International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET22962

Autonomous Intelligence in Problem-Solving by Searching in the field of Artificial Intelligence

Anup Kumar Biswas
Assistant Professor, Computer Science and Engineering, Kalyani Govt. Engineering College, Kalyani, Nadia, West Bengal, India

Article Info
Volume 9, Issue 6
Page Number : 53-62

Publication Issue :
November-December-2022

Article History
Accepted : 02 Nov 2022
Published: 15 Nov 2022

ABSTRACT

Problem-solving strategies in Artificial Intelligence are steps to overcome the barriers to achieve a goal, the "problem-solving cycle". Most common steps in such cycle involve— recognizing a problem, defining it, developing a strategy in order to fix it, organizing knowledge and resources available, monitoring progress, and evaluating the effectiveness of the solution. Once a solution is obtained, another problem usually comes, and the cycle starts again. Searching techniques in problem-solving by using artificial intelligence (A.I) are surveyed in this paper. An overview of definitions, development and dimensions of A.I in the light of search for solutions to problems are accepted. Dimensions as well as relevance of search in A.I research is reviewed. A classification of searching in the matter of depth of known parameters of search was also investigated. At last, the forethoughts of searching in AI research are brought into prominence. Search is noticed that it is the common thread that binds most of the problem-solving strategies in AI together.

Keywords - Problem-Solving, Search, Autonomous Intelligence, Search Techniques, Bottle-Necks

I. INTRODUCTION

AI forethought is considered to be a system which thinks like humans do: - This is the new effort to make computers think —machines with minds, in full and literal sense. It is the automation regarding activities with which we associate thinking of human beings, activities such as problem solving, learning, decision-making, etc. [6].

- It is a system that functions like a human: - This is such an act that creates machines which perform functions requiring intelligence when performed by human. In addition, it is the lesson or practice of how to make the computers do things at the moment when people are better [7].
- AI is a system that thinks of rationally: - the study of mental abilities, by the use models of computations [8]. Moreover, it is the study of computations that causes it possible to reason, perceive, and act [4].
- It is a system that acts rationally: - in this case, computational intelligence is being the study of the designing of intelligent agents [5], [10]. Also,

it is connected with the intelligent characteristic in artefacts not in natural objects.

Artificial intelligence (AI) was first invented by JohnMcCarthy in the year 1956 [11]. This field was established upon the claim that an internal property of humans, that is intelligence – the expertness of Homo sapiens – can be so exactly discussed that it can be simulated by a machine. This draws philosophical issues regarding the mind nature and ethics or principle of generating artificial beings, issues that are addressed by fiction, myth and philosophy from the antiquity to the great civilizations of antipathy [6]. AI has become the subject of optimism but has also suffered from hazards and, at present, has become an essential part of the technological industry, yielding the heavy lifting for most difficult problems in computer science & engineering, particularly by applying search techniques. The kernel problems of AI include such criteria like: Reasoning, Planning, Knowledge, Learning, Perception, Communication, Ability to move, and Manipulate objects [10].

II. AI and SEARCH

For the intention of solving the problems in computer Science and related Engineering sections for the last 5 decades, the researchers have developed a great number of Artificial intelligence tools which are as follows: Search and optimization, Probabilistic methods for uncertain reasoning, Logic, Control theory, neural networks, Classifiers with statistical learning methods and Languages.

In AI, different types of problems are solved theoretically and intelligently by searching through different probable solutions.

Scientists and Researchers have investigated and observed that Reasoning reduces the prices for performing a search. As an example, we can find that a logical proof may be viewed as searching for a communication a path from its premises to the destinations/conclusions in which every step is considered as an application of a "rule of inference [5]." Searching, using planning algorithm, through trees to get target and sub targets (goals) for the purpose of finding out of a path connecting the initial point to target goal point. This is a process defined as "means-end-analysis [5]." For grasping objects and moving limbs, robotic algorithms do the use of local searches in the case of configuration space [10]. Different learning algorithms make use of search algorithms on the basis of optimization.

For the most real world problem cases, simple exhaustive searching process are not sufficient. The total number of points to be searched, called search space, gives rise to an astronomical numbers. As a result the search is to slow to reach the destination or never to complete the searching. The solution we can expect for different problems is using the "thumb rule-Heuristics" which reduces the choices and are not likely to show the destination (defined as pruning the search tree). Heuristic, of course, provide the program that has "best guess" for which the best solution relies upon [10-12]. A peculiar type of search came in the 1990s, which is really based upon the "mathematical theory of optimization". Possibly, we can start the search having some form of guess and it can refine the guess step by step until no extra refinement can be made up of. We can visualize these algorithms as the blind hill climbing used for finding out the peak(s). We can start searching point randomly on the landscape and next by jumping the steps, we stay moving uphill till one can reach the peak. There are some optimization algorithms as: beam search, simulated annealing, and random optimization [10-12].

III. DEFINITION & SEARCH IN AI

a. Definition of Search in AI

The key concept of an AI lies on its definition. It is intrinsic to AI problem solving techniques. Really, AI problems are intrinsically complex to solve. When we are attempting to solve the problems that human beings start to handle by applying inborn cognitive abilities, pattern recognition and so on by using computers, this attempting to solve the problems must turn to a consideration of search [11].

IV. Classification of Search Techniques

Different search techniques may be thought of belonging to any of two categories of searching techniques like:

a) Exhaustive (blind) method
b) Heuristic or informed method (given in section IX)

a) Exhaustive (Blind) Search

Exhaustive search or Blind search of a search space is not really workable or it is impractical for the cause of the problem space size being considered. For some examples, someone can define a set of transformations for a state space (moving in the game world).

In some instances, it is however necessary, more often, we are able to define a set of legal transformations of a state space (moves in the world of games) that are likely to bring us next to a desired destination or goal state selected; whereas others are not explored again. This technique we consider in problem solving is called SPLIT and PRUNE. In AI, the technique implementing "split and prune" is said to be Generate and teat. The primary procedure we can accept is as follows:

Repeat:

- first create a candidate solution
- To test or verify the candidate solution till a desired solution we have found out or no more candidate solution is produced or created.
- Whenever a desired solution is found, we must announce it, else, we declare failure.

A fine candidate solution generator would produce all the probable solutions and will not predict redundant solutions. These solutions are also informed, i.e., they limit the solutions which they pursue or propose.

V. Means-End-Analysis

Means-End-Analysis is also a state search method which is apart from the generate- and test-technique(s). The aim of it is to minimize the differences between present and goal state, provided an initial state is given. Measuring the difference (or distance) between (i) any one state and (ii) goal state can be simplified with the different procedure matching tables that prescribe successively what the next state would be. The action to perform the Means-End-Analysis[2] will be like below:

Repeat

- To indicate the present state, the goal state and the difference between them.
- To use the difference (i.e., distance) between the two states (i) present state and (ii) goal state, perhaps keeping descriptions of the present- and goal-state to choose a promising procedure.
- Make use of the promising procedure and bring up to date the current state, till the "goal point" is obtained or not any procedure is available.
- When the "goal" is achieved, announce success; else, announce failure.

VI. Depth First Search (DFS)

Among the uninformed or blind or exhaustive search algorithms [1], DFS is basic and fundamental. This search facilitates the probing down a strong solution path with the hope that solutions will not lie down the tree. So DFS is a fine concept for us to be confident that all fractional paths whether reaching dead ends or becoming complete paths after legitimate of steps. In the bad sense, DFS keeps in a bad idea when there is long and long paths which neither reach dead nor complete paths [4].

When we want to apply DFS, we start it with the root node. When applying this type of searching procedure, we commence with the starting node or root node and entirely look for the descendants of a node as we are going to explore siblings of it in the fashion of left-to-right [14].

DFS Procedures:-

a) Place the starting node on the queue called OPEN
b) If OPEN is empty, exit with failure and stop; otherwise continue
c) Take aside the first node out of OPEN. Then put it on a list marked as CLOSED. Say it is node n.
d) When the depth of n becomes equal to the depth bound, go to (b); otherwise continue.
e) Spared the node n, creating each successors of n, arrange them (in any order) at the starting point of OPEN, pointing back to n.
f) Whenever any of the successor nodes is a goal node, drain with the solution taken by tracing back through the pointers; else go to (b).

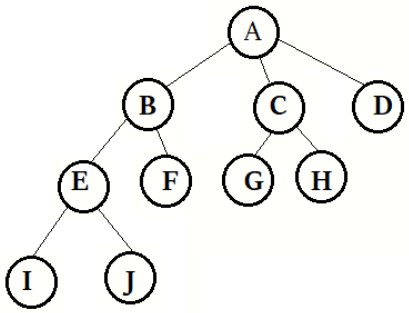

Figure 1. Depth First Search

DFS always manipulates first the deepest node, the farthest down from the root of the tree. To prevent from the consideration of unacceptable link paths, a depth bound is frequently applied to give boundary of the depth of the search. The DFS must investigate the tree in the order like:

A → B → E → I → J → F → C → G → H → D

DFS with iterative deepening is a search dealing with many of the faults of the DFS. The concept of this searching is to bring to pass a level by level DFS. It begins with the depth bound value equal to 1. If it is not found a good, it starts to perform a DFS taking the depth bound value equal to 2. It keeps continuing, with the depth bound by increasing 1(one) with each iteration, though with each increment of depth, the algorithm, of course, re-perform the DFS to a predefined bound [11].

VII. Breadth First Search (BFS)

BFS constantly searches closest to the root node first, for this it visits all nodes of a given depth first before going down to any longer paths. Maintaining the uniformity, it pushes into the search tree. For (unbounded) depth-first search, the problem sustaining is that it can be troublesome to estimate correctly a good value for the bound. We will be able to overcome these problems with the help of breadth-

first search where we will explore (right-hand) siblings before children [14].

BFS must be most effective whenever the paths, we are considering, to a goal node are of uniform depth. Bad idea, when the branching factor (i.e. the average number of offspring for each node) is either greater or infinite. Breadth first search for the tree shown in Fig. 1 will proceed alphabetically like:

A → B → C → D → E → F → G → H → I → J

The algorithm for BFS is given below:

a) Put the starting node on a queue called OPEN
b) If the OPEN is null/empty, return failure and stop, otherwise continue
c) Carry away the first node from OPEN, and put it on another list marked as CLOSED, call this node n.
d) Expand node n, every one of its successors is being created; if no one successors is available, go directly to (b). Put the successor at the terminal point of OPEN and directs pointers from these successors back to n.
e) If any successor nodes becomes a goal node, exit with taking the solution obtained by tracing back through the pointers; else return to (b).

In (unbounded) depth-first search, the problem is that we will get lost within an infinite branch whenever there is another short branch which leads to a solution. On the other hand, the problem in depth-bounds depth-first search is that it can be troublesome to estimate correctly a good value for the bound. BFS becomes utmost effective when all paths to a goal node are of uniform depth. It is also a bad concept when the branching factor is larger or infinite. BFS is also preferred to DFS when we are worried about the long paths (or even infinitely long paths).

VIII. Bidirectional Search

Up to this point, we have described all the search algorithms (with the exception of means-end- analysis and backtracking) which are on the basis reasoning. Searching backwards from the goal node to the predecessor ones is comparatively easy. Combining forward as well as backward reasoning in searching process for finding goal/solution is called bidirectional search [15]. The concept is— replacing a single search graph, which grows exponentially, with two smaller graphs —(i) starting from initial state and (ii) starting from goal. This searching process is approximated to terminate as soon as the two graphs intersect. This Bidirectional Search algorithm guarantees us to find the shortest solution path through a general state-space graph [16].

IX. Heuristic Search Method

Heuristic, no doubt, is a thumb rule which guides search that probably leads to a solution. This rule plays an important role in search techniques as of the exponential behaviors of the most problems. Thumb rule comes in handy for minimizing the number of alternatives from the exponential numbers [17] [18]. Heuristic search contains a generic meaning, also a more specialized meaning in Artificial Intelligence. In a common sense, heuristic may be accepted for any advice becoming once and again effective but not guaranteed that it will work in every space within the architecture of heuristic search. The term heuristic commonly indicates the noted case for an authentic heuristic evaluation function [18]. Penchants are to ask for algorithmic solutions, which are formal and rigid, to a specific problem domains certainly than the general approaches development that can be selected and applied for, appropriately, different specific problems.

Heuristic search's target is to reduce the node-numbers which are searched in quest of a goal. In other terms,

problems growing combinationally large can be approached.

Although information, insight, knowledge, analogies, rules and simplification in addition to a host of other heuristic search techniques aims to minimize the number of objects being tested.

Heuristic is not able to give the guarantee for achieving a solution, though heuristics must open up to obtain this.

Many authors define the Heuristic search from different angles and perceptions: -

a) Heuristic search being a practical trick that raises the effectivity of complex problem-solving [15].
b) This search directs to a solution along the most possible link/path, giving up the nominal promising ones [20].
c) It must qualify one to turn aside the investigation of dead ends, and to use already collected data [21].

The making use of heuristic search for the construction process of solution rises the ambiguity of obtaining the result for the cause of the making use of informal knowledge (intuition, rules, laws, etc.) whose usefulness are never fully proved. This searching methods allow us to exploit inexact and uncertain data in an ordinary way. The main purpose of heuristics is to improve and facilitate the effectivity of an algorithm solving a problem. It is most necessary that the mitigation out of consideration of some subsets of objects which are still not investigated.

Modern heuristic searches are expected to be bridged the lacuna in between the completion of algorithms and optimal complexity of them [22]. Tricks are being tried to modify in order to obtain a quasi-optimal in lieu of an optimal solution along with best cost reduction [23].

X. Hill Climbing

It is a DFS with a heuristic measurement which gives order for choices as nodes are expanded. The measurement of heuristic provides the value of the estimated remaining distance to the final or goal state. Hill climbing effectivity relies on the accuracy of the heuristic calculation or measurement. The principles for performing a *hill climbing search*[2] are:

a) Construct a single element queue consisting of a zero-length having only the root node.

Repeat

b) Eliminate the first path out of the list/queue.
c) Generate new paths by prolonging the first path to every annexed nodes relating to the terminal node.
d) Avoid all new paths bearing loops.
e) If there are any paths, sort them by considering the estimated distance between their terminal nodes and the goal, until the first path in the queue terminates at the goal node or the queue is null.
f) When the goal node is obtained, announce "success" and stop, else announce failure.

The probable problems influencing hill climbing has been transparently explained [4]. They are really in connection with the issue of "global vision" versus "local vision" of the search space. In particular, the foothills problems is subject to local maxima where global maxima are sought. The plateau problem happens when heuristic measure rarely hit towards any significant gradient of proximity to a goal. The ridge problem explains just what it's called. In fact, when someone is traveling along a ridge that prevents him/her from actually attaining his/her goal, then (s)he may get impression that the search is taking him/her closer to a goal state [11].

XI. Best First Search (Best FS)

A generic algorithm for the purpose of searching by using heuristics for a state space graph is **Best FS** [2]. Best FS is used where searches having variety of heuristic which is ranging from a *state's "goodness"* to sophisticated measures. The measure is on the basis of the probability for a state that leads to a goal and that can be explained by providing examples of Bayesian statistical measures. It really finds applications regarding data and goal drivens and supports a various heuristic evaluation functions.

Best first search uses lists to control or maintain states as the DFS and BFS algorithms do. "*OPEN*" for keeping tabs on the current fringe of the search and "*CLOSED*" for keeping proof of states already been visited.

Moreover, the Best First Search (Best FS) algorithm orders states on "*OPEN*" in accordance with some heuristic estimates of their "*closeness*" to a goal. In such a way, every iteration regarding the loop is considered to be the most "*promising*" state on the "*OPEN*" list [11]. In which hill climbing fails, it becomes $short - sighted$, and in which the Best FS enhances, it becomes *local vision*. The narration of the algorithm given below closely follows the description of Lugar and Stubble Field [11].

At the instant of iteration, the Best FS takes away the first element out of the list "OPEN". The algorithm gives back the solution path leading to the goal, provided it fulfils the goal condition. Each state retains its predecessor information in order to permit the algorithm to return the final solution path. In case the first element on OPEN is not a goal, the algorithm creates its descendants. If we find a child state is already on OPEN or CLOSED, the algorithm in checks makes sure that the state records the short of the two partial solution paths. Duplicate states are not kept alive. After updating the history of ancestor nodes on OPEN and CLOSED, if they are rediscovered, the algorithm possibly find a shorter path to a goal.

This search process then heuristically calculates the states on the list OPEN and then the list is sorted out in according with the heuristic values. This introduces the "best" states in front of list OPEN. It is noticeable that naturally, scores are being heuristic and the subsequent state to be tested may be from anywhere of the state space. OPEN is often referred to as priority queue when it is maintained as a sorted list [11].

For instance,

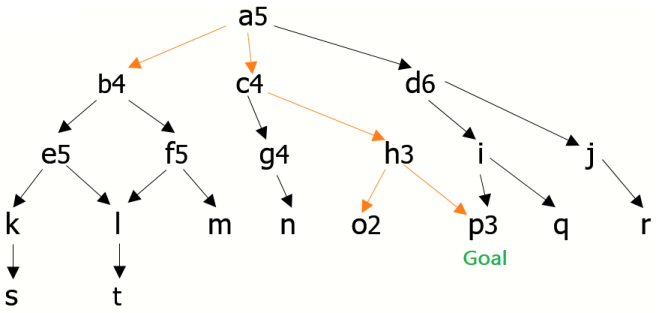

Figure 2. Heuristic Evaluations in Hypothetical State Space

The states together with the "*attached evaluations*" are being those which are actually created in the Best FS methods. We have depicted the figure called "Heuristic Evaluations in Hypothetical State Space" in Fig. 2 whereto some of its states are connected or attached. These states extended by the heuristic search (HS) are marked in rose, it is mentioned that this algorithm is not searching all of the space. The goal of Best FS is to find out the final or goal state by observing as small a number of states as possible.

Where the more informed the heuristic available, the smaller number of states are processed for finding out the goal. A marking of the execution of procedure as to *Best first search* is given below. States through the path to p3 (in fig.2) tend to keep a low heuristic values. Where the node $p3$ is the goal state in this example. Broadly speaking, heuristic is fallible. The state o2 contains a smaller value in comparison with the goal

and is investigated first. Unlike hill climbing in, this algorithm is able to recover value(s) from this error and leads to the correct goal.

$Open = \{a5\}$; $Closed = \{\ \}$

I. $measure\ a5$; $Open = \{b4, c4, d6\}$; $Closed = \{a5\}$
II. $measure\ b4$; $Open = \{c4, e5, f5, d6\}$; $Closed = \{b4, a5\}$
III. $measure\ c4$; $Open = \{h3, g4, e5, f5, d6, \}$; $Closed = \{c4, b4, a5, \}$.
IV. $measure\ h3$; $Open = \{o2, p3, g4, e5, f5, d6\}$; $Closed = \{h3, c4, b4, a5\}$.
V. $measure\ o2$; $Open = \{p3, c4, e5, f5, d6\}$; $Closed = \{o2, h3, g4, b4, a5\}$.
VI. $measure\ p3$; the solution is obtained.

Always, the most promising state on "$OPEN$" is selected by the algorithm for further expansion. Even if it is heuristic, $Best\ First\ Search$ (BFS) using for measurement of distance from the goal state may turn out to be erroneous; it does not give up all the other states and controls them on OPEN. It should retrieve some previously created “next best” state from OPEN and shifts its focus to other part of the space, when the algorithm leads to search down a false path. In the example depicted above, children as to state B(b4) were seen that they bear poor heuristic evaluations and from here, the search is transferred to state C(c4). Howsoever, children of B were kept on OPEN state and returned to later.

XII. APPLICATIONS OF SEARCH IN AI PROBLEM SOLVING

Many real-time searching methods use heuristic knowledge for the purpose of guiding planning, they can be interrupted at any state and they resume the execution at a different state and promote their plan execution time, since they solve the similar type of planning tasks, till their execution becomes optimal.

a) AI researchers, in computer science and Engineering, have generated many different tools to solve the most troublesome problems which are search techniques. They have been adopted by mainstream computers science & engineering and they are no longer considered to be a part of AI.

b) In finance, Bank systems use search algorithm techniques to organize operations, manage properties and invest in stocks. Robots are being able to beat humans in a competition of simulated financial trading. Finance related institutions have been using artificial neural network (ANN) techniques to detect claims or charges outside the norm, flagging these for human investigation.

c) In medicine, search strategies observed that the applications for organizing bed schedules (in time), make a worker/staff rotation, and give all type of medical information. ANNs can be used as clinical decision support systems for the purpose of medical diagnosis.

d) In communications, many telecommunication companies, in the management of their workforces, use heuristic search, for instance heuristic search has been deployed by BT GROUP in a scheduling application that maintains the work schedules for 20,000 engineers.

However, it can be found in application in the areas of; Aviation, Automated reasoning, Concept mining, Data mining, Robotics, Semantic web etc.

XIII. FORETHOUGHT OF SEARCH IN AI RESEARCH

Searching algorithms have depended upon links to distribute relevant results, but since researches related to artificial intelligence, input methods and natural language processing (NLP) make the techniques likely to adapt. However, keeping in

view with the acute tendency of searching techniques in AI research, we can believe that sooner than later we will be able to talk to our computers more than are typing upon them i.e., performing conversation with our laptop, with the internet, with Google– which is a much more efficient way of detecting and picking up information and learning. The appearance of the linguistic user interface (LUI) would be a transformative for human like the graphic user interface (GUI) was.

It is also come up with that with the tendency of searching in artificial intelligence research, in days to come we shall be talking of autonomous intelligence in lieu of artificial intelligence since it will no longer be artificial. At last, it will be just another form of life. We would not be able to differentiate between AI and a human being. Thus, AI would be an autonomous intelligence and it will be able to transform how search algorithms will determine what is considered to be pertinent, significant and essential [24].

XIV. CONCLUSION

It is said that around all artificial intelligence (AI) programs are doing some form of problem-solving. Searching is one of the kernel issues or reaching goals in problem-solving systems. It becomes so because whenever the system, instead of lack of knowledge, is encountered with a choice from a number of alternatives, where every choice leads to the requirement to make further choices, and so on till problem is not solved. The search amount involved can be lessened if there is a method that estimates how effective an operator will be in moving the initial problem state towards a solution. This is the place in which heuristics search methods become more effective in finding out solutions. A tension in AI among the investigations of general purpose methods which can be applied across the domains and the discovery and exploitation of special knowledge. Heuristics, Tricks and shortcuts that can also be applied particularly domains to improve efficiency. Challenge to researchers is the proficiency to manipulate the different problem-solving techniques, exemplary, search algorithms that have been developed for a variety of conditions. In this paper, a classification has been presented and we expect it will reduce this bottle-necks available at present.

XV.REFERENCES

[1]. Dan W. Patterson, “Introduction to Artificial Intelligence and expert Systems” Prentice Hall of India 2006

[2]. Rich, Knight and B Nair. “Artificial Intelligence” Tata McGraw Hill 3rd Edition

[3]. “What is Artificial Intelligence”, on 15.09.2011 from http://www.formal.standford.edu/jmc/whatisai/ node1.ht ml

[4]. P. H. Winston, Artificial Intelligence. Reading, Massachusetts: Addison-Wesley, 1984.

[5]. D. Poole, A. Mackworth and R. Goebel, Randy, Computational Intelligence: A logical Approach New York: Oxford University Press, 1998.

[6]. P. McCorduck, Machines Who Think, (2nd ed) Natick, MA: A.K. Peters Ltd, 2004.

[7]. from http://www.cs.edu/afs/cs.cmu.edu/academic/cla ss/15381- s06/www/introcol.pdf

[8]. from http://www.exampleessays.com/essay_search/C harniak_ McDermott.html

[9]. from http://en.wikipedia.org/wiki/history_of_artifici al_intellig ence

[10]. S. J. Russell, P. Norving, Artificial Intelligence; A Modern Approach (2nd Ed). Upper Saddle River, New Jersey: Prentice Hall, 2003.

[11]. G. Luger, and W. Stubble Field, Artificial Intelligence: Structures and Strategies for

Complex Problem Solving (5th ed). The Benjamin/Cummings Publishing Company, Inc, 2004.

[12]. N. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann Publishers, 1998.

[13]. from http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.19 1.288

[14]. from http://www.cs.ubbcluj.ro/-csatol/log_funk/prolog/slides/7-search.pdf

[15]. from http://en.wikipedia.org/wiki/bidirectional.search

[16]. from http://www.sci.brookyln.cuny.edu/-kopec/publications/publications/O_5_AI.pdf

[17]. L Bolc, and J. Cytowski, Search Methods for Artificial Intelligence, London; Academic Press, 1992.

[18]. from http://en.wikipedia.org/wiki/How_to_Solve_It

[19]. E. Feigenbaum and J. Feldman, Computers and Thought, New York; McGraw-Hill, 1963.

[20]. S. Amarel, On Representation of Problems of Reasoning About Actions, Machine Intelligence, No. 3, pp. 131- 171, 1968.

[21]. D. Lenat, Theory Formation By Heuristic Search: Search and Heuristics, J. Pearl (Ed.), New York, North Holland, 1983.

[22]. M. Romanycia, and F. Pelletier, what is Heuristic? Computer Intelligence, No. 1, pp. 24-36, 1985.

[23]. J. Pearl, Heuristic: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1985.

[24]. H. C. Inyiama, V. C. Chijindu and Ufoaroh S. U., Intelligent Agents - Autonomy Issues, African Journal of Computing & ICT, Vol 5. No. 4, pp 49-52, June 2012.

XVI. BIOGRAPHY

ANUP KUMAR BISWAS, is an Assistant Professor in Computer Science and Engineering, Kalyani Govt. Engineering College,West Bengal, India. He defended his Ph.D.[Engg.] thesis in 2005 in the stream of Electronics and Telecommunication Endineering at Jadavpur University. He was a Senior Research Fellow(SRF) in Faculty of Engineering and Technology (FET) from 2002 to 2005 at the Department of Electronics & Telecommuncation Engineering. Dr. Biswas has published more than 40 papers in national, international journals and conferences. The area of his scientific interests are Artificial Intelligence, nanotechnology, and computer-aided design. He is engaged in research activities and teaching last 20 years.

Cite this article as :

Anup Kumar Biswas, "Autonomous Intelligence in Problem-Solving by Searching in the field of Artificial Intelligence", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 9 Issue 6, pp. 53-62, November-December 2022. Available at doi : https://doi.org/10.32628/IJSRSET22962
Journal URL : https://ijsrset.com/IJSRSET22962