IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 14, Issue 3 Ver. II (May - June 2018), PP 06-11
www.iosrjournals.org

An Exploratory study of Critical Factors Affecting the Efficiency of Uninformed Tree based Search Algorithms

O. A. Rotimi[a], A. O. Olowoye[a], R. A. Ganiyu[a], S. O. Olabiyisi[a], O. T. Amumeji[b]
[a]Dept.of Computer Science and Engineering, Ladoke Akintola University of Technology (LAUTECH), Ogbomoso, Nigeria.
[b]Dept. of Mathematical Sciences, Ondo state University of Technology, Okitipupa, Nigeria.
(All Authors are correspondence author)

Abstract: *Uniformed search (also called blind search) is a technique that has no information about its domain and the only thing that it can do is to distinguish a non-goal state from a goal state. The efficiency of bidirectional, breadth first, depth first, iterative deepening and uniform cost search in terms of time taken, memory usage and number of nodes visited was studied. Experiments were conducted on these algorithms by varying the input route lines of Arik airlines and results subjected to factor analysis by SPSS. The performance of each of the experimented algorithm was evaluated using the number of nodes visited, time taken and memory usage as performance metrics. The results showed that number of node visited contributed 61.92, 58.12, 59.63, 60.08 and 56.94 % for bidirectional, breadth first, depth first, iterative deepening and uniform cost search respectively, time taken contributed 27.13, 35.33, 26.92, 28.46 and 35.06 % and memory usage was the least contributing 10.95, 06.56, 13.45, 11.47 and 08.00% for bidirectional, breadth first, depth first, iterative deepening and uniform cost search techniques respectively. The study revealed that number of node is the most critical factor and uniform cost is the most efficient uninformed tree based search algorithm.*

Index Term: *SPSS - Statistical Package for Social Science, BDS - Bidirectional Search, BFS -Breadth- First Search, DFS - Depth-First Search, FIFO - First-In-First-Out, IDS - Iterative Deepening Search, LIFO - Last-In-First-Out, PCA - Principal Component Analysis, SPSS-Statistical Package for Social Science, UCS - Uniform Cost Search*

Date of Submission: 07-05-2018 Date of acceptance: 22-05-2018

I. Introduction

Search is a key computational mechanism in many artificial intelligence agents and the basic principle of searching is simply a deterministic goal.The efficiency with which search techniques is carried out has a significant impact on the overall efficiency of a program. Uninformed search algorithms use no information about the likely direction of the goal state. The search algorithms are implemented as special cases of normal tree traversal [1]

Uninformed search is a class of general purpose search algorithm that operates in a brute-force way. These algorithms can be applied to various search problems but do not take into account the target problem.The search problem is based on path, path cost, solution and optimal solution. It is a sequence of states and operators; a number associated with any path which is measured in quality of the path. The smaller the path, the better it is for the path to have minimum cost [2].

This paper aim at studying the efficiency with which uninformed tree based algorithms was carried out using bidirectional, breadth first, depth first, iterative deepening and uniform cost algorithms. The efficiency of the five algorithms was analyzed in terms of number of nodes visited, memory usage and time taken. The time taken, memory usage and number of nodes visited were used in this study as decision variables to evaluate the efficiencies.

II. Background to Uninformed Tree based Search Algorithms

Tree search strategy can end up repeatedly visiting the same nodes and unless it keeps track of all nodes visited, it can take vast amounts of memory. Exploration of state space generates successors of the already explored states in searching. The search algorithms are implemented as special cases of normal tree traversal [3].Breadth-first search is an algorithm that begins at the root node and explores all the neighboring nodes. Then, for each of those nearest nodes, it explores their unexplored neighbor –nodes and so on, until it finds the goal state. Breadth-First Search expands shallowest unexpanded node with a fringe. Fringe is a node waiting in a queue to be explored. It is also called OPEN [2].

Depth first search is an uninformed search that progresses by expanding the first child node of the search tree that appears and thus going deeper and deeper until a goal node is found or until it hits a node that has no children. Then the search backtracks, returning to the most recent node if it has not finished exploring. In a non-recursive implementation, all freshly expanded nodes are added to a last- in-first- out (LIFO) stack for expansion [4].

Iterative deepening search combines the advantage of both the breadth-first and depth- first search techniques [5]. It is complete and optimal. It has a memory requirement similar to that of depth-first search. IDS may seem wasteful because states are generated multiple times but it turns out not expensive. In an IDS, the nodes on the bottom level are generated once, those on the next to bottom level are generated twice and so on, up to the children of the root, which are generated *n* times [6].

Bidirectional search idea will simultaneously search forward from start point S and backwards from goat state G. it stops when both "meet in the middle", a need to keep track of the intersection of 2 open sets of nodes. Bidirectional search is implemented by replacing the goal test with a check to see whether the frontiers of the two searches intersect. If they do, a solution has been found. The check can be done when each node is generated or selected for expansion and with a hash table, the check will take constant time [6].

Uniform Cost Search always expands the node on the fringe with minimum cost g(n). it should be noted that if costs are equal (or almost equal), it will behave similarly to breadth first search. Bidirectional, breadth first, depth first, iterative deepening and uniform cost have their strengths and weaknesses as to time taken, memory usage and number of nodes visited.

This paper aims at determining the most critical of the three factors. Experimental results for the decision variables were generated from an algorithms implemented in C# in which input route lines of Arik airlines were varied for the five uninformed search algorithms. Factor analysis by principal components of the obtained experimental data was carried out using Statistical Package for Social Sciences (SPSS) for the purpose of estimating the contribution of each factor to the success of the uninformed search algorithms and one factor was extracted. Further statistical analysis was carried out to generate eigenvalue of the extracted factor. The eigenvalue forms the basis for estimating the contribution of the extracted factor. Moreover, a system of linear equations which can be used to estimate the assessment of each assessor of the uninformed search algorithms was proposed.

[7] presented a paper on the practical performance of the best first, A* search and hill climbing to find the shortest path. These search algorithms were implemented by data structures and an alternative data structure which is multi-level link list was presented and applied the heuristic techniques to it. The results indicated that use of multi-level link list helps in improving the performance of the algorithms than the data structure. [8], compared the performance of popular artificial intelligence for n-puzzle and 8-queen puzzle that included BFS, DFS, A*search and hill climbing. The research looked at the complexity of each algorithm and identifies the better functioning one. The result showed that A*search is seen to perform best with its heuristic and faster convergence at cheaper cost. [9], proposed the evaluation of critical factors affecting the efficiency of searching techniques. The paper explained the efficiency of searching techniques used in a task which determine how fast such task can be completed. In this work, the efficiency of linear and binary searching techniques in terms of running time, memory usage and the numbers of comparison was evaluated. The search time, memory used and numbers of comparison were used as decision variables to evaluate their efficiencies. Experimental results for the decision variables were generated from a software tool written using Java programming language in which the arrays of number searched were varied for the four different searching techniques of linear and binary. The results were subjected to factor analysis states the eigenvalues were used to indicate how well each of the extracted factors fits the data from the experimental results. The result showed that numbers of comparison contributed; 84.459% and 72.876% for linear and binary searching techniques respectively. The time taken came second, contributing 15.538% and 27.041% while memory used was the least of all contributing 0.003% and 0 .082% for linear and binary search respectively [10]. It can be concluded that number of comparison was the most critical factor affecting the searching techniques. Binary search is the most efficient of all the searching techniques considered.

III. Materials and Methods

The decision variables of the impact of time taken, memory usage and number of nodes visited relate to one another. The general form of the mathematical model for evaluating the decision variables is presented as [11]:

$$Y_i = \sum_{k=1}^{n} a_{i,k} - X_k \ldots\ldots i = 1,2,3 \ldots\ldots.. m, \qquad (1)$$

Where Y_i represents the i^{th} assessor's observation of decision variable X_k; $a_{i,k}$ represents the assessment of k^{th}decision variable by i^{th}assessor.

This mathematical model can be expressed as:

$$\begin{bmatrix} Y_1 \\ Y_2 \\ . \\ . \\ Y_m \end{bmatrix} = \begin{bmatrix} a_{1,1}X_1 + \cdots + a_{1,3}X_3 \\ . \\ . \\ . \\ a_{m,1}X_1 + \cdots + a_{m,3}X_3 \end{bmatrix} \qquad (2)$$

The factor analysis by principal components was adopted in the evaluation of the decision variables of the impact of time taken, memory usage and number of nodes visited. The primary goal is to obtain the contribution of each of the factors to the efficiency of the uninformed search algorithms. The following statistics were generated and used for the stated objective; descriptive statistics, correlation matrix, bartlett's test and Kaiser-Meyer olkin (KMO), communalities, initial factor loadings, rotated factor loadings, factor score coefficient matrix, eigenvalues.

The descriptive statistics presents the mean and standard deviation of the raw score of each performance indices given by the sample assessors. The correlation matrix presents the degree of pair wise relationships of the performance indices. The bartlett's test of sphericity is used to test the adequacy of the sample population. Another measure of the adequacy of sample is Kaiser-mayerolkin (KMO).

In factor analysis, there is a set of factor which is generally referred to as "common factor" each of which loads on some performance indices and another set of factors which are extraneous to each of the performance indices. The proportion of a variance of a performance indices explained by the common factor is called the "communality" of the performance indices. The communality of the performance index ranges between 0 and 1, where 0 indicates that the common factors explains none of the variance and 1 indicates that all the variance is explained by the common factors.

Component matrix presents the initial factor loadings. The factor loadings associated with a specific index is simply the correlation between the factor and standard score of the index. The component matrix can be rotated by varimax, promax, equamax or quartimax for the purpose of establishing a high correlation between indices and factors. The factor score coefficient matrix can be used to evaluate the assessment of each assessor is generated. The eigenvalues and percentage variance of the factors considered are generated, as well, for the purpose of evaluating the contributions of each factor to the efficiency of uninformed search algorithms.

3.1 Data Collection, Analysis and Interpretation of Results

A randomly selected route line scheduling was generated using Arik airlines and it was incorporated into C# programming language written for each of the five uninformed search techniques. The number generated in all was 140 data and each uninformed search techniques has 28 records in terms of number of nodes, memory used and time taken. The program automatically prompts for how many random numbers to be generated and numbers of iterations because of other process sharing system resources.

3.2 Data Acquisition

The descriptive statistics of the data collected exhibits the mean and standard deviation of the rating of the impact of time taken, memory usage and number of nodes visited on the efficiency of the uninformed search algorithms by the experimental results generated. For instance, the descriptive statistics for bidirectional search presented the mean, standard deviation and the sample size of the three variables considered. While the sample sizes of the three variables remained the same, the mean and standard deviation of number of nodes visited, time taken (nanosec.) and memory usage (bits) has the results to be 13.93 and 10.684, 5552.29 and 2437.282, 5827 and 2836.084 respectively. For breadth search, the sample sizes of the three variables remained the same, the mean and standard deviation of number of nodes visited, time taken (nanosec.) and memory usage (bits) were 12.75 and 8.669, 5900.46 and 2396.102, 5172.50 and 3309.985 respectively. For depth first search, the sample sizes of the three variables remained the same, the mean and standard deviation of number of nodes visited, time taken (nanosec.) and memory usage (bits) were 12.29 and 8.272, 5328.61 and 2570.213, 5639.39 and 3173.668 respectively. Thereafter, the final data were subjected to factor analysis by principal components using SPSS package.

The extraction method was by principal component analysis and the rotation method promax with Kaiser Normalization. According to the computed analysis, bidirectional search for instance show that the correlation of 0.005 exists between number of nodes visited and time taken. The correlation of 0.341 exists between the time taken and memory usage. The bartlett's test for bidirectional for instance produces a X^2 of 17.617, degree of freedom of 3 and significance level of 0.001, which indicates the adequacy of the sample data. The results obtained from the bartlett's test and KMO test are good indicators of the suitability of factor analysis as well.

The communalities of the performance indices generated for the uninformed search techniques with principal component analysis as the extraction method are presented in table 1 through 5, with initial values for all the three factors (time taken, memory usage and number of nodes visited) considered taken as 1.000 for bidirectional, breadth first, depth first, iterative deepening and uniform cost algorithms.

Table 1: Communalities Statistics Bidirectional Search

	Initial	Extraction
Number of nodes visited	1.000	1.000
Time taken (nanosec.)	1.000	1.000
Memory usage (bits)	1.000	1.000

Table 2: Communalities Statistics Breadth First Search

	Initial	Extraction
Number of nodes visited	1.000	1.000
Time taken (nanosec.)	1.000	1.000
Memory usage (bits)	1.000	1.000

Table 3: Communalities Statistics Depth First Search

	Initial	Extraction
Number of nodes visited	1.000	1.000
Time taken (nanosec.)	1.000	1.000
Memory usage (bits)	1.000	1.000

Table 4: Communalities Statistics Iterative Deepening Search

	Initial	Extraction
Number of nodes visited	1.000	1.000
Time taken (nanosec.)	1.000	1.000
Memory usage (bits)	1.000	1.000

Table 5: Communalities Statistics Uniform Search

	Initial	Extraction
Number of nodes visited	1.000	1.000
Time taken (nanosec.)	1.000	1.000
Memory usage (bits)	1.000	1.000

The generated component score coefficient matrices are used to estimate the assessment of each assessor of the impact of time, memory and number of nodes visited on the efficiency of uninformed search algorithms.
This can be achieved by formulating a linear equation of the form:

$$C_{i,j} - \sum_{k=1}^{3} b_{k,j} S_{i,k} i = 1,2,3 \ldots\ldots n; \quad j - 1 \tag{3}$$

Where $C_{i,j}$represents the contribution of i^{th}assessor toj^{th} factor; $b_{k,j}$represents the component score coefficient of k*th* decision variable for j^{th} factor; $S_{i,k}$represents the standard score of i^{th}assessor for k^{th} decision variables and *n* represents the number of sampled assessors.
$S_{i,k}$is estimated by;

$$S_{i,k} = A + \frac{X_i + y_i}{d_i} \tag{4}$$

Where *A* represents the allowable minimum raw score for decision variables; in this instance it is *I*;x_i represents the raw score of *ith*decision variable;y_i represents the mean of the raw scores of *ith* decision variable; $d_i$represents the standard deviation of the raw scores of *ith* decision variable. For each sampled assessor, the system of linear equations for the single extracted factor can be represented as follows;

$$b_{1,1}S_{i,1} + b_{2,1}S_{i,2} + \cdots + b_{4,1}S_{i,4} = C_{i,1} \tag{5}$$

In an attempt to evaluate the percentage contribution of each factor to the efficiency of the uninformed search algorithms, the eigenvalues of each factor is generated. The eigenvalue of j^{th}factor denoted by 'E_j' is calculated by:

$$E_j = \sum_{k=1}^{3} X_{i,j}^2 i = 1,2,3; \qquad j = 1 \tag{6}$$

Where $X_{i,j}$represents the loading ofj^{th}factor on *ith* decision variable.

The eigenvalue is used to indicate how well each of the factors fit the experimental data. The percentage of variance is given as;

$$P = 100\left(\frac{E_j}{n}\right) \tag{7}$$

Where n represents the number of decision variables considered in the study. Table 6 to 10 present the eigenvalues, the percentage of variance and cumulative percentage contribution of the time taken, memory usage and number of nodes visited for each of the uninformed search algorithms according to [9].

Table 6: Total Variance Explained Bidirectional Search

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings[b]
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total
1	1.858	61.921	61.921	1.858	61.921	61.921	1.343
2	.814	27.132	89.052	.814	27.132	89.052	1.299
3	.328	10.948	100.000	.328	10.948	100.000	1.572

Table 7: Total Variance Explained Breadth First Search

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings[b]
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total
1	1.744	58.117	58.117	1.744	58.117	58.117	1.424
2	1.060	35.325	93.442	1.060	35.325	93.442	1.175
3	.197	6.558	100.000	.197	6.558	100.000	1.589

Table 8: Total Variance Explained Depth First Search

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings[b]
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total
1	1.789	59.630	59.630	1.789	59.630	59.630	1.297
2	.808	26.922	86.551	.808	26.922	86.551	1.246
3	.403	13.449	100.000	.403	13.449	100.000	1.468

Table 9: Total Variance Explained Iterative Deepening Search

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings[b]
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total
1	1.802	60.078	60.078	1.802	60.078	60.078	1.427
2	.854	28.456	88.534	.854	28.456	88.534	1.148
3	.344	11.466	100.000	.344	11.466	100.000	1.511

Table 10: Total Variance Explained Uniform Search

Component	Initial Eigenvalues			Extraction Sums of Squared Loadings			Rotation Sums of Squared Loadings[b]
	Total	% of Variance	Cumulative %	Total	% of Variance	Cumulative %	Total
1	1.708	56.935	56.935	1.708	56.935	56.935	1.339
2	1.052	35.063	91.998	1.052	35.063	91.998	1.201
3	.240	8.002	100.000	.240	8.002	100.000	1.534

The three factors contribute a total of 100% to the efficiency of the five uninformed search algorithms. From the results, time taken contributed 27.132%, number of nodes visited contributed 61.921% and memory usage contributed 10.948% impact on the efficiency of bidirectional search algorithm. This can be visualized in Figure 1.

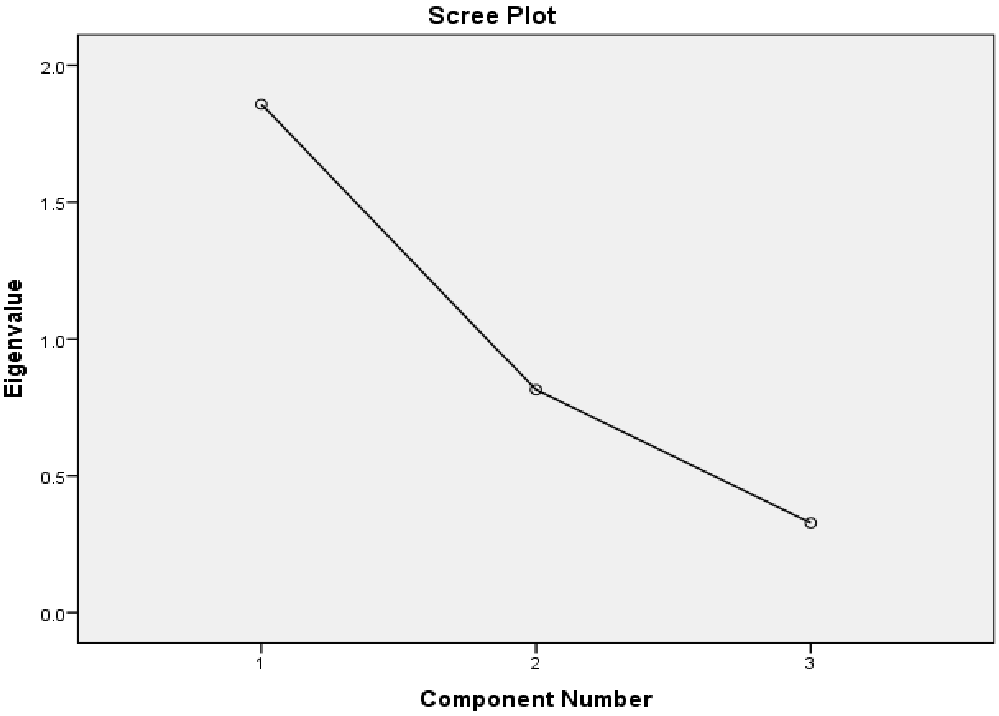

Figure 1: Scree test plot of Bidirectional Search

IV. Conclusion and Future Work

This research has established that number of node is the most critical factor and uniform cost is the most efficient uninformed tree based search algorithm. This research serves as a platform to pre-inform developer to know the most effective uninformed techniques.It assists in no small measure, code writer, team leaders to determine the worst and best case scenario for effective operational mode.Future work can be explored with system environment and other software factors affecting searching algorithms. Also, informed search algorithms could be explored using other programming languages.

References

[1]. Russell, S. and Norvig, P. (2012): "Artificial Intelligence: A Modern Approach" AuthorStuart.
[2]. Padharaic (2007) ICS271,UC Irvine Pinar Yolum:"Principles of Artificial Intelligence", Department of Mathematical and Computer Science, University of Southern Denmark, 2007.
[3]. Tamphire University of Technology, Department of Software System, Artificial Intelligence, 2012.
[4]. Thomas H., Charles E., Ronald L., and Clifford S., (2001), Introduction to Algorithms Second Edition, The MIT Press Cambridge, Massachusetts London, England McGraw-Hill.
[5]. Akinyokun O., Angaye O., and Ubaru O. (2009) -Factor Analysis of the Performance Indices and Communications Technology Projects in Public Sector of the Nigerian Economy. Journal of Technology Research, Vol. 1.
[6]. Russell, Peter Norvig, (2010): Publisher: Prentice Hall Pa." 1152
[7]. Girish P. Potdar, and Thool, R.C.,: Comparison of various Heuristic Search Techniques for Finding Shortest Path", International Journal Of Artificial Intelligence Applications (IJAIA): 5(4), Department of Information Technology, 2004.
[8]. Kuruse, Robert L.andmujah (2014): "Data Structures and Program Design in C++", Prentice-Hall, ISBN 0 13-768995-0, page 280.
[9]. Olabiyisi S.O., Aladesae T.S., OyeyinkaF.I. and Oladipo Oluwasegun (2013) Evaluation of critical factors affecting the efficiency of searching techniques.International Journal of Advanced Research in Computer Science and Software Engineering, Vol.3 (7), pp34-40.
[10]. Aladesaye T.S (2013) – Evaluation of critical factor affecting the efficiency of searching techniques. July 2013.
[11]. Olabiyisi S.O., Adetunji A.B., Oyeyinka F.I. (2012). Evaluation of the critical factors affecting the efficiency of some sorting techniques. I.J. Modern Education and Computer Science, Vol 5, no 2, pp 25-33.

O. A. Rotimi "An Exploratory study of Critical Factors Affecting the Efficiency of Uninformed Tree based Search Algorithms. "IOSR Journal of Mathematics (IOSR-JM) 14.3 (2018): 06-11.