State-Space Reduction through Preference Modeling*

Radosław Klimek, Igor Wojnicki, and Sebastian Ernst

AGH University of Science and Technology
Department of Applied Computer Science
Al. Mickiewicza 30, 30-059 Kraków, Poland
{rklimek,wojnicki,ernst}@agh.edu.pl

Abstract. Automated planning for numerous co-existing agents, with uncertainty caused by various levels of their predictability, observability and autonomy, is a complex task. One of the most significant issues is related to explosion of the state space. This paper presents a formal framework which can be used to model such systems and proposes the use of formally-modeled agents’ preferences as a way of reducing the number of states. A detailed description of preference modeling is provided, and the approach is evaluated by examples.

1 Introduction

Automated planning is an active research field, with applications ranging from motion planning [1], through resource allocation and scheduling, to coordination of autonomous agents [2]. This paper presents a formal framework which can be used to model sophisticated systems, featuring heterogenous entities (agents), characterized by varying levels of autonomy, predictability and observability, and proposes the use of preferences to reduce the size of the state space.

The paper is organized as follows. Section 2 provides a brief introduction to automated planning and the state-space representation of planning problems. Section 3 presents the framework, indicating certain and uncertain knowledge elements, and provides intuition how this model can be mapped to real-world cases. Section 4 introduces a formal tool which can be used to model preferences (or agent constraints) using temporal logic, and Section 5 provides an example how formally represented preferences can be used to reduce the state space.

2 Motivation and State-of-the-Art

The most general, intuitive definition of planning is that it is the reasoning part of acting [3]. Planning can be performed by people, either implicitly or explicitly. Automated planning is a branch of AI which is concerned with computation and execution of plans by machines.

* This work is supported by the Polish National Science Centre (NCN) grant 2011/01/D/ST6/06146.
A common conceptual model for planning is a state-transition system, also called discrete-event system. It can be defined as a 4-tuple \( \Sigma = (S, A, E, \gamma) \) \[3,4\], where:

- \( S = \{s_1, s_2, \ldots \} \) is a set of states,
- \( A = \{a_1, a_2, \ldots \} \) is a set of actions,
- \( E = \{e_1, e_2, \ldots \} \) is a set of events,
- \( \gamma : S \times A \times E \to 2^S \) is a state-transition function.

This definition is often used together with a graph model, where states \( s \in S \) are nodes, and state transitions (given as pairs \( (a, e), a \in A, e \in E \) ) are directed edges. The semantic difference between actions and events is that actions are applicable to states by the plan executor (if \( \gamma(s, a) \neq \emptyset \) ), while events are contingent – they might occur due to the system’s characteristics, and every event \( e \) which takes the system from state \( s \) to state \( s' \) must have a corresponding transition function \( \gamma(s, e) = \{s'\} \). In a typical instance of a planning problem, the system is in some initial state \( s_I \in S \), and the goal is to take it to one of the goal states \( s_G \in \{s_{G1}, s_{G2}, \ldots \} \subset S \).

Based on that definition, derivation of a plan consists in finding a sequence of state transitions which take the system from the initial state to one of the goal states. Numerous methods can be used here – either “blind” (uninformed) ones, which do not take the characteristics of the system into account \[4\] sec. 3.4] or informed (heuristic) methods, which are especially useful for large state spaces \[5\]. Determination of heuristics for domain-independent planning is a problem which has recently received a lot of attention \[6,7,8,9\].

The size of the state space and the complexity of the planning process can grow rapidly, due to several reasons:

1. **Partially predictable action (and event) results.** If it is unclear which state the system will be in after an action is taken or an event occurs, the planner will have to consider all possible outcomes, which may result of a combinatorial explosion of the search tree.
2. **Partially determinable world state.** If the world state cannot be fully observed, the planner needs to assume all possible states which match the observations.
3. **Multiple agents.** The size of the state space grows exponentially with the number of entities (agents), as the local state of an agent can be combined with almost all states of every other agent. Planning for such a case (multi-body) is not different than for a single agent though \[4\].
4. **Multi-variant plans.** Sometimes, it may be preferable to provide multiple possible actions to the agent and let it choose the most favorable option. This is true especially in situations, where the planner’s observations of the system state are less detailed than (local) observations made by the agents.

In needs to be noted, that the state space is not a Cartesian product of all possible states. Some states can be unreachable due to world constraints (obstacles) or agent constraints restricting its behavior. Defining these constrains decreases planning process complexity.