Stability Oriented Task-Structure Based Multi-Agent RePlanning

(Extended Abstract)

Li Jin
Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716, USA
jin@cis.udel.edu

ABSTRACT

In this paper, we evaluate the stability of a plan solution at the task level and argue that it is important for task structure based multi-agent replanning. With a plan solution represented by a TÆMS (Task Analysis, Environment Modeling and Simulation) task structure, we define new metrics of the stability of a TÆMS solution and use it to rank and recommend plan candidates. We preliminarily evaluate our approach with a producer-consumer-transporter domain.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search; I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems

General Terms

Algorithms

Keywords

Multi-agent planning, adaptation

1. INTRODUCTION

Prior planning and coordination efforts are often used to adapt a plan to new circumstances or contexts. Previous research on multi-agent replanning has presented the value of plan stability at a coordination level [1] and at an action level [2]. In this paper, we identify a more general metric, the stability of a task structure solution, as a major consideration to evaluate a candidate adapted from an HTN (Hierarchical Task Network) plan. By the term "solution stability ", we refer to a measure of the difference between the task structure solutions of an original plan and a newly adapted plan. This "solution stability" should take into account the differences in task decomposition, actions, agent assignment, etc. We argue that minimizing the differences between the new and old plans should be considered for replanning because of the following reasons:

Cite as: Stability Oriented Task-Structure Based Multi-Agent RePlanning, (Extended Abstract), Li Jin, *Proc. of 8th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009),* Decker, Sichman, Sierra and Castelfranchi (eds.), May, 10–15, 2009, Budapest, Hungary, pp. 1209–1210

- Generally, preserving the stability of actions of a plan can reduce the cognitive load on human observers of planned activities and can allow planner to generate high quality plans by exploiting the reasoning performed in the original plan generation [2].
- Specifically, for HTN planning, it is computationally expensive to derive an effective task decomposition. Therefore, in order to avoid incurring another round of significant overhead, a planner might prefer to decompose the tasks similarly as before.
- Particularly, when more and more agents join a multi-agent system, how to assign them tasks will become increasingly important for executing a plan efficiently. Agent assignment of an old plan might have been built based on some valuable properties, such as agent reputation. Thus, we argue that changes in agent task assignment should be minimized.
- It is possible to generate a plan more quickly by seeking to preserve stability than by replanning from scratch.

2. TÆMS SOLUTION STABILITY

In this paper, we use TÆMS task structures to define the metrics of "solution stability". A coordinated TÆMS plan solution to a problem is defined as a tuple $P = \{A, T, M, t, R, D, \mathrm{SNL}, d\}$, where A is a set of agents, T is a set of TÆMS tasks, M is a set of TÆMS actions (methods), t is a mapping from tasks to sets of subtasks and/or actions, $R \in T$ is a set of goal tasks that represent the root task nodes of independent task trees, D is a mapping from root tasks in R to deadlines, and SNL is a set of labeled arcs in the task structure graph representing the relations between elements of T and/or M [4].

Definition 1. Given a TÆMS plan solution, P, and a new solution, P', we define the following criterion to measure the difference between these two solutions:

$$D(P, P') = w_a \times D(A, A') + w_t \times D(T, T') + w_m \times D(M, M') + w_r \times D(\mathrm{SNL}, \mathrm{SNL'}) \quad (1)$$

where

- $D(A, A') = \sum_i w_i \times ||A_i| - |A'_i||$, where A_i, A'_i are sets of agents respectively appearing in P and P' that have a specific role or function i, $|A_i|$ is the number of agents in A_i, $|A'_i|$ is the number of agents in A'_i, w_i is the weight for agents playing role i;

- $D(T, T')$ is the difference between the number of (task / method) nodes appearing in P and P';
- $D(M, M')$ is the number of methods contained in P not in P' plus the number of methods in P' not in P;
- $D(\text{SNL}, \text{SNL'})$ is the difference between the number of labeled relations arcs existing in P and in P';
- w_a, w_m, w_t, and w_r are weights, $w_a+w_m+w_t+w_r = 1$.

Definition 2. Given two methods $m(a_1, a_2, ..., a_n)$ and $m'(a'_1, a'_2, ..., a'_m)$, the difference between m and m' is:

$$d(m, m') = \begin{cases} 0 & \text{if } m \text{ and } m' \text{ are same methods with} \\ & \text{the same values of the attributes} \\ 1 & \text{otherwise} \end{cases}$$

This definition is used to evaluate $D(M, M')$.

Definition 3. Given a set of goal tasks, R, that is different from, but related to the original goal task set, R_0 , that has a solution as P_0, we define that a planning TÆMS solution, P_1, has a greater stability than another solution, P_2, if P_1 and P_2 satisfy $D(P_0, P_1) < D(P_0, P_2)$.

3. TASK STRUCTURE BASED PLAN ADAPTATION

A plan failure is grouped into one of the classes (1)-(4) and repair strategies are suggested. (1)**Failing leaf nodes:** If an action fails because of any unavailable or unsuitable agent/resource, replace it with an alternative one or add a new task to re-achieve the agent /resource; (2)**Failing task nodes:** If any external pre-condition of a task fails, it can be simply re-achieved by a new task. If any post-condition of a task cannot be achieved, repair its subtasks or re-decompose it; (3) **Failing interdependence relations:** If any relation arc cannot be satisfied, remove it and modify any node at the two ends; (4)**New incoming tasks:** Find a plan solution for this new incoming task, then merge it into the original task structure. Whenever any repair strategy is taken, the related interdependence arcs have to be updated.

We define the following equation (2) to guide a repair process. This function approximately evaluates how the current partially refined plan is affected by a repair candidate, r, that is used to repair a failure point, fp. In the equation, P_0 is the original TÆMS solution, P is the current partially refined solution, P_r is the solution after the repair r, and the weights $\eta_d + \eta_f + \eta_r = 1$.

$$E(r,\text{fp}) = \eta_d \times [D(P_0, P_r) - D(P_0, P)] + \eta_f \times FailurePoints(P, r) - \eta_r \times RepairedPoints(P, r) \tag{2}$$

4. PRELIMINARY EVALUATION

To help make this notion of stability more concrete, we implement a variant of the producer-consumer-transporter domain [3] with DECAF [?]. We introduce three kinds of transporting agents as shown in table 1. A producer generates simple objects S, M, L, where the volume of M is twice the volume of S and L is four times the S volume. A consumer makes complex objects (X, Y, W, Z, V) following the rules: $S \to X$, $S + S \to Y$, $M \to W, M + M \to Z$, $L \to Z, L \to V$. Figure 1 shows a simple example: two producers supply resources to two consumers and transporters may pick up different routes to deliver resources from a producer to a consumer.

In order to evaluate our approach, we generate plans from scratch for 10 miscellaneous problems consisting of multiple tasks; then the plans are broken by changing the goals, adding new tasks, or making an action fail. These broken plans are fixed by using the repair strategies with stability bias or replanning from scratch. Figure 2 shows the comparison of time consumed to repair a plan that contains failures at the goal task level.

Table 1: Transporting agent type and ability

Transporting agent type	Transporting ability
t_s	$\leq$ volume of S
t_m	$\leq 2\times$ volume of S
t_l	$\leq 4\times$ volume of S

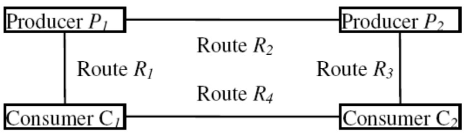

Figure 1: A simple example case of PCT domain.

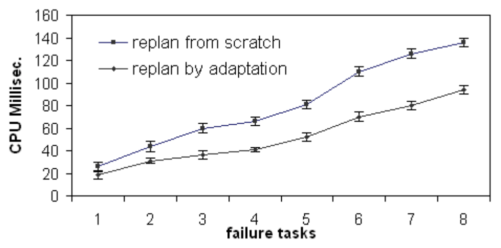

Figure 2: CPU times needed to find a solution (considering the average value over 5 runs).

5. REFERENCES

[1] T. Bartold and E. Durfee. Limiting disruption in multiagent replanning. In *Proc. of the 2nd Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS-03)*, 2003.

[2] M. Fox, A. Gerevini, D. Long, and I. Serina. Plan stability: Replanning versus plan repair. In *Proc. Of the 16th Int. Conf. on Automate Planning and Scheduling (ICAPS-06)*, 2006.

[3] B. Horling, B. Benyo, and V. Lesser. Using self-diagnosis to adapt organizational structures. In *Proceedings of the 5th Int. Conf. on Autonomous Agents (AAMAS-01)*, 2001.

[4] V. R. Lesser, K. Decker, T. Wagner, and et al. Evolution of the gpgp/tæms domain-independent coordination framework. *Autonomous Agents and Multi-Agent Systems*, 9(1-2):87–143, 2004.