

A Planning.Domains Plugin for Heuristic Visualization

Caitlin Aspinall,* Cam Cunningham,* Ellie Sekine,* Christian Muise

School of Computing
Queen's University
Kingston, Ontario, Canada

{16cea5, 16cjc12, 17ees, christian.muise}@queensu.ca

* *Equal contribution and listed alphabetically*

Abstract

Heuristics are at the heart of every planner. From the simple to the complex, they drive both satisficing and optimal planning approaches. Despite their pervasiveness, there has been relatively little effort towards systematically visualizing them. We introduce a plugin for the online editor at Planning.Domains that is capable of visualizing the heuristic computation of a manually explored state space. Our initial implementation demonstrates the h^{add} heuristic, but the framework serves as an extensible base for other heuristics in the field of automated planning.

1 Introduction

Modern planners are massive and complex systems that have decades of advanced engineering practices built in. This complexity leads to some difficulty when learning and understanding the way that these systems work. As it stands, there are few ways to visually interact with a planning state space. Existing space planning visualizers can only process small domains due to the large nature of the potential space tree (Magnaguagno et al. 2017): another problem that impedes truly understanding this process. There are no current tools that visually and interactively explain these core concepts of modern planners. Specifically, the field lacks any tools that visualize the common *heuristic computation* in planners. Shy of manually writing out heuristic calculation updates by hand, there is no mechanism for students learning these techniques to readily visualize the rather complex processes of heuristic computation.

To address this gap, we created a plugin for the online Planning.Domains Editor (Muise 2016). It is capable of two main functionalities: dynamic state space traversal and heuristic visualization. The plugin allows for parsing and processing of PDDL files in the browser, the generation of an interactive state space, and an interactive visualization of heuristic computation. This tool leveraged various libraries to aid in data representation and PDDL processing, including D3.js for visualization (Bostock 2012) and a browser-embedded version of the Tarksi planning library (Ramírez and Francès 2021). We detail each of these two functionalities in the remainder of the paper.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

2 State Space Exploration

While AI Planning state space traversal tools already exist, there were some key considerations when designing our system that justified the undertaking of the project. Firstly, the visualization had to load the state space dynamically, as needed, rather than all at once during the initial processing of the problem and domain. Limited by the small amount of processing power available in a browser, it was impossible to enumerate the immense number of reachable states in a planning problem. Additionally, we wanted the user to have an interactive way to explore a domain's state space. Elements such as clicking to expand on a state to reveal its successors, and hovering to display the states' fluents were key considerations for the plugin.

A screenshot of the visualization is shown in Figure 1. This view is shown to the user once they select the domain and problem file from the editor that should be visualized, and the user can subsequently expand nodes of interest to have the states dynamically created and added to the interaction. Computation of heuristic values and descriptions of the state are also possible, but not shown.

An additional feature that we would like to see implemented is an automatic search process, using a selected searching strategy. We believe that this would also be useful in an educational setting when teaching algorithms like depth-first and breath-first searches.

3 Heuristic Visualization

The second key part of this plugin is the ability to visualize heuristic value computation for a specific state in the explored state space. This functionality is invoked – with a new tab being opened with the heuristic visualization – when a state is double-clicked in the interface described in the previous section.

We chose to focus on the h^{add} heuristic for our initial heuristic visualization in the plugin for several reasons. Firstly, it is a relatively simple heuristic, which is ideal for teaching purposes, as well as keeping the visualization relatively simple. Secondly, another benefit of this choice is that its visual representation is common among heuristics. Since the aim of this plugin was to create a framework where others could add their heuristics to be visualized, choosing a

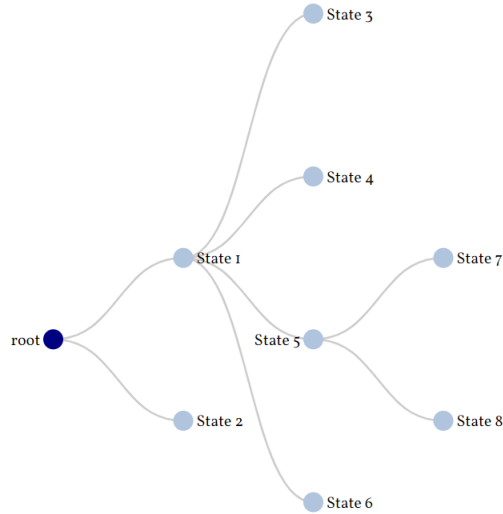


Figure 1: State space visualization.

heuristic that was representative of many existing ones was crucial to the success of this visualization.

When the interface is opened, the users are prompted with a high level view of each fluent and action, their connections through action preconditions/effects, and the goal node. Each node has its (intermediate) heuristic value visible, and the user can click on any node to update the graph based on the heuristic’s computation – that is to say, propagating the values of predecessor nodes to the currently selected one. When the pointer hovers over a node, all items except for that node and its predecessor nodes are turned opaque for clarity. The nodes in the graph can be moved around for exploration, and then frozen in place when a satisfactory alignment is found.

A screenshot for the heuristic visualization can be seen in Figure 2. Red nodes correspond to ground actions, blue nodes correspond to fluents, and the gold node corresponds to the goal (which can be viewed as an action with the precondition set to the original goal and no effects). Edges connect fluents-to-actions (capturing preconditions) and actions-to-fluents (capturing effects).

Every ground fluent and action will appear as a node in the graph. The force-directed layout naturally partitions these core elements of the planning problem, and a natural clustering emerges. To a large extent, these follow the stratification of shortest causal chains from the goal node.

There are a variety of ways we hope to improve this heuristic visualization. The first, and most important issue is that of visualizing larger graphs. With its current implementation, larger graphs are hard to understand due to the volume of fluent and action nodes in the visualization, as such, the purpose of the plugin is diminished in these cases. The hope is to improve the ability to represent these large

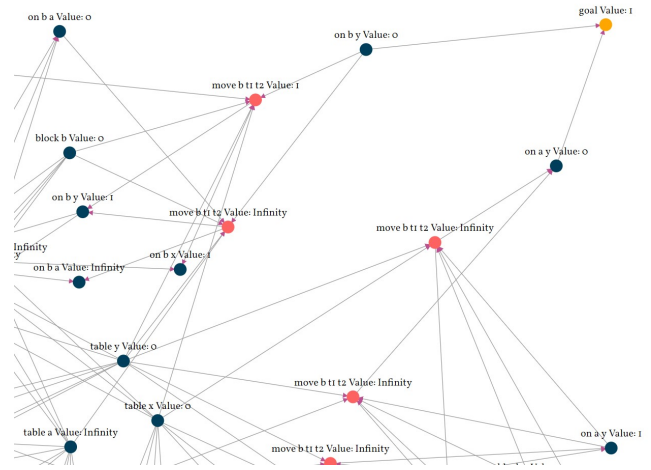


Figure 2: h^{add} heuristic visualization

spaces in an intuitive manner. Another feature that we would like to see added is dynamically modifying the updates. Currently, the only way to update a node’s heuristic value is for the user to click on it or for the entire process to be run to a fixed point. Ideally, a user could start and stop the automatic auto cascading of heuristic updates throughout the graph. Between user initiated updates, and auto updates, this visualization would allow for a interactive, hands-on experience with the heuristic.

4 Summary

We have introduced a plugin for the online Planning.Domains editor that allows users to (1) interactively explore the reachable state space of classical planning problems; and (2) visualize the h^{add} heuristic computation for any of those reachable states. The goal and motivation for this plugin, as well as the editor as a whole, is to provide a central location for learning and developing modern planning techniques. This plugin, available in the online editor’s list of featured plugins, contributes to the growing ecosystem of tools for planning educators and researchers. It lays the ground-work for a broad suite of heuristic visualizations and interactions that are embedded within the online editor.

References

- Bostock, M. 2012. D3.js - Data-Driven Documents. URL <http://d3js.org/>. Accessed on 2021-05-17.
- Magnaguagno, M. C.; Pereira, R. F.; Móre, M. D.; and Meneguzzi, F. 2017. WEB PLANNER: A tool to develop classical planning domains and visualize heuristic state-space search. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning, UISP*, 32–38.
- Muise, C. 2016. Planning.Domains. In *The 26th International Conference on Automated Planning and Scheduling - Demonstrations*.
- Ramírez, M.; and Francès, G. 2021. Tarski. URL <https://github.com/aig-upf/tarski>. Accessed on 2021-05-17.