Proceedings of
National Conference on Innovative Trends in Computer Science Engineering (ITCSE-2015)
held at BRCMCET, Bahal on 4th April 2015

COMPARATIVE STUDY OF VARIOUS SEARCHING ALGORITHMS

Deepika Garg
Department of Computer Science Engineering,
The Technological Institute of Textile & Sciences Bhiwani, India
deepugarg22@gmail.com

ABSTRACT-*Artificial intelligence (AI) is the study of how to make computers do things which, at the moment, people do better. Thus Strong AI claims that in near future we will be surrounded by such kinds of machine which can completely works like human being and machine could have human level intelligence. One intention of this article is to excite a broader AI audience about abstract algorithmic information theory concepts, and conversely to inform theorists about exciting applications to AI. The science of Artificial Intelligence (AI) might be defined as the construction of intelligent systems and their analysis. . We investigate the use of informed search algorithms and uninformed search algorithms for intelligent travel planning.In the last we introduce the field of AI and review the use of fuzzy logic.*

KEYWORDS: Algorithms, Artificial Intelligence, Fuzzy logic.

I. INTRODUCTION

Artificial intelligence (AI) is the intelligence of machines and the branch of computer science that aims to create it. AI textbooks define the field as "the study and design of intelligent agents" where an intelligent agent is a system that perceives its environment and takes actions that maximize its chances of success. John McCarthy, who coined the term in 1955, defines it as "the science and engineering of making intelligent machines." AI research is highly technical and specialized, deeply divided into subfields that often fail to communicate with each other. Some of the division is due to social and cultural factors: subfields have grown up around particular institutions and the work of individual researchers. AI research is also divided by several technical issues. There are subfields which are focused on the solution of specific problems, on one of several possible approaches, on the use of widely differing tools and towards the accomplishment of particular applications. The central problems of AI include such traits as reasoning, knowledge, planning, learning, communication, perception and the ability to move and manipulate objects. General intelligence (or" strong AI") is still among the field's long term goals. Currently popular approaches include statistical methods, computational intelligence and traditional symbolic AI. There are an enormous number of tools used in AI, including versions of search and mathematical optimization, logic, methods based on probability and economics, and many others. The field was founded on the claim that a central property of humans, intelligence—the sapience of Homo sapiens—can be so precisely described that it can be simulated by a machine. This raises philosophical issues about the nature of the mind and the ethics of creating artificial beings, issues which have been addressed by myth, fiction and philosophy since antiquity. Artificial intelligence has been the subject of optimism, but has also suffered setbacks and, today, has become an essential part of the technology industry, providing the heavy lifting for many of the most difficult problems in computer science.

1.1. Applications of AI

1.1.1. R&D Plan for Army Applications of AI/Robotics:

- Robotic Reconnaissance Vehicle with Terrain Analysis,
- Automated Ammunition Supply Point (ASP),
- Intelligent Integrated Vehicle Electronics,
- AI-Based Maintenance Tutor,
- AI-Based Medical System Development.

1.1.2. Expert system

An expert system, is an interactive computer-based decision tool that use both facts and heuristics to solve difficult decision making problems based on knowledge acquired from an expert. An expert system compared with traditional computer:

Inference engine + Knowledge = expert system

(Algorithm + data structures = program in traditional computer)

1.1.3. Fuzzy Logic

Fuzzy logic is more than thirty years old and has a long-lasting misunderstanding with artificial Intelligence, although the formalization of some forms of commonsense reasoning has motivated the development of fuzzy logic.

1.1.4. Mobile Robotics and Games (Path Planning)

Mobile robots often have to re plan quickly as the world or their knowledge of it changes. Examples include both physical robots and computer-controlled robots (or, more generally, computer-controlled characters) in computer games. Efficient replanting is especially important for computer games since they often simulate a large number of characters and their other software components, such as the graphics generation, already place a high demand on the processor. In the following, we discuss two cases where the knowledge of a robot changes because its sensors acquire more information about the initially unknown terrain as it moves around.

Uninformed Search

Uninformed search (blind search) has no information about the number of steps or the path costs from the current state to the goal. They can only distinguish a goal state from a non-goal state. There is no bias to go towards the desired goal.

II.SEARCHING TYPES

2.1 Breadth-First Search

In breadth-first search the frontier is implemented as a FIFO (first-in, first-out) queue. Thus, the path that is selected from the frontier is the one that was added earliest. This approach implies that the paths from the start node are generated in order of the number of arcs in the path. One of the paths with the fewest arcs is selected at each stage.

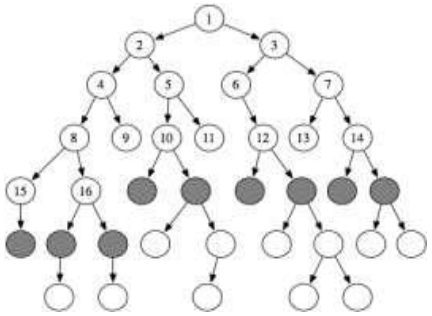

Fig 1 Breadth-first search

Breadth-first search is useful when

- Space is not a problem
- You want to find the solution containing the fewest arcs;
- Few solutions may exist, and at least one has a short path length; and
- Infinite paths may exist, because it explores all of the search space, even with infinite paths.

It is a poor method when all solutions have a long path length or there is some heuristic knowledge available. It is not used very often because of its space complexity.

2.2 Depth-First Search

In depth-first search, the frontier acts like a last-in first out stack. The elements are added to the stack one at a time. The one selected and taken off the frontier at any time is the last element that was added. This Algorithm is also called as Recursive Algorithm.

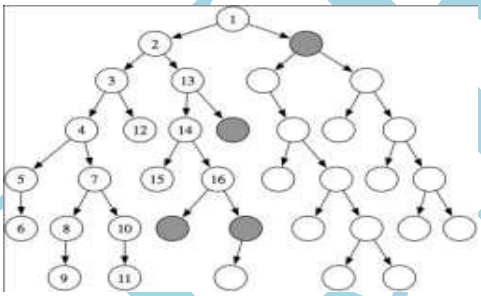

Fig. 2The order nodes are expanded in depth-first search.

Depth-first search is appropriate when either

- Space is restricted;
- Many solutions exist, perhaps with long path lengths, particularly for the case where nearly all paths lead to a solution; or
- The order of the neighbors of a node are added to the stack can be tuned so that solutions are found on the first try.
- It is a poor method when it is possible to get caught in infinite paths; this occurs when the graph is infinite or when there are cycles in the graph; or
- Solutions exist at shallow depth, because in this case the search may look at many long paths before finding the short solutions.

2.3 Informed Search

In informed search, a heuristic is used as a guide that leads to better overall performance in getting to the goal state. Instead of exploring the search tree blindly, one node at a time, the nodes that we could go to are ordered according to some evaluation function h(n) that determines which node is probably the "best" to go to next. This node is then expanded and the process is repeated (i.e. Best First Search). A* Search is a form of BestFS. In order to direct the search towards the goal, the evaluation function must include some estimate of the cost to reach the closest goal state from a given state. This can be based on knowledge about the problem domain, the description of the current node, the search cost up to the current node BestFS optimizes DFS by expanding the most promising node first. Efficient selection of the current best candidate is realized by apriority queue.

III. COMPARISON OF SEARCH ALGORITHMS

3.1 A* search algorithm

The A* algorithm combines features of uniform-cost search and pure heuristic search to efficiently compute optimal solutions. A* algorithm is a best-first search algorithm in which the cost associated with a node is f(n)=g(n)+h(n), where g(n) is the cost of the path from the initial state to node n and h(n) is the heuristic estimate or the cost or a path from node n to a goal. Thus, f(n) estimates the lowest total cost of any solution path going through node n. At each point a node with lowest f value is chosen for expansion. Ties among nodes of equal f value should be broken in favour of nodes with lower h values. The algorithm terminates when a goal is chosen for expansion.A* algorithm guides an optimal path to a foal if the heuristic function h(n) is admissible, meaning it never overestimates actual cost. For example, since airline distance never overestimates actual highway distance, and Manhattan distance never overestimates actual moves in the gliding tile. A* uses a best-first search and finds a least-cost path from a given initial node to one goal node (out of one or more possible goals). As A* traverses the graph, it follows a path of the lowest known heuristic cost,keeping a sorted priority queue of alternate path segments along the way.I t uses a distance-plus-cost heuristic function (usually denoted) to determine the order in which the search visits nodes in the tree. The distance-plus-cost heuristic is a sum of two functions :the path-cost function, which is the cost from the starting node to the current node (usually denoted)an admissible "heuristic estimate" of the distance to the goal (usually denoted).The part of the function must be an admissible heuristic; that is, it must not overestimate the distance to the goal. Thus, for an application like routing, might represent the straight-line distance to the goal,since that is physically the smallest possible distance between any two points or nodes.

3.2 AO* Algorithm

1. Let G consists only to the node representing the initial state call this node INTT.Compute h' (INIT).
2. Until INIT is labeled SOLVED or hi (INIT) becomes greater than FUTILITY, repeat the

following procedure.
(I) Trace the marked arcs from INIT and select an unbounded node NODE.
(II) Generate the successors of NODE. if there are no successors then assign FUTILITY as
h' (NODE). This means that NODE is not solvable. If there are successors then for each one
called SUCCESSOR, that is not also an ancester of NODE do the following
(a) add SUCCESSOR to graph G
(b) if successor is not a terminal node, mark it solved and assign zero to its h ' value.
(c) If successor is not a terminal node, compute it h' value.
(III) propagate the newly discovered information up the graph by doing the following . letS be a
set of nodes that have been marked SOLVED. Initialize S to NODE. Until S is empty repeat the following procedure;
(a) select a node from S call if CURRENT and remove it from S.
(b) compute h' of each of the arcs emerging from CURRENT , Assign minimum h' to CURRENT.
(c) Mark the minimum cost path a s the best out of CURRENT.
(d) Mark CURRENT SOLVED if all of the nodes connected to it through the new marked are have been labeled SOLVED.
(e) If CURRENT has been marked SOLVED or its h' has just changed, its new status must be propagate backwards up the graph. Hence all the ancestors of CURRENT are added to S.

3.3 AO* Search Procedure
1. Place the start node on open.
2. Using the search tree, compute the most promising solution tree TP .
3. Select node n that is both on open and a part of tp, remove n from open and place it no closed.
4. If n is a goal node, label n as solved. If the start node is solved, exit with success where tp is the solution tree,remove all nodes from open with a solved ancestor.
5. If n is not solvable node, label n as unsolvable. If the start node is labeled as unsolvable, exit with failure.Remove all nodes from open ,with unsolvable ancestors.
6. Otherwise, expand node n generating all of its successor compute the cost of for each newly generated node and place all such nodes on open.
7. Go back to step (2)

3.4 Properties of AO*:
• AO* is a generalization of A* for AND-OR graphs.
• AO*, like A*, is admissible if the heuristic function isadmissible and the usual assumptions (finite branching factor etc) hold.
• AO*, like A* is also optimal among the class of heuristicsearch algorithms that use an additive cost /evaluation function. AO* will always find minimum cost solution.

3.5 Use of Fuzzy Logic in Artificial Intelligence
Fuzzy logic is a superset of conventional logic that has been extended to handle the concept of partial-truth values between the boolean dichotomy of true and false. Fuzzy logic usually takes the form of a fuzzy reasoning system and its components are fuzzy variables, fuzzy rules and a fuzzy inference engine. Using fuzzy set theory, variables are fuzzified by selecting a set of membership functions on their possible range of values. The membership functions map the crisp value of a variable to a linguistic label with a degree of truth, usually in the range [0,1]. For example, the range of a weapon in a game can be divided into melee, ranged or out-of-range, as seen in Figure 1 (Top). Variables can be considered to be input, such as measures from sensors (even virtual sensors, as in games), or they can be considered to be output. Fuzzy rules can be created using the defined fuzzy variables and their sets. Those rules are usually in the form if-then, with a grammar similar to boolean logic. Rules determine the value of output variables given the current value of input variables. Operators such as AND, OR and NOT are also given a meaning in fuzzy logic. Once the rules have been created, a fuzzy inference engine is used to
infer conclusions from the current values of the variables and the rules that govern the system. Output variables are thus given a fuzzy value. Eventually, a procedure of defuzzification can be performed, transforming the output of the fuzzy engine to a crisp value.

IV.CONCLUSION

We conclude that by using this algorithm we can solve AI problems easily.AI algorithms are also
called as a problem solving algorithms. Artificial Intelligence will surpass human intelligence. Although it has proven itself to be similar to the human brain, computers do not think in the same way. In this report, we have discussed Algorithms of AI and their impact on problem solving and our lives. Fuzzy logic brings many benefits to the modeling of intelligent game agents. Its main benefit is the simplicity of its formulation, which paired with its input-output nature allows developers to integrate it into many games with ease, a huge benefit if we consider the tight schedules of game developers. Even in its simplicity, fuzzy logic can be a powerful AI technique, especially due to the possibilities to model nonlinearities, to achieve complex behavior with simple rules, to be a candidate for learning and to mimic human reasoning and emotions.

REFERENCES

[1] http://www.synergy.ac.in/intranet/e-book/(Ebook%20-%20Paper)%20Artificial%20Intelligence%20Search%20Algorithms.pdf
[2] Robin ,"A star algorithm"December 18th, 2009 .
[3] Didier Dubois-Henri Prade," The place of Fuzzy Logic in AI".
[4] Philippa Avery and Zbigniew Michalewicz. Adapting to human game play. In Proceedings of the 4th international conference on Computational Intelligence and Games, CIG'08, 2008.
[5] L. Cardamone, D. Loiacono, and P. L. Lanzi. Overtaking opponents with blocking strategies using fuzzy logic. In Proceedings of the 6th international conference on Computational Intelligence and Games, CIG'10, 2010.