

A STUDY ON ARTIFICIAL INTELLIGENCE PLANNING

Mariya Ushshaque¹, Deependra Pandey²

¹Department of Electronics & Communication, Amity School of Engineering and Technology, Amity University Lucknow Campus

²Department of Electronics & Communication, Amity School of Engineering and Technology, Amity University Lucknow Campus

Abstract

Human planning is an act of approaching goals. Its nature is explicit that is humans may proceed with prior planning and without planning. We mostly plan out things when the situation is too demanding. Artificial intelligence planning is the intelligence of planning exhibited by machines or software. It is an academic field of study which studies goal of creating intelligence to plan virtually for obtaining optimum results by problem formulation. The initial state, the set of possible actions, goal and path cost functions are main attributes of AI planning. This paper is a concise study of planning algorithms in Artificial Intelligence.

Keywords: Planning, intelligence, optimum, attributes.

1 INTRODUCTION

Graph plan algorithm came into the research field of AI Planning in 1995 by Blum and Furst. It is a different approach from the traditional methodologies of planning which worked on partial order planning. Graph plan finds method of fixed length and uses reachability information of pruning trees. Graph plan encourages researchers to look beyond the traditional AI toolbox. The realistic planning in dynamic situations try to observe and interpret planning activities here. The more realistic model is interleaved planning and execution. This covers plan superposition, plan revision and re-planning. Dynamic planning is a closed loop between planner and controller.

Multiple planning agent platforms behind some practical AI planner is the basis for SRI international SIPE planner. The open planning architecture is basis for O Plan. It handles multiple planner roles and levels efficiently namely task assigner, planner, plan specialists, plan and execution.

Description of problem

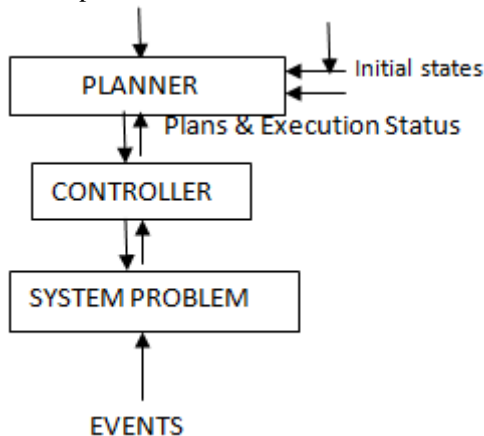


Fig.1: Problem Model

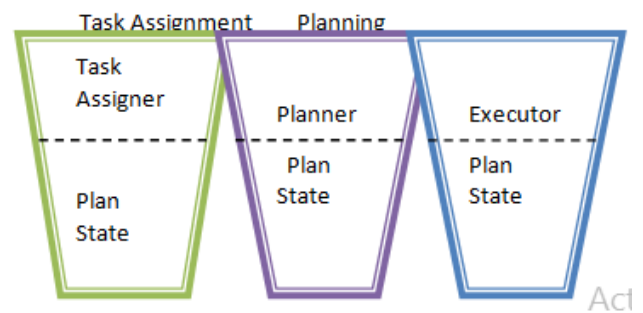


Fig.2: O Plan 3-level of agents

The constraint based model follows graph plan. Another domain of research works on various methods of exploring state space using ordered binary decisions diagrams. OBDD algorithms are efficient for classical planning but they have a limitation of memory consumption during run time.

2. SEARCH TECHNIQUE

Planning is processed by first trying to find out the plans of lengths $n=0,1,2,\dots$ till plan is found. The graphplan algorithm which uses backward chaining also known as regression initiates from the description of state of the goal. To reduce the amount of search graphplan uses information from the failed nodes and isomorphic trees are not traversed numerous times. The description of the goal and starting stage as well as conditions are translated into propositional logic and thereby plans are found using algorithms. Direct solution can also be obtained by using constraint solver.

3. CONDITIONAL AND PROBABILISTIC PLANNING

The classical planning of any problem in which there is uncertainty and incomplete information, conditional planning is required. More often general problem arises because

planning is required where data is insufficient to employ and events taking place during execution is nondeterministic. This does not allow one to choose the set of operators by observing and plan mostly assigns the execution states an operator.

4. AI PLANNING ALGORITHMS

In AI Planning, planners do not represent the transition relations associated with state space exclusively.

4.1 Forward State Space Search Algorithm

The simplest classical planning method is the space search algorithm. Here the search space is the subset of state space. Each node corresponds to a state of the world and the current plan to the current path followed. This can be forward and backward both types. Forward search signifies the planning from the initial state to the goal.

Function fwd search(O, Si, g)

State-Si

Plan<>

Loop

If state. satisfies(g) then return plan

Applicable-{ground instances from O applicable in state}

If applicables. is empty() then return failure

Action-applicables. choose one()

State-y(state, action)

Plan- plan.<action>.

4.2 Heuristic Search Algorithm

Heuristic method is applied when classical planning algorithms provide the result at a slower rate. This may not give the best result but it approximates the result. This method is faster and easier to work on.

The heuristic function h: state space – R

$h(n)$ = estimated cost of cheapest path from node n to a goal node. If n is a goal node then $h(n)$ must be zero. This function encodes problem specific knowledge in a problem independent way. This gives the correct value of goal.

4.3 Best First Search Algorithm

This algorithm works by exploring a graph by expanding the most promising node chosen according to the set rules.

$$F(n) = h(n) + g(n)$$

$g(n)$ is the cost to reach the goal node.

In the case of general search tree algorithm:

Strategy is select next node based on an evaluation. func f: state space- R

Select node with lowest value $f(n)$.

While implementing select from (fringe, strategy). Priority queue maintains fringe in ascending order of values.

4.4 Greedy Best First Search Algorithm

A heuristic search in which attempts to predict the closest path to a solution is.

It uses heuristic function as evaluation function.

$$f(n) = h(n)$$

It always expands node which is closest to goal node and eats the largest chunk out of remaining distance hence is termed as greedy. This gives results of optimisation far more longer than goal cost function is.

4.5 A* Algorithm

This algorithm always gives the best optimisation result and is applicable in every AI applications. It always underestimates the distance. $H(n) \leq$ actual distance of nearest goal node. If $h(n)$ is admissible then $f(n)$ never overestimates true cost of a solution through n.

This algorithm is optimal in nature that it guarantees to find out the shortest path of goal accurately.

Function a star tree search(prob, n)

Fringe- priority queue(new search node(problem initial state))

All nodes- hash table(fringe)

Loop

If empty(fringe) then return failure

Node- select from (fringe)

If prob goal test(node state) then

Return path to (node)

For successor in expand(problem, node)

If not all nodes contain(successor) then

Fringe- $f + \text{successor} @ f(\text{successor})$

All nodes add(successor)

5. CONCLUSION

AI Planning is a sequence of actions for transforming a given state into a state which fulfils a predefined set of goals using perception, reasoning and language understanding. It also develops and formulates algorithms, calculi of logic, graph theory and complexity theory. This is used in many real world applications like command and control, emergency responders, central authorities and isolated personnel. There is one point to be ponder over is that the real world problem is different from the model described for algorithms study.

REFERENCES

- [1]. [Bacchus and Kabanza, 2000] Fahiem Bacchus and Froduald Kabanza. Using temporal logics to express search control knowledge for planning. *Artificial Intelligence*, 116(1–2):123–191, 2000
- [2]. [Bacchus and Nau, 2001] Fahiem Bacchus and Dana Nau. The AIPS-2000 planning systems competition. *AI Magazine*, 2001. to appear.

- [3]. [Bonet and Geffner, 2000] Blai Bonet and Héctor Geffner. Planning with incomplete information as heuristic search in belief space. In Steve Chien, Subbarao Kambhampati, and Craig A. Knoblock, editors, *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems*, pages 52–61. AAAI Press, 2000.
- [4]. [Bonet and Geffner, 2001] Blai Bonet and Héctor Geffner. Planning as heuristic search. *Artificial Intelligence*, 2001. to appear.
- [5]. [Hoffmann, 2000] Jörg Hoffmann. A heuristic for domain independent planning and its use in an enforced hill-climbing algorithm. In Zbigniew W. Raś and Setsuo Ohsuga, editors, *Foundations of Intelligent Systems, 12th International Symposium, ISMIS'00*, number 1932 in Lecture Notes in Artificial Intelligence, pages 216–227. Springer-Verlag, 2000
- [6]. Lima, E. O., 2008. Hybrid Genetic Algorithm Applied to the Optimization of Functions. Bachelor Thesis. Lavras Federal University - Minas Gerais.73p
- [7]. Nagar, P. and Srivastava, S., 2008. Application of Genetic Algorithms in Data Mining. 2nd National Conference on Challenges & Opportunities in Information Technology. 4p