ISSN: 2321-7782 (Online)
Volume 3, Issue 4, April 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

MAO*: A Modification to AO*Algorithm

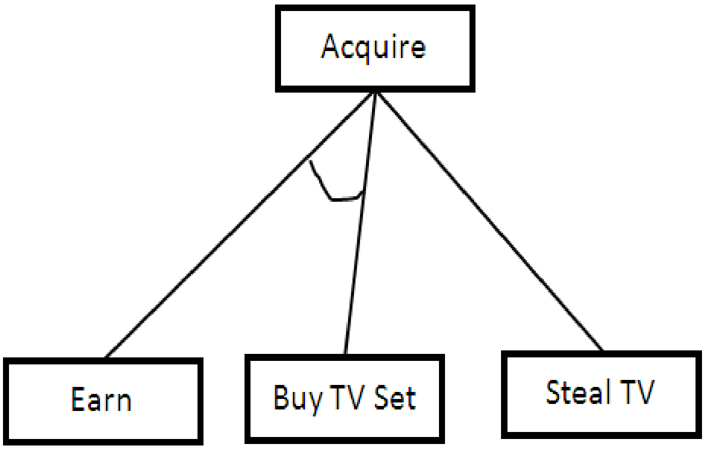
Dr. Mahesh Kumar[1]
Associate Professor
MRKIE&T
Rewari, Haryana, India

Shikha Aggarwal[2]
M.Tech Scholar
MRKIE&T
Rewari, Haryana, India

Abstract: Search techniques have always been a keen area of interest for the researchers. Many techniques have been proposed till date. AND-OR graphs provide yet another dimension for searches. Till date only AO* algorithm is capable of dealing with such problem areas. Researches proposed MAO* algorithm using heuristic technique for handling such problems more efficiently.

I. INTRODUCTION

AND OR Graphs have always been an area of interest for researchers which are very less explored. AND OR Graphs is an arrangement where two siblings in a graph represent two different actions which are complementary and equally important to solve the problem. Performing only one action will not solve the purpose. Therefore, any one from them cannot be chosen. Both the nodes are considered and solved.



Consider the above example. Here the goal is to acquire a TV Set. One could either take the f option of stealing a TV Set. Else while looking for the second option, examining the other two nodes, an arc is encountered between the two. This arc is known as AND arc. This option asks for the inclusion of both the nodes forming the AND arc. It is a series of action to reach the goal [3] i.e. to acquire a TV Set. Initially one should acquire money and then buy a TV Set.

What actually these AND OR Graphs are. They represent a series of action at the same level in a search tree. [4]

To search in an AND OR graph, the most general algorithm we often come across is AO* Algorithm. But it has been noticed that AO* Algorithm also faces the problem of local maxima.

This paper helps to study the new dimensions of AND OR graphs.

II. LITERATURE REVIEW

A lot of work has been done on Heuristic search techniques. Various search techniques have been proposed. Some of them are mentioned below with a brief introduction to their respective heuristic functions.

2.1. Hill climbing

In this technique the goal node is always known. The heuristic function, which guides the progressing of search, chooses the node with least distance from the goal node

The key feature of the technique is its simplicity. But it faced some major problems, namely:

» Local maxima

» Plateau

» Ridge

2.2. Best first search

This search technique uses an evaluation function which indicates the distance of the goal node from the current node. The evaluation function of the goal node is zero.

This search guarantees completeness. But the optimality is not guaranteed.

2.3. A algorithm*

The technique uses a heuristic function called fitness number. This fitness number is the measure of goodness of a state.

This technique guarantees optimality and completeness both.

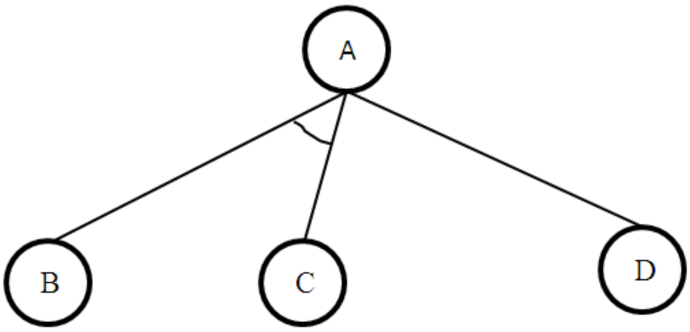
All the above techniques are very efficient in their own way but they all fail when it comes to solve the AND OR Graphs [1]. Only AO* algorithm, which is a modified version of A* Algorithm, could deal with the problem.

But literature is quiet limited regarding AO* and AND OR Graphs.

III. PROBLEM STATEMENT

The only algorithm available to solve the AND/OR graphs i.e. AO* often gets stuck into the local maxima and generate incorrect results.

Consider the AND OR Graph below. Here A is the root node with B, C, D as its children [3]



Now suppose the weight of B, C, D is as follows:

Wt. B= 2

Wt. C=3

Wt. D=4

According to AO* Algorithm, we should consider the best suited node with the lowest weight. So the move should have been made towards node B. But an AND arc exists between B and C which makes the total weight of B and C as

Total weight = Wt. B+ Wt. C =2+3=5

So ultimately the next explored node is D [4], which is obviously not the least weighing node.

Hence, there is a need of analysing the AND/OR graphs more precisely.

IV. Proposed Solution

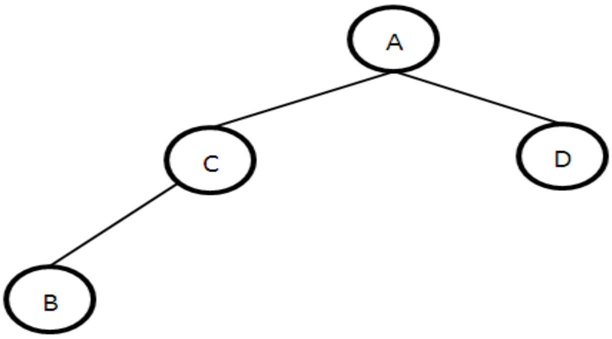
To overcome the problem of local maxima it is proposed that whenever an AND arc is encountered between two nodes, compare the weight of the two nodes. The node with the lower weight should be made a child of the node with the higher weight

The solution to the problem stated above came out to be node D. but it is quiet clear that node B should have been the answer as it was the lowest cost node.

4.1. MAO Algorithm:*

```
1. Analyse the initial graph.
2. Find pair of marked nodes connected via an AND arc. Name them N1 and N2.
    a. If (wt. (N1) > wt.(N2))
           Parent (N2) = N1
    b. Else
           Parent (N1) = N2.
3. Repeat Step2 for all pairs of AND arc connected nodes.
4. Find the fitness number of all nodes.
5. Select the root node.
6. Generate its children.
7. Add them to list of available nodes.
8. Find the fittest node from the list of available nodes.
    a. If it  is goal node
           End
    b. Else
           Goto Step6
```

Applying the MAO* to the stated problem, the new graph would be:



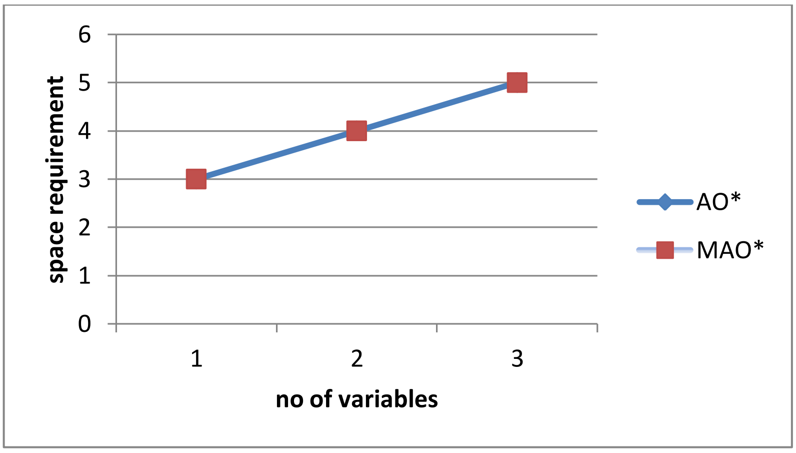
The graph converts from an AND OR graph to a simpler graph.

When the lowest weight node is searched, using the MAO* Algorithm, the Search will proceed towards C from A. From C it moves to B which is the required lowest weighing node.

V. Experimental Simulation

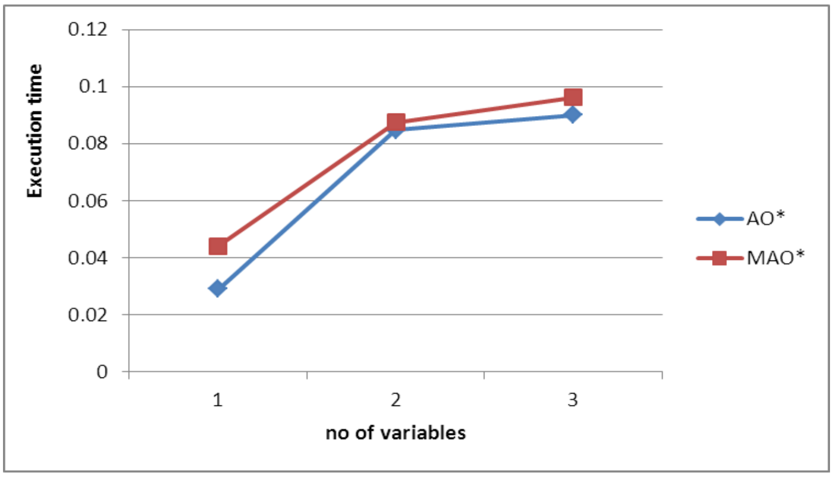
» Correct result is obtained; the problem of local maxima is removed.

» Space complexity of MAO* is exactly equal to AO* Algorithm. This is depicted in the graph by overlapping lines



» General time complexity for both MAO* and AO* are almost equal.

- The execution time difference is 0.12 milliseconds.
- The graph depicts the comparison of time complexity



» All the other properties of AO* remains intact.

» The stimulator used is MATLAB Version 7.12.0.635(R2011a)

VI. CONCLUSION

A modification of the original AO* algorithm has been generated.

By employing this algorithm, the space complexity remains same. The time complexity increases slightly by a fraction of nanoseconds. But in today's world, with the changing technologies and the commencement of high speed processors, this fraction of time is quiet negligible.

This algorithm leads to correct result by solving the problem of local maxima. The original algorithm is not able to generate correct results even after hundreds or thousands of iterations.

This new algorithm is tested over smaller problem. It could be implemented on the larger problem domains too in the future.

References

1. Artificial Intelligence by Rich & Knight
2. http://artificialintelligence-notes.blogspot.in/2010/07/problem-reduction-with-ao-algorithm.html
3. https://www.youtube.com/watch?v=6CoX9GmDRhU
4. https://www.youtube.com/watch?v=oPVj17gB7gY