$ is the most prevalent kind of causal-links, where the sub-link $sub\text{-}goal_1 \rightarrow sub\text{-}goal_2$ indicates that $sub\text{-}goal_1$ is used for obtaining $sub\text{-}goal_2$ without the requirement for performing other public actions. Moreover, if $goal_2$ is conjunctive, then $sub\text{-}goal_1$ or $sub\text{-}goal_2$ can be one of its conjuncts.

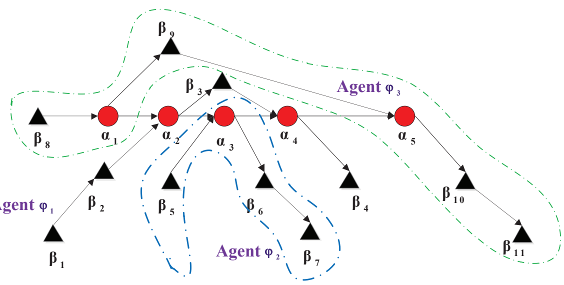

Fig. 3. The simplified causal-links graph

To use causal-links to build causal graph, we must consider the causal-links $sub\text{-}plan_1 \rightarrow sub\text{-}goal_1 \rightarrow goal_1 \rightarrow \cdots \rightarrow sub\text{-}goal_m \rightarrow sub\text{-}plan_2 \rightarrow sub\text{-}goal_2 \rightarrow goal_2 \rightarrow \cdots \rightarrow goal_n$ in the two special cases. The first case is that $sub\text{-}goal$ is true without the need to carry out a further local plan. Then, we simply add a dummy operation "$start^*$", that takes precedence to all other items in the causal-links. Therefore, we can obtain the form of causal links as $start^* \rightarrow sub\text{-}plan_1 \rightarrow sub\text{-}goal_1 \rightarrow goal_1 \rightarrow \cdots \rightarrow sub\text{-}goal_m \rightarrow sub\text{-}plan_2 \rightarrow sub\text{-}goal_2 \rightarrow goal_2 \rightarrow \cdots \rightarrow goal_n$. The second case is that $goal_n$ is the ultimate objective of this causal-links. Again, we add another dummy operation "$finish^*$", and the causal links changes into the following form $start^* \rightarrow sub\text{-}plan_1 \rightarrow sub\text{-}goal_1 \rightarrow goal_1 \rightarrow \cdots \rightarrow sub\text{-}goal_m \rightarrow sub\text{-}plan_2 \rightarrow sub\text{-}goal_2 \rightarrow goal_2 \rightarrow \cdots \rightarrow goal_n \rightarrow finish^* \rightarrow goal_n$. After the procedures above, we can easily derive and implement the related theories based on the causal-links. There are several causal-links with different sub-goals that are used to achieve the global goal for multi-agent planning problem. If these causal-links do not give rise to destructive interference with each other, we can merge them into a causal-links graph. We can simply understand "destructive interference" as the destruction of the preconditions of other agents' actions, and we will formally state this definition below. Fig. 3 is a simplified causal-links graph with three agents' sub-plans, where the blue and green dotted lines represent the causal-links of Agents 2 and 3, respectively. The key point is to determine the order of execution of the five coordination points that are marked by red circles. Since the internal nodes can be solved by the Agent itself, here we adopt the strategy of ignoring internal nodes and only considering the solution for external nodes.

Destructive interference: Given a multi-agent planning solution $< \alpha_1, \alpha_2, \alpha_3, \alpha_4, \cdots, \alpha_n >$, if only α_1 provides α_4 with the precondition p, and some β_i have an effect $p(\sim p \in add(\beta_i))$ during their execution.

Consider this extreme multi-agent planning problem; its goal conditions are already true in the initial state, and we can obtain the solution of the multi-agent planning problem without any planning steps. This empty planning solution is what we call the null-plan, and its causal-links has the form of $start^* \rightarrow goal \rightarrow finish^* \rightarrow goal$. Our "goal-oriented" technique employs null-plan to verify whether the objective has been reached when dealing with multi-agent planning. Then there's Generate-null-plan, which has the following statements.

Generate-null-plan: During solving process of the "goal-oriented" method, if exploring for the objective that is already reached, then create a null-plan for obtaining the objective.

The "goal-oriented" technique is given a formal formulation in this paper. To attain a particular objective $\mathcal{G}$, we must execute local planning $\mathcal{P}_L$ under the constraint $\mathcal{C}$. The "goal-oriented" technique is denoted by $\mathcal{P}_L/\mathcal{C} \blacktriangleright \mathcal{G}$. Supposing $\mathcal{P}_C$ is a sub-plan for achieving constraint $\mathcal{C}$, we build the new execution sequence $\mathcal{P}_N$ as follows:

- Inserting a new execution sequence $\mathcal{P}_L$ behind the sub-plan $\mathcal{P}_C$,
- Sorting all of the actions that occur in the local planning $\mathcal{P}_L$ after the sub-plan $\mathcal{P}_C$, and
- Modifying the causal-links accordingly.

When a "goal-oriented" approach is used in multi-agent planning, the goal $\mathcal{G}$ commonly has the conjunctive form in problem-solving techniques. Then, we may split G into multiple a conjunctive form composed of multiple propositions, namely $\mathcal{G} = \mathcal{G}_1 \wedge \mathcal{G}_2 \wedge \cdots \wedge \mathcal{G}_n$. Each proposition denotes a sub-goal $\mathcal{G}_i$ to be satisfied. Moreover, consider the most basic conjunctive form $\mathcal{G} = \mathcal{G}_1 \wedge \mathcal{G}_2$, then we may further split the form $\mathcal{P}_L/\mathcal{C} \blacktriangleright \mathcal{G}$ into $\mathcal{P}_{L_1}/\mathcal{C}_1 \blacktriangleright \mathcal{G}_1$ and $\mathcal{P}_{L_2}/\mathcal{C}_2 \blacktriangleright \mathcal{G}_2$. Suppose $\mathcal{P}_1$ and $\mathcal{P}_2$ is planning to achieve $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, their corresponding causal-links are $sub\text{-}plan_1 \rightarrow sub\text{-}goal_1 \rightarrow \cdots \rightarrow \mathcal{G}_1 \rightarrow finish^* \rightarrow \mathcal{G}_1$ and $sub\text{-}plan_2 \rightarrow sub\text{-}goal_2 \rightarrow \cdots \rightarrow \mathcal{G}_2 \rightarrow finish^* \rightarrow \mathcal{G}_2$ respectively. $\mathcal{P}_1 + \mathcal{P}_2$ is their joint planning for achieving the conjunctive goal $\mathcal{G}$. Then, we apply the necessary adjustments to the form of their causal-links that becomes the following form $sub\text{-}plan_1 \rightarrow sub\text{-}goal_1 \rightarrow \cdots \rightarrow \mathcal{G}_1 \rightarrow \mathcal{G}_1 \wedge \mathcal{G}_2 \rightarrow finish^* \rightarrow \mathcal{G}_1 \wedge \mathcal{G}_2$, $sub\text{-}plan_2 \rightarrow sub\text{-}goal_2 \rightarrow \cdots \rightarrow \mathcal{G}_2 \rightarrow \mathcal{G}_1 \wedge \mathcal{G}_2 \rightarrow finish^* \rightarrow \mathcal{G}_1 \wedge \mathcal{G}_2$. $\mathcal{P}_1 + \mathcal{P}_2$ can attain the objective $\mathcal{G}$ if there is no detrimental interference between the synthetic planning. We get the following statement of "Solve-conjunctive-goal" in more detail.

Solve-conjunctive-goal: For a given conjunctive objective $\mathcal{G} = \mathcal{G}_1 \wedge \mathcal{G}_2$, if there are plans $\mathcal{P}_1$ and $\mathcal{P}_2$ to reach the sub-goal $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively, and these plans don't interfere with each other. Then, we present the plan $\mathcal{P}_1 + \mathcal{P}_2$ for achieving the objective $\mathcal{G}$.

It is vital to make each local planning solution consistent while handling global planning problems. As a result, we establish the following limitations.

(R1) Causal-links Restriction

A complete causal-links composed of n coordination points $< \alpha_1, \alpha_2, \cdots, \alpha_n >$ satisfies (R1) iff, for $1 \leq i \leq n$, $\mathcal{P}_{L_i}/\mathcal{C}_i \blacktriangleright \mathcal{G}_i$, and $\alpha_i \in \mathcal{P}_{L_i}$, the following statements are correct:

a) $i = 1$, and agent φ_i who performs $\mathcal{P}_{L_i}$ can provide the condition $\mathcal{C}_i$;

b) $1 < i < n-1$, and $\mathcal{P}_{L_i}$ can achieve $\mathcal{G}_i$ that is either $\mathcal{G}_n$ or a conjunction of $\mathcal{G}_n$, and for $i < j \leq n$, $\mathcal{P}_{L_j}/\mathcal{C}_j \blacktriangleright\!\sim \mathcal{G}_i$ does not hold;

c) $i = n$, and $\mathcal{P}_{L_n}$ can achieve $\mathcal{G}_n$, and the sub-plans $\{\mathcal{P}_{L_j}\}(1 < j < n-1)$ which produce a series of actions $< \alpha_1, \alpha_2, \cdots, \alpha_{n-1} >$ can supply $\mathcal{C}_n$;

(R2) Local-planning Restriction

The agents' sub-plans $\mathcal{P}_i, (1 \leq i \leq l)$ satisfy (R2) iff, for each $\mathcal{P}_i =< \alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{ik} >$, The local planning problem with action landmarks $< \mathcal{F}_i, A_i^{\odot}, I \cap F_i, \emptyset, < \alpha_{i1}, \alpha_{i2}, \cdots, \alpha_{ik} >>$ can be solved.

The causal-links restriction (R1) assures that the local plans are consistent and do not conflict with one another. This external limitation allows for conflict-free local planning among the various agents. The Local-planning Restriction (R2) assures that each agent is capable of carrying out the coordination points and meeting their preconditions. In other words, the agent φ_i develops local plans to create specific internal preconditions for its public actions.

IV. FRAMEWORK AND THEORETICAL ANALYSIS

In this section, we first introduce our GCL_Γ algorithm framework, and then analyze its specific implementation process in detail. Furthermore, we analyze GCL_Γ theoretically.

A. Framework of our algorithm

An overview of our GCL_Γ algorithm is shown in Algorithm 1. To use the goal-oriented approach, it is easy to introduce a dummy agent φ_d with only one action that produces the completed state. First, we will initialize multi-agent planning problem Γ, including the previously described actions that include both public and private types. Agent φ_d produces the global goal $\mathcal{G}$ to insert an open queue that records whether to achieve the goals or sub-goals. In Step 7, we use the "goal-oriented" method presented above to solve the subgoal $\mathcal{G}_i$. Further, we also combine the relaxed plan heuristics[20] and action landmark information in the local planning[21]. For the information that has been solving local planning, we will record the minimum cost path. In Step 8, different sub-goals may cause destructive interferences between internal or external, and then we try to adjust to the local plan, if not for an external adjustment. Furthermore, we introduce appropriate ordering-constraints to repair destructive interferences. In Step 9, agent φ_i uses the broadcast communication method to obtain the necessary information. In Step 13, once a globally consistent solution for Γ is found, planning is terminated, and the plan $\mathcal{P}$ returns. Otherwise, there is no solution plan.

Algorithm 1 Framework of our GCL_Γ algorithm

```
1:  Input F, I, G, and {φ_i}_{i=1}^l
2:  Initial the multi-agent planning problem Γ;
3:  While ∃G_i ∈ subgoals(G) do;
4:      If G_i is the conjunctive goal then;
5:          Solve-conjunctive-goal(G_i);
6:      End if;
7:      Use the "goal-oriented" method to solve G_i;
8:      Check and resolve destructive interferences;
9:      Agent φ_i communicates with the others {φ_j}(i ≠ j);
10:     If G_i is true then;
11:         Generate-null-plan(G_i);
12:     End if;
13:     If achieving G or no solution then;
14:         Break;
15:     End if;
16: End while;
17: Return plan P.
```

B. Theoretical analysis

We verify the accuracy characteristics of our GCL_Γ algorithm theoretically below.

Theorem 1 (Soundness) Given a multi-agent planning problem $\Gamma = \langle \mathcal{F}, \{\mathcal{A}_i\}_{i=1}^l, \mathcal{I}, \mathcal{G} \rangle$, if there exists a causal-links of n coordination points $< \alpha_1, \alpha_2, \cdots, \alpha_n >$ that reaches the global goal $\mathcal{G}$, then we can extend it into a completed plan $\mathcal{P}$ for Γ.

Proof Theorem 1 is proved in a straightforward manner. (R1) Causal-links restriction can achieve the sub-goals, and solve the sub-goals whose execution requirements are provided by the agents' local planning, while these requirements are validated by (R2) Local-planning Restriction.

Theorem 2 (Completeness) Given a multi-agent planning problem $\Gamma = \langle \mathcal{F}, \{\mathcal{A}_i\}_{i=1}^l, \mathcal{I}, \mathcal{G} \rangle$ that can be solved, GCL_Γ can surely find a solution $\mathcal{P}$ for Γ.

Proof The procedures for proving theorem 2 are identical to those for proving theorem 1. The identical proof portion is no longer a tautology, and the difference is that GCL_Γ executing local planning will try to loosen the constraints first, requiring the legal plan $\mathcal{P}$ to be in.

V. EXPERIMENTAL EVALUATION

In this section, we compare our GCL_Γ algorithm to the state-of-the-art algorithm (PF) to illustrate the advantage of GCL_Γ. We evaluate these two algorithm in the three benchmark domains from the International Planning Competition (IPC), and analyze the reasons why algorithm PF cannot effectively solve the multi-agent planning problem Γ.

A. Multi-agent planning domains

We performed the experiments on three domains (Logistics, Statellite, and Rovers) developed from the transport field or NASA missions to evaluate our approach. In the previous sections, we have already described the Logistics domain in which the agents can fly or move between locations, and transport goodss to specified locations. The Statellite domain has one or more satellites that make observations by turning to the target directions, and each is equipped with instruments that can support different shooting modes. In the Rovers domain, rovers equipped with instruments can travel to different waypoints to obtain and store data and transmit the soil or rock information to a lander. We note that the planning problems from the Logistics domain are more tightly coupled than the those of the Rovers and Satellites domains because in these domains most of their actions are generally public actions. For the Rovers and Satellites domains, we design various kinds of multi-agent planning problems with different degrees of complexity. That is, we generalize the loosely coupled planning multi-agent problems with weak interactions between the agents to the tightly connected problems where not only many agents, but also many coordination points must be addressed.

Table 1. Comparison of PF and GCL_Γ

$NO.$	$\|\varphi\|$	$\|\mathcal{A}_a^\otimes\|$	PF			GCL_Γ		
			$Time$	$Cost$	Msg	$Time$	$Cost$	Msg
Logistics domain								
1	3	2	0.87	11	10	0.43	11	8
2	3	3	0.90	16	12	0.47	16	11
3	4	4	×	×	×	1.08	14	12
4	4	6	×	×	×	1.28	24	22
5	5	4	×	×	×	1.60	24	23
6	5	6	×	×	×	1.91	27	23
7	6	4	×	×	×	35.00	24	25
8	6	5	×	×	×	38.09	32	32
9	7	5	×	×	×	44.21	38	45
10	7	6	×	×	×	73.14	40	59
11	8	8	×	×	×	435.68	42	61
12	10	8	×	×	×	1590.16	49	70
13	12	10	×	×	×	2126.59	53	82
14	14	10	×	×	×	3272.16	56	95
Rovers domain								
1	3	1	0.31	35	6	0.30	35	5
2	3	2	0.40	42	6	0.33	42	6
3	3	3	×	×	×	1.69	51	8
4	4	1	0.38	46	10	0.28	45	8
5	4	2	0.55	53	10	0.44	53	9
6	4	3	×	×	×	1.81	71	12
7	5	1	1.34	54	15	0.45	54	13
8	5	2	×	×	×	0.75	77	15
9	5	3	×	×	×	1.83	118	19
10	5	4	×	×	×	3.25	151	22
11	6	4	×	×	×	3.59	170	30
12	6	5	×	×	×	4.35	183	36
13	7	6	×	×	×	5.58	196	42
14	8	6	×	×	×	10.66	322	54
15	10	8	×	×	×	16.21	356	72
16	12	10	×	×	×	20.78	384	86
Statellite domain								
1	2	1	0.14	6	3	0.14	6	3
2	2	2	0.19	7	3	0.18	7	3
3	3	2	0.22	12	9	0.19	12	7
4	4	1	0.33	13	10	1.18	13	5
5	4	2	9.58	15	20	0.24	14	9
6	5	2	27.62	33	28	3.21	32	14
7	5	3	×	×	×	3.50	43	17
8	5	4	×	×	×	3.71	57	21
9	6	4	×	×	×	4.75	61	23
10	6	5	×	×	×	5.89	66	26
11	7	5	×	×	×	8.10	75	31
12	7	6	×	×	×	9.12	79	50
13	8	6	×	×	×	10.22	96	65
14	8	7	×	×	×	11.65	148	102
15	9	8	×	×	×	25.29	185	126
16	10	8	×	×	×	46.66	231	143
17	12	10	×	×	×	456.92	311	156
18	14	12	×	×	×	1073.60	392	212

B. *Results and discussion*

We also report the performance of the baseline algorithm PF to evaluate the effectiveness of our algorithm. All of our experiments were run on an Intel Core 2 Dua 2.0 GHz processor with the memory usage limited to 1024 MB. We also set the cutoff time to 3600 seconds to prevent the algorithm from running indefinitely. The reported times are given in seconds, and does not include the time of initialization. To simulate the agents' planning, we employ asynchronous threads that may interact and work with one another by sending and receiving messages. As is common in the multi-agent planning context, we assume that each agent knows certain individual facts of the initial state and the global goal conditions. For each multi-agent planning domain, we produce more than a dozen different planning problems in each planning domain by increasing the number of agents and public actions. We also change the scene to guarantee that the planning problems have a certain degree of coupling.

Table 1 shows the results of the two multi-agent planning algorithms that provide a clear picture of the performance obtained by the PF and GCL_Γ algorithms. $|\varphi|$, $|\mathcal{A}_a^\otimes|$, $Time$, $Cost$, and Msg denote the number of agents, the number of public actions, the planning cost and the total number of messages passed, respectively. $\times$ represents that the PF algorithm can solve the problems within a limited time. In the Logistics domain, the execution times vary within a certain range when given the same scale of planning problems. The Rovers and Satellites domains' results illustrate that the run-time complexity of multi-agent planning methods varies with the scale of the multi-agent planning problems.

An examination of the data presented in Table 1 reveals that the running time increases as the scale of the planning problem grows higher. If the agent needs other agents to determine the possible solution repeatedly, the PF algorithm will become exponentially harder. Moreover, when the number of agents and coordination points increase to a certain level, the PF algorithm cannot return a solution within a time limit. By contrast, the planning time of our GCL_Γ algorithm grows only linearly with the increase in the complexity of the problem. Table 1 clearly shows that the performance of the LGS algorithm perform is much better than that of PF. This justifies our initial intuitive understanding. We also evaluate the network load by counting the total number of the messages passed for the multi-agent planning problems. It is clearly obsessed that total number of messages required for our algorithm increases slightly with the complexity of the problem. To summarize, these experimental results and analysis show that our algorithm significantly outperformed PF.

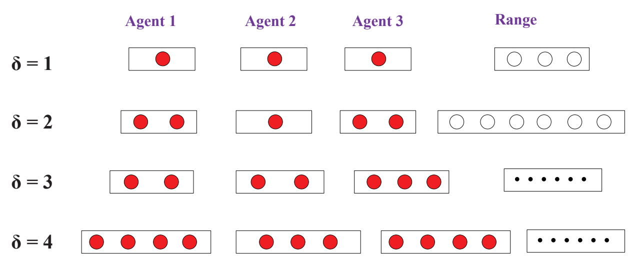

Fig. 4. PF increments the length of δ in each iteration

Here we analyze the reasons why the PF algorithm cannot effectively solve the multi-agent planning problem. When solving the multi-agent planning problem, the PF algorithm does not know the value of δ (δ indicates the number of public actions needed) necessary to reach the global goals for each agent. As a result, PF assumes that all agents have the same δ, and incremenst the length of δ in each iteration until it finds the solution. As shown in Fig. 4, at each iteration, each agent φ_i will determine an abstract time point $t(t \in Range(1, \delta * l)$ to perform $\alpha(\alpha \in \mathcal{A}_i^{\otimes})$. Only the combination of these time points can reach $\delta^{(\delta * l)}$ orders of magnitude! When the multi-agent planning problem becomes closely linked, it is challenging to achieve consistency in coordination points and internal planning. Furthermore, the execution time of the agents' public actions cannot be easily predicted in advance. Therefore, it is not surprising that PF is inefficient, and even cannot solve the problems in the limited time.

VI. Conclusion

In this research, we describe a novel method for handling multi-agent planning problems in the MA-STRIPS model using a completely distributed circumstances that is more efficient. Our GCL_Γ algorithm is capable of dealing with a wide range of multi-agent planning problems, including those involving complicated agent interactions.

Our GCL_Γ algorithm uses the goal-oriented procedures that iteratively produce the coordination points. These points are useful to build the causal-links constraints in the multi-agent planning. The constraints ensure that the local partial plans are coherent and work together to generate consistent executable actions. In addition, all of the relevant agents interact with other agents by exchanging local planning information required by the other agents. We incorporate and adapt the newest planning technologies into a multi-agent planning algorithm, which is more significant. To put it another way, we employ a relevant partial-order causal link approach to create constraints between distinct agents, and we apply reachability analysis to detect and fix implicit contradictions as soon as possible. We also suggested that the cause graph can be used to facilitate performance improvement and cost saving. Finally, we utilize the cause graph to find the execution time points using the breadth-first approach. Our practical and theoretical findings reveal that our approach outperforms the state-of-the-art PF multi-agent planning algorithm in terms of both performance and planning cost. In deterministic planning domains, we currently assume that numerous agents cooperate. We'd want to expand our method in the future to include non-cooperative and nonderministic scenarios[22].

References

[1] S. Shekhar, R. I. Brafman, and G. Shani, "Improved knowledge modeling and its use for signaling in multi-agent planning with partial observability," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, 2021, pp. 11 954–11 961.

[2] A. Ma, M. Ouimet, and J. Cortés, "Exploiting bias for cooperative planning in multi-agent tree search," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1819–1826, 2020.

[3] A. I. Bhatti *et al.*, "Multi-agent planning for thermalling gliders using multi level graph-search," *arXiv preprint arXiv:2007.01334*, 2020.

[4] V. P. K. B. T. C. K. G. Valada, A. and P. Scerri, "Development of a low cost multi-robot autonomous marine surface platform," *Field and Service Robotics*, pp. 643–658, 2014.

[5] J. Bajo, F. De la Prieta, J. M. Corchado, and S. Rodríguez, "A low-level resource allocation in an agent-based cloud computing platform," *Applied Soft Computing*, vol. 48, pp. 716–728, 2016.

[6] Y. Hui and A. Dey, "A study on directed neutrosophic social networks," *IAENG International Journal of Applied Mathematics*, vol. 50, no. 2, pp. 225–232, 2020.

[7] M. Farnaghi and A. Mansourian, "Multi-agent planning for automatic geospatial web service composition in geoportals," *ISPRS international journal of geo-information*, vol. 7, no. 10, p. 404, 2018.

[8] T. Bui and W. Jamroga, "Multi-agent planning with planning graph," in *Proceedings of eunite2003*, 2003, pp. 558–564.

[9] Y.-H. Cheng, P.-J. Chao, and C.-N. Kuo, "Mobile robot path planning using a teaching-learning-interactive learning-based optimization," *I-AENG International Journal of Computer Science*, vol. 46, no. 2, pp. 199–207, 2019.

[10] J. S. Cox and E. H. Durfee, "An efficient algorithm for multiagent plan coordination," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005, pp. 828–835.

[11] B. A. d. W. M. Tonino, H. and C. Witteveen, "Plan coordination by revision in collective agent based systems," *Artificial Intelligence*, vol. 142, pp. 121–145, 2002.

[12] Y. Dimopoulos and P. Moraitis, "Multi-agent coordination and cooperation through classical planning," in *2006 IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. IEEE, 2006, pp. 398–402.

[13] M. Brenner, "Multiagent planning with partially ordered temporal plans," in *IJCAI*, vol. 3. Citeseer, 2003, pp. 1513–1514.

[14] J. Kvarnström, "Planning for loosely coupled agents using partial order forward-chaining," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 21, no. 1, 2011, pp. 138–145.

[15] J. F. Zhang, X. T. Nguyen, and R. Kowalczyk, "Graph-based multi-agent replanning algorithm," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 2007, pp. 1–8.

[16] Z. Lin and G. Chen, "Multi-agent planing based on causal graph," in *Journal of Physics: Conference Series*, vol. 1601, no. 4. IOP Publishing, 2020, p. 042019.

[17] R. I. Brafman and C. Domshlak, "From one to many: Planning for loosely coupled multi-agent systems." in *ICAPS*, vol. 8, 2008, pp. 28–35.

[18] R. Nissim, R. I. Brafman, and C. Domshlak, "A general, fully distributed multi-agent planning algorithm," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 2010, pp. 1323–1330.

[19] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning: theory and practice*. Elsevier, 2004.

[20] J. H. Domshlak, Carmel and M. Katz, "Redcblack planning: A new systematic approach to partial delete relaxation," *Artificial Intelligence*, vol. 221, pp. 73–114, 2015.

[21] A. Shleyfman, M. Katz, M. Helmert, S. Sievers, and M. Wehrle, "Heuristics and symmetries in classical planning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, 2015, pp. 3371–3377.

[22] J. Fu, V. Ng, F. B. Bastani, I.-L. Yen *et al.*, "Simple and fast strong cyclic planning for fully-observable nondeterministic planning problems," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 3, 2011, pp. 1949–1954.

Lin Zeng received his M.Sc. degree and Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China in 2010 and 2017, respectively. He has worked as a senior engineer in the School of Artificial Intelligence, HeZhou University, HeZhou, China. His research interests mainly include data mining, artificial intelligence and intelligent planning.

GuangZhi Chen received his M.Sc. degree and Ph.D. degree in computer science from Sun Yat-sen University, Guangzhou, China in 2009 and 2016, respectively. He has worked as a lecturer in the School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou, China.