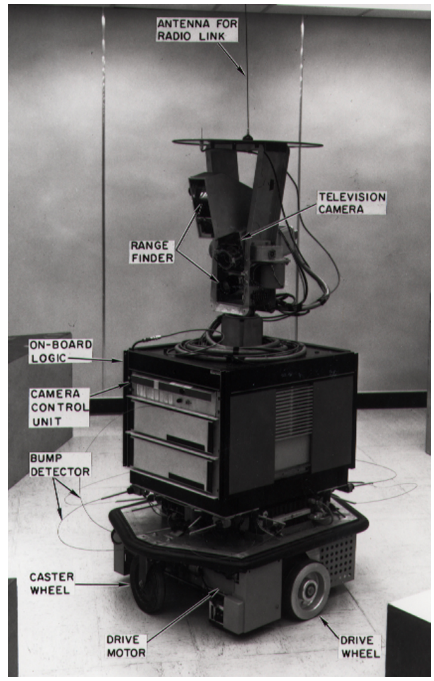
AIND

RESEARCH REVIEW

GIL AKOS // 7 JULY 2017 // PROJECT: IMPLEMENT A PLANNING SEARCH

STRIPS



STRIPS (STanford Research Institute Problem Solver) was developed as the automated planning software component of the Shakey robot at Stanford Research Institute by Richard Fikes and Nils Nilsson in 1971. Shakey was described as the "first electronic person" (Life Magazine, 1970[1]) and marked a significant milestone for the field of artificial intelligence in that it included breakthroughs for search algorithms as well as applications of computer vision and path optimization. The A* search algorithm (Hart, Nilsson, and Raphael 1968) was a key result of the research that went into Shakey and STRIPS and helped the robot define an efficient path in it's environment. STRIPs later became associated with the formal language that was defined for inputs to the Shakey automated planner, and is often cited as providing the seminal framework for attacking the "classical planning problem" (Hart and Nilsson, 1993[2]). Interestingly, Hart and Nilsson's results for STRIPs were made possible by their respective backgrounds in heuristic problem solving and logic based representations (Hart and Nilsson, 1993[3]).

GRAPHPLAN

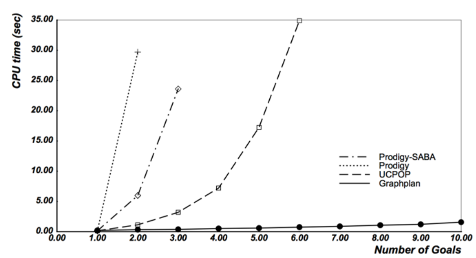
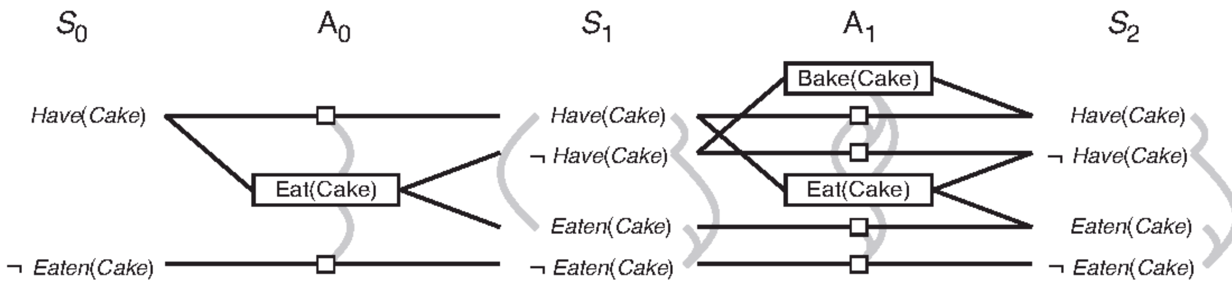


Figure 3: *2-Rockets problem*

Graphplan is an automated planning algorithm developed by Avrim Blum and Merrick Furst in 1995[4]. Graphplan utilizes the STRIPS semantic structure to state the input of the planning problem and an improved data structure called a planning graph. Prior planning algorithms relied upon a state space graph data structure of state – action connections while graph plan data structure includes connections to (preconditions) and from (effects) the action arranged in alternating levels. Organizing the data structure before embarking on the search adds an additional step to the overall planning process; however, "a Planning Graph encodes the planning problem in such a way that many useful constraints inherent in the problem become explicitly available to reduce the amount of search needed" (Blum and Furst 1995[5]). This innovation dramatically improves

[1] https://www.sri.com/work/timeline-innovation/timeline.php?timeline=computing-digital#!&innovation=shakey-the-robot
[2] https://pdfs.semanticscholar.org/076a/e14bfc68acdbaf2ab24913e152d49540e988.pdf
[3] https://pdfs.semanticscholar.org/076a/e14bfc68acdbaf2ab24913e152d49540e988.pdf
[4] https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf
[5] https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf

computing performance – in the original paper's benchmark testing of the 2-Rockets problem, Graphplan significantly outperforms the other algorithms. "Graphplan does well in this domain for two main reasons: (1) the Planning Graph only grows to 3 time steps, and (2) the mutual exclusion relations allow a small number of commitments (unloading something from Rocket1 in Paris and something else from Rocket2 in JFK) to completely force the remainder of the decisions" (Blum and Furst 1995[6]). Additionally, the data structure provides a valuable means of visually diagramming the space of the problem[7]:



HEURISTIC SEARCH PLANNER (HSP)

Although the Graphplan algorithm demonstrated significantly better results than prior methods, it did particularly well when the heuristics were known. The Heuristic Search Planner is an algorithm developed by Blai Bonet and Hector Geffner in 2000 that automatically extracting heuristics for the problem based on the STRIPS encoding inputs by applying a similar search method to the heuristics themselves. This added feature to a base Graphplan approach means that with HSP, the algorithm is generalizable to a wider range of problems, specifically those that do not have predefined or known heuristics. In comparison to approaches contemporary to publication, HSP outperforms on the Gripper problem in both Plan Length and Logistics Time[8]. HSP made the approach of state-space search to large scale problems feasible[9] *and* relieved some of the bottlenecks of forward-only approaches in that it is capable of both forward and backwards state search processes.

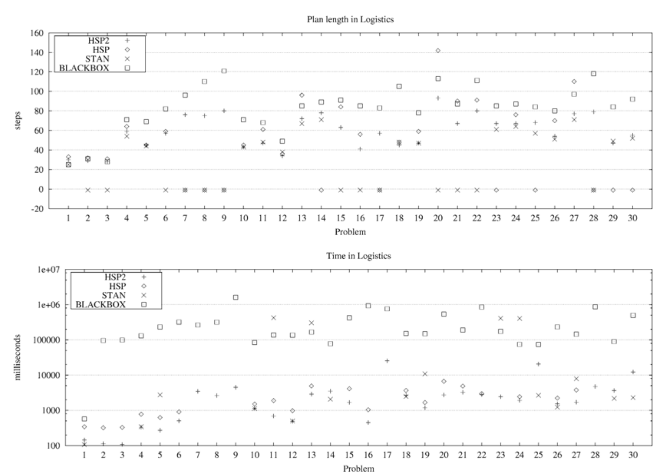


Fig. 2. Solution length (upper) and time (lower) over 30 Logistics instances from Kautz and Selman.

[6] https://www.cs.cmu.edu/~avrim/Papers/graphplan.pdf
[7] http://aima.cs.berkeley.edu/2nd-ed/newchap11.pdf
[8] http://www.cs.toronto.edu/~sheila/2542/s14/A1/bonetgeffner-heusearch-aij01.pdf
[9] http://aima.cs.berkeley.edu/2nd-ed/newchap11.pdf