IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 3 Issue 3, March 2016.
www.ijiset.com
ISSN 2348 – 7968

A Theoretical Approach towards Dealing with Planning and Searching in Automated Designing Systems

R.Chandra[1], G.Prathibha Priyadarshini[2] and Dr.S.Zahoor UI Huq[3]

[1] Research Scholar, Department of CSE, Mewar University
Gangrar, Chittorgarh, Rajasthan, India

[2] Assistant Professor, Department of CSE, Brindavan Institute of Technology & Sciences,
Kurnool, Andhra Pradesh, India

[3] Professor, Department of CSE, G.Pulla Reddy Engineering College,
Kurnool, Andhra Pradesh, India

Abstract

Artificial Intelligence Planning is an art of planning exhibited by software or a machine to solve problems virtually with intelligence. This technology is used to exhibit certain sequence of actions in a system physically or virtually where is no human interaction. Problem solving in Artificial Intelligence (AI) can be done by implementing certain planning and searching algorithms which are specially designed and developed for automated designed systems. These algorithms may be characterized as a systematic search which works through a range of possible actions in order to reach certain solution or predefined goal. In AI problem solving by search algorithms for problem solving is quite common technique but will have big impact on the technologies of the robotics and path finding. This paper is a concise study of planning and searching algorithms in Artificial Intelligence.

Keywords: *Artificial Intelligence, Path Finding, Planning Algorithms, Problem Solving, Virtual, Search Algorithms, Robotics.*

1. Introduction

Most humans make planning as an act of approaching goals whose nature is explicit that they may proceed with or without prior planning. We mostly plan out things when the situation is too demanding.. But Planning is a key ability for intelligent systems, increasing their flexibility and autonomy through the construction of sequences of actions to achieve their goals. It has been a research area in artificial intelligence for over decades. Planning techniques have been applied in a variety of tasks which includes robotics, web-based information gathering, and Process planning, autonomous agents and also spacecraft mission control. Planning involves the representation of reasoning about the effects of actions, and world models, and techniques for efficiently searching the room of possible plans. The last 50 years of research witnessed the developed a large number of artificial intelligence tools for solving the most difficult problems in computer science and engineering. These tools are Probabilistic methods for uncertain reasoning, Search and optimization, Logic, Neural Networks, Control Theory and Languages, Classifiers, Statistical Learning Methods. Many problems in artificial intelligence can be solved theoretically by intelligently searching and many possible solutions [3]. Reasoning has been shown to be reduced up to performing a search. For instance, logical proof can be viewed as searching for a path that leads from near premises to conclusions, where each step is an inference rule of the application[1]. Planning algorithms search through trees of root goals and sub goals attempting to find a path to target goal which is a process called “Means-End Analysis(MES)”. Algorithms on robotics for moving limbs and grasping objects make use of local searches in configuration spaces [2]. Many learning algorithms use optimized search algorithms. Simple Exhaustive searches [4] are sufficient rarely for most real world problems for most real world problems. The search space quickly grows to astronomical numbers. The result is a search which may be too slow or never completes. The solution, for many these problems, is to use rules of thumb or heuristics that eliminate choices that are unlikely to lead to the goal. Heuristics supply the program with most relevant guess for what the solution lies on [3-4]. A very different kind of search came to prominence in 1990s, based on the mathematical theory of optimization. For many such search problems it is possible to begin with some form of a guess and then refine the guess process incrementally until no more refinements can be done further. These algorithms can be visualized as blind hill climbing. We begin our search most probably at a random point on the landscape and then, by steps or jumps, we keep moving our guess uphill, until we reach the top.

ISSN 2348 – 7968

1.1 Research Scope

An AI planning system is charged with generating a plan that is one possible solution to a specified problem. The plan generated will be composed of operator schemata, provided to the system for each domain of application. This study will elaborate the meaning of each of these terms and how they relate to one another.

An initial state and a goal state description problem is characterized. The initial stated description tells the planning system the way the world is right now. The goal state description tells the planning system the way we want the world to look when the plan has been executed. The planning which takes place in the world is often called the application domain. Sometimes we refer to the goal state description as the goal. In many systems, a goal can be transformed into a set of other, usually simpler, goals called sub goals.

This study will cover search techniques for artificial intelligence planning. In today's time the Agents operating in the real world often have to act under the conditions where time is critical and there is a limit on the time they can afford on spending what action to execute next. Planners must produce the best plans they can find within the stipulated amount of time available.

The algorithms presented in this study will all suited better to some domains. They can also benefit from certain optimizations in some cases. The main purpose of this study is to give some intuition as to when the presented algorithms might be useful and when they might not. The strategy searching always for an optimal plan becomes infeasible in these scenarios. Otherwise, we must use an anytime planner, anytime planners operate by quickly finding a highly suboptimal plan first, and then improving it until the available time runs out.

2. Preliminary Study

There are many search and optimization algorithms available in Artificial Intelligence, among which the most popular ones are being Uninformed Search, Heuristic Search and Evolutionary algorithms. As there is a lot of research work has done on individual algorithms but not enough research is done on the comparison of these algorithms under different problems like performance, complexities and limitations. According to the sources the fact that these algorithms behave differently and perform differently for different problems. By analyzing these algorithms performance under a certain problem, the out comings can be found out and more research work could be done do overcome issues if any. For further assistance, this research work helps researchers in choosing the best suited algorithm to a particular problem by determining the pros and cons of the algorithm after testing.

Search algorithms most probably used for a multitude of artificial intelligence tasks in which one of them is being the path finding. The area of search in artificial intelligence is very much connected to solutions for real life problem. Artificial Intelligence has investigated search methods which allow solving path planning problems in large domains. For having formulated problems, we need to follow them by searching through the state space during this process for a solution. Most of the researches have studied how to solve one-shot path-planning problems in problem solving. Search process is mostly a repetitive process and therefore, many artificial intelligence systems re-plan independently from scratch to solve the path planning problem. Search Optimization is one of the most important tasks we have to carry out. They are required new, better, more effective, less complex and less expensive design to improve operations of existing systems in both scientific and industrial world.

3. Planning and Search

From the previous research, planning is essentially a search problem. The program must traverse a potentially large space and find an appropriate plan which is applicable in the initial state and produces a solution to the goal. Such search can be some how difficult because they can contain many interactions between states. These interactions lead to a surprising amount of complexity (Chapman 1987), and the problem of finding a plan which is optimal in even a simple blocks world domain has recently proved in NP-hard (Gupta 1990). Planning which involves conditional effects has been shown to be unpredictable (Chapman 1987). A plan generated in such domains cannot be guaranteed to succeed. Therefore, the problem of organizing the heuristic search, choosing what to be done in cases of failures, and generally finding the ways to make more informed choices among the planning literature.

Early research approaches to planning apply legal moves to some initial states for a state that satisfied the identified goals. There was an overlap with game-playing work. Heuristic evaluation functions were assigned to rate the various intermediate levels of states produced to estimate their accuracy to a goal state [Hart, Nilsson, and Raphael 1968] and the graph traverse [Doran and Michie 1966]. This approach, however, was found difficult in designing work heuristics and identified usual exponential growth of the search space.

To reduce the number of middle-level states considered, Newell and Simon in 1963 introduced *means-ends analysis*, a heuristic that involves only activities that could satisfy outstanding goals.

ISSN 2348 – 7968

Operators were preferred that would cause the present state to more closely resemble the goal state when executed. This technique was used as the basis of the search. With the Noah introduction of procedural nets (Sacerdoti 1975), the search problem was completely changed. In this system and their descendants, the search space does not consist of a set of world states but of a space of partial plans also. For any non-primitive action in the existing network, the planner can consider any method which is known to reduce the action to a set of other actions. In such systems, planning consists of choosing appropriate reductions from among the possibility sets and ordering actions to eliminate interaction which does not lead to solutions. One important technique introduced for searching partial plans is least commitment plan representations.

Such representations are used and allow a set of plans to be represented in a single state of the search. Such representations include the use of a plan ordered partially to represent number of possible action prior orderings to a commitment becoming necessary, as in Noah, or the posting of constraints on objects referred to in the plan rather than the making of an arbitrary selection, as in Molgen 1981. To search the space of partially ordered plans, many solutions have been proposed. Some systems do not search through the possible alternatives at all. Instead, selections are made on the basis of locally available information, and a commitment is made to a particular solution path.

This technique has been proved as most successful where strong domain heuristics can be used for making choices. More general solutions are available where such heuristics use backtracking to allow backing up to occur when the goal cannot be reached based on some choices made earlier. The planning system simply saves the state of the solution at each point at which there is many alternative ways to continue and keep records of choices. The first is chosen, and the search continues. If there is any failure, the saved state can be restored from the last choice point, and the next alternative is chosen. Implementation of Simple stack-based technique can be used for this process. Because good local information is often available to indicate the preferred solution path, it is often appropriate to try the best choice indicated by local heuristic information before considering the many alternatives that might be available should the local choice prove faulty. From the extreme point of view, depth-first search gives search strategy something of the flavor. However, gradual wandering from a valid solution path could identify backtracking when a failure is detected through many levels. An alternative is to focus on the choice currently made and selected the local choice that seems most promising. This process continues while all is going well. However, if a failure occurs, Nonlin considers the entire set of alternatives that were generated. This basic technique was refined further to provide the most widely used approach to controlling search in planning. Any backtracking system based on saved states and resumption points can waste much valuable search effort leads to a solution that can have several unrelated parts. If backtracking on one part has to go back beyond at which point of work was done on an unrelated part, then all the efforts on the unrelated part will be lost. Many planning systems avoid this problem by using a variant of the backtracking methods used in Nonlin. These systems do not keep choice points of the solutions saved. Instead, they record the dependencies between decisions, the assumptions on which they are based, and the alternatives can be made from a selection. They then follow methods undone a failure by propagating and recovering all the dependent parts of the solution. This process leaves unrelated parts intact irrespective of whether they worked on some part of the solution which are undone. Examples of other research works use techniques proposed by Hayes (1975); Stallman and Sussman (1977); Daniel (1983) and Stefik (1981a, 1981b).

4. Literature Review on Search Algorithms

In this section, we studied some of the search algorithms. Symmetry: Symmetry detection is a way to reduce search space [5]. It finds symmetric objects (DTGs in SAS+ formalism) and actions that are indistinguishable with respect to initial state and goal. However, this method proposed by Fox and Long [5] can only detect symmetry from the specification of initial and goal states, and may miss many symmetries.

Factored planning: Factored planning [6-8] is a class of search algorithm that exploits the decomposition of state space. Factored planning finds all the sub plans for each individual sub graph and tries to merge them. There are some limitations of factored planning. The most notable is that search becomes prohibitively expensive when there are many sub plans in each sub graph, and not every sub graph has goal states. Although factored planning has shown potential on some domain-dependent studies, its practicality for general domain-independent planning has not been established yet.

Partial order reduction: Partial order reduction (POR) is a way to reduce the search cost for classical planning [9-10]. It allows a search to explore only part of the entire search space while still maintaining completeness and/or optimality. The idea is to enforce partial orders between states at the time of search. POR algorithms have been studied extensively for model checking [11-12], which also requires examining a state space in order to prove certain

ISSN 2348 – 7968

properties. Model checking is not practical without POR due to its time complexity [13-18].

Stochastic search: An alternative to deterministic search is stochastic search. One representative stochastic search algorithm is called the "Monte-Carlo Random Walk algorithm (MCRW)". Arandom walk in state space is a trajectory of states that are linked by random actions. An MCRW starts from a known state, usually the initial state, by applying random actions to known states, generates a random walk in the search space, and terminates when a goal state is found in the walk. Stochastic search has been used by some of the leading planners in the Seventh International Planning Competition [19-21].

5. Analysis of Planning Approaches

Planning is an area of great present interest within artificial intelligence. The main reason is that it combines the two major areas of Artificial Intelligence we have studied are *search* and *logic*. That means, a planner can be seen either as a program that searches for a solution or as that proves the solution existence. The interference of ideas of those areas has led to both performance amounting in improvements to several magnitude orders in the last decade and an increased use of planners in industrial applications also. But unfortunately, we do not yet identified a clear understanding of techniques that work best on different kinds of problems. New techniques quite possibly will emerge that dominate existing methods. Planning is foremost a practice in controlling combinatorial explosions. If there are P domain primitive propositions, then there are 2P states. For complex domains, p it can increase to a large extent. Consider the domain have properties like Location, Color, etc. and relations At, On, Between, etc. With D domain objects with ternary relations, we have 2d3 states. In the worst case we might conclude that, the planning is hopeless. In this case the divide-and-conquer approach can be a most powerful weapon.

In the best case full decomposability of the problem using divide-and-conquer offers exponential speedup. Decomposability is destroyed, by negative action interactions between them. Partial order planners would deal with causal links, but unexpectedly each conflict must be resolved with a choice and the choices can multiply exponentially. GRAPHPLAN can avoid such choices using mutex links to record conflicts without actually making a choice to resolve them. SATPLAN represents a similar range of mutex relations, but by using the general CNF form and how well this works depends on the SAT solver used.

In some cases it can be possible to solve a problem efficiently by identifying negative SERIALIZABLE interactions which can be later ruled out. We can also say that a problem has sub goals which are serialize. **I**f there exist SUBGOALS then the planner can achieve them in that order, without having to recover any of the subgoals achieved previously.

In more complex examples like Remote Agent planner of NASA's Deep Space One spacecraft, it was determined that the propositions involved in serializable spacecraft commanding. Perhaps, this is not too surprising, because an engineer designed spacecraft as easy as possible to control. Taking this as an advantage of the serialized ordering of goals, the Remote Agent planner was able to eliminate search mostly. This mean that it was enough fast to control the real time spacecraft which was previously considered impossible. There are many ways to control combinatorial explosions. We mean time studied that there are many techniques for controlling backtracking in constraint satisfaction problems. All such techniques can be applied to planning.

6. Future Study

Certain problems are characterized by an initial state and its goal state description. The initial state describes the planning system and the way how the world is right now. The goal state description tells the planning system the way we want the world to look when it has been executed. The world in which planning takes place is often called the application domain and we sometimes refer to the goal state description as simply the goal. Artificial Intelligence planning has bright future in the area of technology. In future, we are analysis the planning and searching algorithms based on certain applications and planning to determine how far it will be mostly used in expert systems.

7. Conclusions

The planning and searching problems in Artificial Intelligence is about the decision making performed by intelligent creatures like human, computer programs or robots to achieve some goal. It involves selecting a sequence of actions that will transform the state of the world in a step by step fashion so that it will satisfy the goal. The world is typically viewed to be consisted of atomic *facts*, and makes some facts true and some false. In this paper, we studied a number of ways to formalize planning, and show how planning and searching problems can be solved automatically using algorithms.

References

ISSN 2348 – 7968

[1] D. Poole, A. Mackworth and R. Goebel, Randy, Computational Intelligence: A logical Approach New York: Oxford University Press, 1998.
[2] S. J. Russell, P. Norving, Artificial Intelligence; aModern Approach (2ndEd). Upper Saddle River, New Jersey: Prentice Hall, 2003.
[3] G. Luger, and W. Stubble Field, Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5thed). The Benjamin/Cummings Publishing Company, Inc, 2004.
[4] N. Nilsson, Artificial Intelligence: A New Synthesis. Morgan Kaufmann Publishers, 1998.
[5] M. Fox and D. Long. The detection and exploitation of symmetry in planning problems. InProc. IJCAI, 1999.
[6] E. Amir and B. Engelhardt. Factored planning. InProc. IJCAI,2003.
[7] R. Brafman and C. Domshlak. Factored planning: how, when, and when not. In Proc. AAAI, 2006.
[8] E. Kelareva, O. Bu↵et, J. Huang, and S. Thi´ebaux. Factored planning using decomposition trees. InProc. IJCAI,2007.
[9] Y. Chen, Y. Xu, and G. Yao. Stratified planning. InProc. IJCAI, 2009.
[10] Y. Chen and G. Yao. Completeness and optimality preserving reduction for planning. InProc. IJCAI, pages1659–1664, 2009.
[11] K. Varpaaniemi. On stubborn sets in the verification of linear time temporal properties. Formal Methods in System Design,26(1):45–67,2005.
[12] E. M. Clarke, O. Grumberg, and D. A. Peled.Model Checking.TheMITPress, 2000.
[13] P. Godefroid. Using partial orders to improve automatic verification methods. In Proc. of International Workshop on Computer Aided Verification, London, UK, 1990.
[14] P. Godefroid and D. Pirottin. Refining dependencies improves partial-order verification methods. InProc. of Computer Aided Verification,pages438–449,1993.
[15] P. Wolper and P. Godefroid. Partial- order methods for temp oral verification. In Proceedings of the 4th International Conference on Concurrency Theory, pages 233–246, 1993.
[16] R. Gerth, R. Kuiper, D. Peled, and W. Penczek. A partial order approach to branching time logic model checking. InProc. of ISTCS,1995.
[17] D. Peled. Partial order reduction: linear and branching temporal logics and process algebras. InProceedings of the DIMACS workshop on Partial order methods in verification, pages233–257, and 1997.
[18] G. J. Holzmann. The model checkers SPIN.IEEE Trans. on Software Engineering, 23:279–295, 1997.
[19] H. Nakhost and M. M¨uller. Monte-carlo exploration for deterministic planning. InProc. IJCAI, pages1766–1771, 2009.
[20] Q. Lu, Y. Xu, R. Huang, and Y. Chen. Roamer planner random-walk assisted best-first search. InThe Seventh International Planning Competition, 2011.
[21] Y. Xu, Q. Lu, R. Huang, and Y. Chen. The roamer-p planner. In The Seventh International Planning Competition, 2011.
[22] Mariya Ushshaque1, Deependra Pandey – " A Study on Artificial Intelligence Planning", IJRET: International Journal of Research in Engineering and Technology, Volume: 04 Special Issue: 10 | COTII-2015 | Sep-2015, pp.no 36-39.
[23] *James Hendler, Austin Tate, and Mark Drummond* – "AI Planning: Systems and Techniques", AI Magazine Volume 11 Number 2 (1990), pp.no 61-71.
[24] Ashwani Chandel, Manu Sood – " Searching and Optimization Techniques in Artificial Intelligence: A Comparative Study & Complexity Analysis", *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 3, March 2014, pp.no 867 – 879.*