# Advanced Artificial intelligence - ENHSP Solver

Romain André - Clément Chamayou - Joseph Thibault

January 22th 2023

## 1 Introduction

ENHSP is a search-based planning algorithm that uses heuristic search to find a plan for a problem described in PDDL. The planner takes the problem description and converts it into a graph-search problem, where each node represents a state of the system. The planner uses a heuristic function to guide the exploration of the state space. The heuristic function is used to estimate the distance between a given state and the goal state, and the planner favors states that are closer to the goal state.

## 2 How does it work

The planner builds the graph with an incremental-forward way, starting from the initial state and expanding the graph by adding new states that can be reached from the current state. The planner continues to expand the graph until a goal state is reached. The planner uses a best-first search strategy, where it explores the state that has the highest heuristic value (i.e., the state that is closest to the goal state) first.

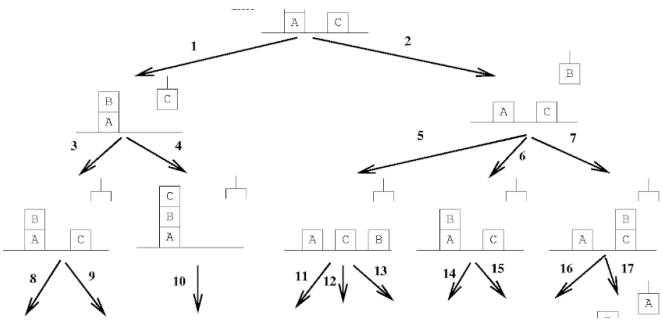

FIGURE 1 – Search expansion tree.

### 2.1 Search Strategy

The default search strategy function used in ENHSP is WA*, which is based on the popular A* search algorithm.

#### 2.1.1 A*

The algorithm A* starts at the initial node and explores all possible paths by maintaining a priority queue of nodes to visit next, where the priority is determined by a cost function that takes into account both the distance from the starting point g(n) and the estimated distance to the goal h(n). The algorithm terminates when the goal is reached or when no more nodes can be expanded. It is guaranteed to find the optimal path if the heuristic function h(n) is admissible (never overestimates the distance to the goal) and consistent (satisfies the triangle inequality).

$$\text{f(n)} = g(n) + h(n) \tag{1}$$

With n the next node on the path, f(n) the cost of the next node on the path, g(n) the cost of the path from the start node to n, and h(n) a heuristic function that estimates the cost of the least expensive path from n to the goal.

#### 2.1.2 WA*

In the WA* search strategy however, the heuristic function h(n) is multiplied by a constant W>1.

$$\text{f(n)} = g(n) + W * h(n) \tag{2}$$

The weighted A* search focuses on states that are closer to the goal, giving up the guarantee of finding the optimal solution. However it allows for a significant lowering of time and memory requirements.

### 2.2 Heuristics

ENHSP uses heuristics that can reason effectively over numeric problems. Numeric planning problems are those in which the variables can take on numeric values, such as "move the robot to a location that is x meters away from its current position." Traditional planning heuristics are not well-suited for such problems, as they often rely on symbolic reasoning. ENHSP uses specialized heuristics that can reason about numeric variables,

such as heuristics which can estimate the distance to the goal state based on a linear approximation of the problem.

### 2.3 Search Expansion

At each step, the algorithm selects the node with the lowest cost (or highest priority) from the priority queue and expands it by adding its neighbors to the queue. Then the cost of each new neighbor is calculated by adding the cost of reaching that node from the current node to the estimated cost of reaching the goal from that node (as determined by the heuristic function). The neighbors are considered as the child nodes of the current node which are used to expand the search space.

If the algorithm determines that the goal cannot be reached from the current path, it will backtrack to the previous node and try a different path. This process of backtracking and re-exploring different paths is repeated until the goal is found or until it is determined that the goal is not reachable. The WA* algorithm uses an informed search strategy, that is guided by the heuristic function and the cost function, to explore the search space more efficiently and to avoid getting stuck in paths that do not lead to the goal.

## 3 Examples

The ENHSP solver takes two PDDL (Plan Domain Definition Language) files as input. A domain for the problem and the problem itself. It will then (if the arg -sjr was used) generate a sp_log file that is the result of the search expansion.

### 3.1 PDDL Domain

The PDDL Domain file, defines the main environment in which the problems will take place. This means defining, the names of the variables, the predicates, the functions, the actions, the processes and the events that can be used to describe the problem.

We chose to describe an example domain called "paco3d" that is a model for a a 3D robotic arm that can move in two directions, increase or decrease, with different speeds and is able to be blocked or unblocked. The robotic arm has two types of links. The predicates include things like whether a link is connected to another link, if it is free to move, and if it is increasing or decreasing in angle with respect to a specific axis. The actions include starting and stopping movement in the arm, increasing or decreasing angle of a link and affecting the angle of a link. The processes include moving the angle of a link in the arm and moving the angle of the gripper.

### 3.2 PDDL Problem

The PDDL Problem file describes the objects, the init state and the goal of the problems.

In our example, the problem specifies four objects, "L1", "L2", "L3", and "L4", which are links in the arm, and one object "xyaxes" and "ZAXES" which are axis. The problem also defines the initial conditions for the arm, such as the angles of each link with respect to the axes, the speeds of the links, and the connectivity and affectivity between the links.

The goal of the problem is to achieve a specific angle for the link "L3" with respect to the "xyaxes" and "ZAXES". The planner is expected to find a plan to move the arm such that the goal conditions are met.

### 3.3 Tree Visualizer

The website we developed is used as a tool to visualize the search space tree. Here is an example of the rendering of the file "P4_i1.pddl.sp_log" generated via the ENHSP solver with the domain and problem previously described.

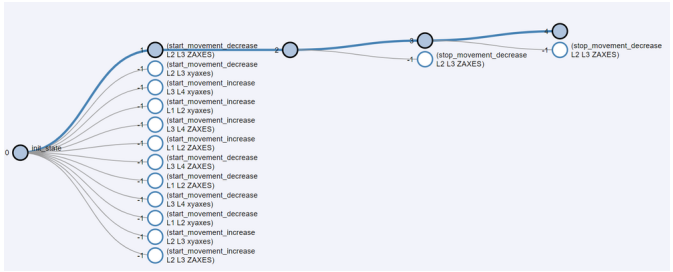


FIGURE 2 – Tree visualization from our tool.

On the left of the tree we can see the root node which is the init state. The next column of nodes displays all the different options considered. Among them the first one is selected because the WA* search algorithm has established that the distance (3.15) and the cost of 1 is the best solution to reach the goal state according to the heuristic. The link between the 2 nodes is consequently highlighted in blue to display the choice of the algorithm. At each step this method continues until finally reaching the goal (top right on this tree).

This example is pretty easy to understand but it could have been less straightforward with some backtracks like the following example on another problem.

FIGURE 3 – Tree with backtracks.

A branch can be developed deeply (bottom one) but suddenly WA* algorithm realizes that this path is more expensive than a previous one. In this case the algorithm goes back in the search tree to develop a more interesting path.

# 4 Conclusion

In conclusion, the ENHSP solver is a powerful search-based planning algorithm that utilizes heuristic search to find a plan for problems described in PDDL.

The planner converts the problem into a graph-search problem and uses a heuristic function to guide the exploration of the state space. By default, the algorithm uses the WA* search strategy, which allows for a reduction in time and memory requirements. ENHSP also uses specialized heuristics that can reason effectively over numeric problems and estimate the distance to the goal state based on a linear approximation of the problem.

Overall, this project provides a valuable tool for solving planning problems with numeric variables and can be applied to a wide range of domains such as the ones seen in the examples.