

The Effect of Planning State Space Topology on Search Performance

Blazej Bulka and Marie desJardins and Fusun Yaman

Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250
{bulka1, mariedj, fusun}@cs.umbc.edu

Abstract

The topological characteristics of the state space graph for a planning problem are related to the internal structure of the underlying problem, and can affect the efficiency of planning. We outline a proposed approach for using this knowledge to build a novel type of plan library (which we call a *planning backbone*) that organizes states hierarchically based on their connectivity.

Introduction

Many factors influence a planner's performance in a planning domain. Certain benchmark domains represent inherently hard computational problems (Helmert 2006); that is, finding a plan is NP-hard (at least). Other factors are related to the *domain representation*, including expressivity of the description language, number of available actions, use of object types, regularities in the state space, and goal structure (Roberts & Howe 2006; Porteous, Long, & Fox 2004).

Our research goal is to improve the execution of domain-independent planners by enabling them to analyze a planning domain, and look for regularities in its search space. The first step towards achieving this goal is to identify useful topological features for characterizing planning domains and planning problems. The next step is to design a method for analyzing planning domains effectively in order to gather information about these features. Finally, the planning algorithms must be modified to use the feature information in order to improve their performance.

We are mostly interested in features that are directly related to the planner's search space. While the planner's ability to navigate its search space efficiently has a significant impact on the overall performance (e.g., by using efficient heuristics for guidance), the size and shape of the search space are also important. We focus on state-space planners because the shape of their search space is the same as the shape of the state space. Specifically, the structure of the state space depends only on the structure of the underlying planning problem and its description, rather than on the particular algorithm used in the planner. Therefore, the results should be more universal and not limited to a particular class

of state-space planners. Moreover, we only use the information about the topology of the space assuming that the states are *opaque* to the analysis (no internal structure of the state is used). As the result, the proposed approach should be applicable to heuristic search more broadly, not just to planning.

Information reuse between planning episodes

Analyzing the topology of the state space graph can be computationally expensive. However, it is reasonable to expect that this additional expense can be amortized across multiple planning episodes. Most real-world planners will be used to generate multiple in a given domain. The cost of specifying of a planning domain, which requires the involvement of a human expert and possibly time-consuming debugging, make "single-use" domains unrealistic. Moreover, many planning problems within a planning domain will have the same state space, if they involve the same objects (possibly in different configurations) and the same static facts (statements whose value cannot be changed by any action). For example, in real-world logistics domains, the locations of cities, airports, depots and the number of trucks will remain the same for a long time, and the analysis can be performed for some maximum expected number of packages. Additionally, even when two state spaces differ, parts of the first state space may be similar to parts of the second state space (e.g., through subgraph isomorphisms).

Using features to build more efficient plan libraries

Our ultimate goal is to design a novel type of plan library called a *planning backbone*. The library would consist of reasonably sized fragments of the state-space graph. Ideally, the fragment of the state-space graph should consist of frequently traversed states, and would be shaped in such a way that most of the states in the state space are close to it. More information about this idea is provided later in the section on planning backbones.

Topological features of state spaces

As mentioned earlier, we can think of a state space as a very large, finite graph defined implicitly by the domain definition and the planning problem. This allows us to define multiple properties of the graph based on graph theory, which in turn can improve the speed of planning.

The basic problem with analyzing the state space graph is its size, which in practice prohibits its full analysis for realistic planning problems. However, some features may be easily approximated without significant exploration, either through analysis or by gathering data about the connections between the expanded states during normal planning operation. These features include: distribution of degrees of nodes (e.g., *hub nodes*), distribution of distances between the nodes, and probability of a node having a certain predicate satisfied.

There are also other properties that can be approximated without the knowledge of a significant portion of the state space, but they may require performing additional computations or preplanning rather than just reusing information gathered during planning (e.g., proactive exploration and using random walks on the generated portion of the graph). The most interesting features include: the diameter of the search space, and the probability of visiting a state during a random walk.

Unfortunately, there are also a few useful properties that cannot be approximated without significant exploration of the state space. These include *symmetries* and *almost symmetries* (Porteous, Long, & Fox 2004), all-pairs distance between the states, and localization of nodes satisfying certain predicate.

Planning backbone

Our goal is to use the knowledge of state space topology to build a novel type of plan library called a planning backbone. The library would store fragments of the state space that are frequently used or positioned in important locations. A planner using such a library would first find a path from the initial state to the backbone, then find a path within the precomputed states, and finally compute the rest of the path outside the backbone to a goal state.

Building such a structure presents multiple challenges. One issue is how one could search efficiently within the data structure. Below, we describe a solution that organizes the states in the library into a hierarchy. Another issue is managing the size, and deciding which states should be included.

Searching within the backbone In general, searching for a path in an already precomputed graph is faster than reasoning about preconditions and effects of actions, and creating new planning states. Moreover, once a path in the graph is determined, extracting a sequence of actions is trivial. Nevertheless, efficient finding of a path in the graph of this size may be challenging. An additional challenge is determining the part of the backbone that is closest to a goal state. (If multiple parts are selected as potentially closest, this may require computing multiple paths.)

We plan to solve the issue of finding the paths by organizing planning states into *hierarchical nodes* in such a way that the hierarchy depends on state space connectivity. Initially, a node (usually a highly interconnected one; i.e., a hub node) would be selected as a center of a new hierarchical level, and all its direct neighbors (either state-space states or other hierarchical nodes) would be added to it. It is worth noting that

the center of the newly created node would not have any direct connections outside of the node (all direct neighbors are already included), and all external edges of the non-central nodes would determine the neighbors for the new hierarchy level. This procedure would then be applied recursively until only one top-level node is left for each contiguous fragment of the statement.

The hierarchy of state-space states built in the way described above allows efficient search of paths between any two nodes in the hierarchy. All such nodes have a common parent, and connectivity at each level of the hierarchy is guaranteed through the center node. While this algorithm does not produce paths that are optimal in terms of length, the way the center nodes are selected during the hierarchy building affects the balance of the tree, and therefore affects the average length of generated paths. The path length can also be influenced by the topology of the underlying state space (e.g., densely or sparsely interconnected graphs). We have identified multiple localized policies, and our immediate next step is to evaluate them using a variety of more realistic benchmark problems.

Size management of the backbone Applying the backbone idea requires efficient management of its size, and deciding which states should be included, so that it provides good “coverage” of the state space (i.e., most of the states are close to the backbone). We expect the backbone to be built incrementally across multiple planning episodes (a form of experience reuse). This will allow us to take into account the frequency with which states are encountered. We can also use proactive random walks in the parts of the state space. Yet another approach may be to take advantage of symmetries in the space and possibly to parameterize the data structure.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0325329. I would like to thank Daniel Bryce, Jeremy Frank, Derek Long, and Mark Roberts for useful thoughts and conversations.

References

- Helmert, M. 2006. New Complexity Results for Classical Planning Benchmarks. In *Proceedings of ICAPS-06*, 52–61.
- Porteous, J.; Long, D.; and Fox, M. 2004. The Identification and Exploitation of Almost Symmetry in Planning Problems. In *Proceedings of the 23rd UK Planning and Scheduling SIG*.
- Roberts, M., and Howe, A. E. 2006. Directing Portfolio with Learning. In *Proceedings of the AAAI 2006 Workshop on Learning for Search*.