Proceedings of the Ninth International
Symposium on Combinatorial Search (SoCS 2016)

Trial-Based Heuristic Tree-Search for Distributed Multi-Agent Planning

Tim Schulte
Institut für Informatik
Albert-Ludwigs-Universität
Freiburg, Germany
schultet@cs.uni-freiburg.de

Bernhard Nebel
Institut für Informatik
Albert-Ludwigs-Universität
Freiburg, Germany
nebel@cs.uni-freiburg.de

Abstract

We present a novel search scheme for privacy-preserving multi-agent planning, inspired by UCT search. We compare the presented approach to classical multi-agent forward search and evaluate it based on benchmarks from the *CoDMAP* competition.

Introduction

In collaborative multi-agent planning multiple agents attempt to achieve a common goal by planning and coordinating their actions appropriately. In this paper, we introduce a novel search technique for privacy preserving distributed multi-agent planning based on *trial-based heuristic tree-search* (THTS) (Keller and Helmert 2013), a general scalable framework for solving different types of planning tasks. Our main contribution is the definition of the resulting search framework, which we call *distributed multi-agent trial-based heuristic tree-search* (DMT). This framework extends the way of how distributed plans can be generated and so might be useful for portfolio approaches to multi-agent planning. We exemplify two DMT algorithms and provide preliminary empirical evaluation.

Background

A *multi-agent planning task* consists of a finite set of n agents $\{\varphi_i\}_{i=1}^n$, a finite set of *state variables* V with finite domains, a variable assignment s_0 over V called the *initial state*, a partial variable assignment $s_\star$ over V called the *goal*, and a finite set of actions A_i for each agent φ_i. Each *action* a is specified via *precondition* $pre(a)$ and *effect* $eff(a)$, both being partial variable assignments over V. An action a is *applicable* in state s if s agrees with $pre(a)$ wherever $pre(a)$ is defined. Application of action a in state s yields the *successor state* $a(s)$ which agrees with $eff(a)$ where $eff(a)$ is defined, and agrees with s, elsewhere. The solution to a planning task is a sequence of actions $\pi = (a_1, \ldots, a_k)$ such that a_1 is applicable in s_0, every subsequent action is applicable in the state generated by its preceding action, and $a_k(\ldots(a_1(s_0))\ldots) \models s_\star$.

Privacy constraints are defined in terms of *private variables* and *private actions*. The private variables of an agent φ_i can only be observed and be affected by actions of φ_i. Private actions of φ_i are only known to φ_i and only depend on or affect private variables of φ_i. Public actions can affect or depend on both public and private variables of the agent. Therefore, other agents can only access *projections* of φ_i's public actions, where private variables are hidden.

Trial-based Heuristic Tree-Search (Schulte and Keller 2014; Keller and Helmert 2013) algorithms maintain a tree of search nodes and select one of its leaf nodes for expansion in each search step. Three phases are executed repeatedly; their specific behaviour must be defined to derive concrete algorithms. *Selection* is the first phase of the algorithm with the objective to select one of the leaf nodes for expansion. Beginning from the root, a selection strategy recursively selects a child, until a leaf node is reached. In the *initialization* phase, the previously selected leaf node is initialized. Successor nodes are generated and integrated into the tree. During the *backpropagation* phase new information, like value estimates or the number of times a node has been visited during selection, is propagated through the tree. After the backpropagation phase, the algorithm starts again with the first phase. This process is repeated until a goal state is generated, or some limit is reached. By alternatingly executing selection and initialization phase, multiple nodes can be initialized in each search step. The number of nodes that get initialized in a single search step is denoted as *trial length*.

Distributed Multi-Agent THTS

We now present a complete and privacy preserving scheme for the distributed application of trial-based heuristic tree-search. The concept is similar to MAFS (Nissim and Brafman 2014), where forward-search is concurrently executed while state information is exchanged between the planning agents according to a specific message passing scheme. Each agent performs THTS locally, using its own actions only. Whenever agent φ_i expands a state s in which a public projection of an action of φ_j is applicable, φ_i will send a message to φ_j containing s. φ_j then integrates s into its search tree, such that it can prospectively select s for expansion. To accomplish this, φ_j identifies a suitable parent and adds s as a child to it. In principle, any node can be used as a parent without soundness or completeness being compromised. However, since the tree structure is crucial to the success of THTS algorithms, it is important where new states are inte-

---

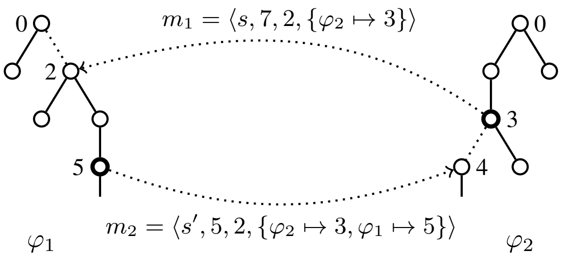

Figure 1: State integration.

grated. Let s be the result of applying the sequence of actions $(a_1, \ldots, a_k)$ in the initial state, i.e. $a_k(...(a_1(s_0))...) = s$. If φ_j has no action in the sequence, it adds s as a child to the root. Otherwise, let a_j be the last action of φ_j in the sequence. Then, φ_j adds s as a child to $s' = a_j(...(a_1(s_0))...)$. Note that φ_j is not aware of all actions in the sequence leading to s and hence cannot compute s'. We enable φ_j to identify s' by using a special message type.

Definition 1 (State message). *A* state message *from φ_i to φ_j for state s is a tuple $m = \langle s, h_i, g_i, T\rangle$, where s is a state, h_i is a value estimate of φ_i for s, g_i is the cost of φ_i to establish state s from the nearest* ancestor state *contributed by φ_j (or the root if no such state exists), and T is a set of* state tokens. *Private components of s are encrypted, such that each agent can only decrypt its own private components.*

Each *state token* belongs to an agent φ_k and contains a state identification number. This number references a node in the local search space of φ_k and is meaningless to all other agents. Figure 1 illustrates how tokens are used to integrate states. Numbers next to nodes depict state IDs that correspond to the local state represented by the node. Nodes associated with states for which the other agent has an applicable public projection are rendered in bold. When φ_2 initializes the node with state ID 3, it transmits message m_1 to φ_1, containing a token that enables φ_2 to identify the node. When φ_1 receives m_1, it creates a new search node for s. Because m_1 contains no token for φ_1, the new node is attached as a child to the root. Later on, φ_1 initializes the node labelled with 5 and sends message m_2 to φ_2. Because the state with ID 5 was generated in a branch to which φ_2 contributed an *ancestor* state, the token $\varphi_2 \mapsto 3$ is attached to the message, along with the new token $\varphi_1 \mapsto 5$ of φ_1. On receiving m_2 φ_2 looks up its token $\varphi_2 \mapsto 3$, creates a new node for s', and attaches it as a child to the node with state ID 3.

Algorithms. We propose two DMT algorithms, which only differ in the selection strategy used. Both algorithms generate all successors in the initialization phase and propagate the best (minimum) value estimate in the backup phase. *DMT-BFS* selects the successor node with the best value estimate. *DMT-UCB* uses a balanced selection strategy based on UCB1 (Kocsis and Szepesvári 2006). Both algorithms were shown to be sound and complete (Schulte and Nebel 2016) when privacy constraints of MA-STRIPS (Brafman and Domshlak 2013) apply.

mafs	*dmt-bfs*$_1$	*dmt-ucb*$_1$	*dmt-bfs*$_{100}$	*dmt-ucb*$_{100}$
80	69	52	**92**	84

Table 1: Coverage. DMT algorithms use a trial length according to their indices (1 or 100).

Evaluation

DMT and MAFS were both implemented in a distributed multi-agent planning system using the FF heuristic (Hoffmann and Nebel 2001). Experiments were run on a PC with a quad-core CPU and 4 GB of RAM. Table 1 shows coverage results on 240 planning tasks from the CoDMAP competition (Štolba, Komenda, and Kovacs 2015) with a time limit of two minutes per task. The results show that both DMT configurations have significantly higher coverage when a trial length of 100 is used and could even outperform the MAFS approach. Increasing the trial length causes additional exploration and encourages faster escape from local minima. A portfolio planner running *dmt-ucb*$_{100}$, *dmt-bfs*$_{100}$ and *mafs* for 2 minutes each solves 117 instances, which shows that MAFS and DMT complement each other well.

Conclusion

In this paper we presented DMT, a novel and privacy preserving scheme for distributing THTS algorithms. We proposed two concrete DMT algorithms and evaluated them empirically. Results have shown DMT and MAFS to complement each other in a promising way. In future work we will create and analyze new DMT algorithms to further exploit such complementary strengths.

References

Brafman, R. I., and Domshlak, C. 2013. On the complexity of planning for agent teams and its implications for single agent planning. *Artificial Intelligence* 198:52–71.

Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research (JAIR)* 14:253–302.

Keller, T., and Helmert, M. 2013. Trial-based heuristic tree search for finite horizon MDPs. In *Proceedings of ICAPS 2013*.

Kocsis, L., and Szepesvári, C. 2006. Bandit based monte-carlo planning. In *Proceedings of ECML 2006*, 282–293.

Nissim, R., and Brafman, R. I. 2014. Distributed heuristic forward search for multi-agent planning. *Journal of Artificial Intelligence Research (JAIR)* 51:293–332.

Schulte, T., and Keller, T. 2014. Balancing exploration and exploitation in classical planning. In *Proceedings of SoCS 2014*.

Schulte, T., and Nebel, B. 2016. Trial-based heuristic tree-search for distributed multi-agent planning. In *Proceedings of DMAP 2016*.

Štolba, M.; Komenda, A.; and Kovacs, D. L. 2015. Competition of distributed and multiagent planners (CoDMAP). In *Proceedings of WIPC 2015*, 24–28.