COMPARATIVE ANALYSIS OF SOME SELECTED METAHEURISTIC ALGORITHMS FOR SOLVING INTELLIGENT PATH PLANNING PROBLEM OF MOBILE AGENTS

Olabode A. O., Abdulkareem B. Q., Ajao T. A.

Ladoke Akintola University of Technology, Ogbomoso, Oyo State, Nigeria

Corresponding Author: Olabode A. O., anthony2olabode@gmail.com

ABSTRACT: Mobile Agents (MA) are objects that migrate through the nodes of heterogenous networks to perform intelligent path finding tasks. Prioritizing the metaheuristics algorithms (PSO, ACO, GA and SA) with the best performance in terms of cost value and computational time efficiency for mobile agents path planning has been a major challenge. The performance of these selected meteheuristic algorithms was evaluated. Prototypes of mobile agents were introduced into two dimensional (2-D) workspace of adhoc network consisting of 16 nodes. The PSO, ACO, GA and SA were introduced into the mobile workspace for path planning and encoding. The corresponding path plans from PSO, ACO, GA and SA were later used for finding path and implemented in MATLAB environment. The performance of the metaheuristic techniques was evaluated using Computational Time (CT), Cost Value (CV), Program Effort (PE) and Program Size (PS).

KEYWORDS: Metaheuristic Algorithms, Mobile Agents, Path Planning, Halstead Complexity Metrics.

1. INTRODUCTION

According to [McG00] defines mobile agents as programs that can migrate from one machine to another in heterogeneous networks. The mobile agent can suspend its execution at an arbitrary point, jump to another machine, and resume execution there. Each agent is typically composed of the agent code, the agent execution thread along with an execution stack and the agent data part, which corresponds to the values of the agent's global variables [F+04]. The main metaheuristic approaches employed in Mobile Agent Path Planning are Bug Algorithm (BA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Gravitational Search Algorithm (GSA), Simulated Annealing (SA) and Tabu Search (TS) [DD99]. In case of static environment, the location of the obstacle is fixed and does not transform with time [SS05]. However, in case of dynamic environment, the position of the obstacle changes with time [SD08].

PSO algorithm is a population-based search algorithm based on the simulation of the social behavior of birds within a flock [AG12]. It optimizes a problem by taking a population of candidate solutions. It moves these particles around in the work-space by considering two mathematical formulae with respect to the position and velocity of the particle. ACO is based on the cooperative behavior of real ant colonies, which are able to find the optimal path nest to food source [BU08]. The ant colony optimization process can be viewed by multilayered graph in which ant has to move from node to node by deposition the pheromone chemical [M+13].

GA is an adaptive search method simulating the natural evolutionary process, which is first proposed by Holland. The algorithm can realize the overall learning for population composed of individuals through the population search strategy and the information exchange among individuals in population, in which each individual represents a solution of given problems in the search space, and it is generally reflected in binary coded form, also known as chromosome. SA is an optimization method, which resembles the annealing process of metals. When a hot metal temperature goes down in slower rate the crystal formed is well structured with minimum possible energy. SA was used to find an optimized path for autonomous mobile agent in a static environment. SA mimics are the annealing process of metal. At high temperature, the atoms of metals move freely with respect to each other. But, as the temperature goes down the atoms start to get ordered and forming crystal depends on cooling rate that is, if the process is fast, crystals may not be formed and it will reach to a state having higher energy state.

Improved performance of PSO, ACO, SA and GA, in terms of metrics like shortest path and reduced search time are sought-after capabilities of each of these techniques. Hence, there is a need to identify which of these techniques is most suitable for path planning of mobile agent in dynamic environment

relative to these capabilities. Therefore, this study is required to comparatively analyze the performance of these techniques for solving intelligent path planning in mobile agent environment.

2. MATERIALS AND METHODS

The comparative analysis of the selected four metaheuristics algorithms (PSO, ACO, GA and SA) for intelligent path planning of mobile agents was conducted. The sequence of steps involved in solving path planning problems was indicated by the flowchart depicted by Figure 1.

The first step involves the initialization of mobile agent migration parameters. These initialized parameters become utilized by each of the metaheuristic techniques (PSO, ACO, SA, GA), after which optimization is performed based on the selected optimization strategy for mobile agent path planning solution. The comparative analysis of selected metaheuristic technique is done and evaluated based on Halstead measures, path length, speed of operation.

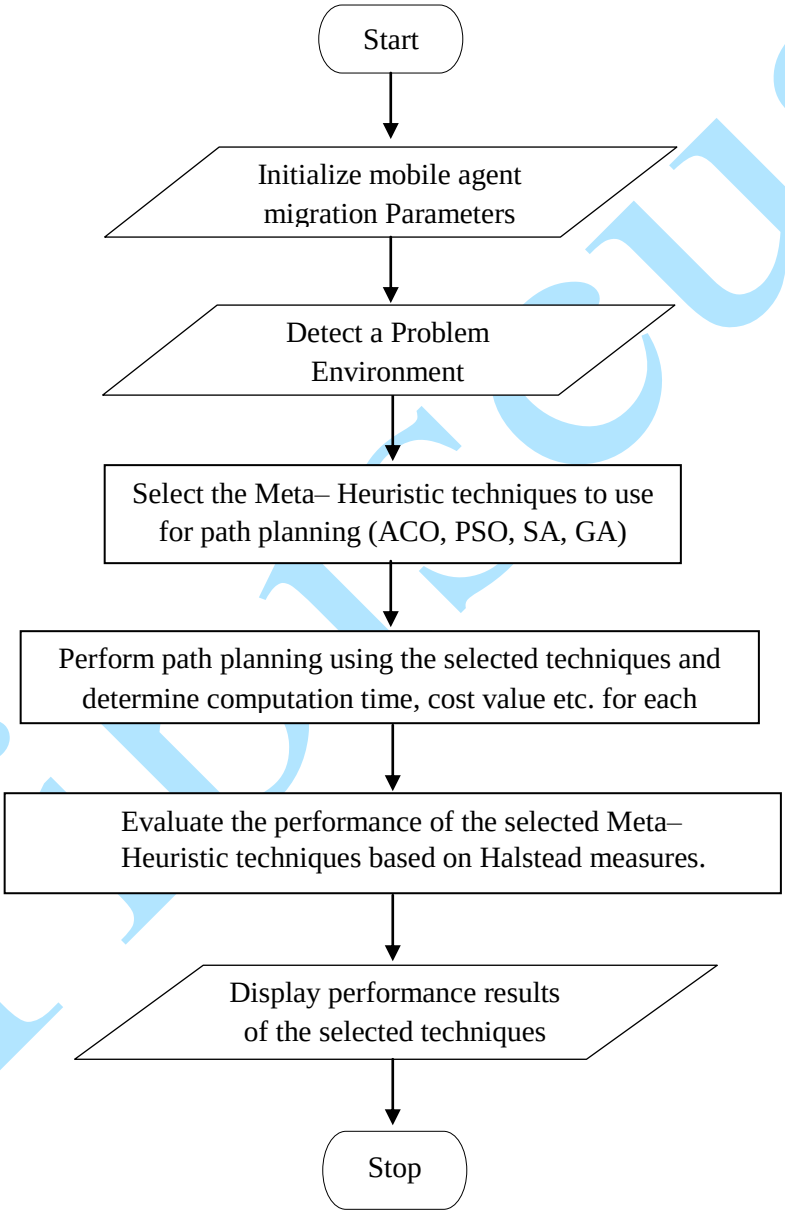

Figure 1: Flowchart showing the process of evaluating Metaheuristic Algorithms

2.1. Problem Environment and Workspace

The mobile agent environment is considered as a grid based environment in which each can be represented by an ordered pair of row number and column number. The environment of the mobile agent is represented in 2-D grid model. The map consists of a square grids of identical pattern in which the x and y axis are divided into equal part. The size of grids is considered in such a manner that they can be used to accommodate obstacles of variable size and shape . The origin or the source station (St) of the mobile agent is at the bottom left corner with coordinate (X_{st}, Y_{st}). The target station

(Dt) of the mobile agent is at a location with coordinate (X_{Dt}, Y_{Dt}). The mobile agent is considered as a point in the environment to reduce the computational complexities. The objective is to construct a shortest path from the source station St, to the target station (goal point) Dt, which avoids every obstacle in the map in the optimal time; (a path which not touch any obstacle).

2.2. Mobile Agents Path Planning Problem Formulation

Path length and path safety of the mobile agent are regarded as an important performance criteria of the path planning problem. Let $d_0$and d_{n+1} denote the start location and the destination location of a mobile agent, respectively. The planning problem can be mathematically formulated as;

$$finding: Pth = [St = d_0, d_1, d_2, \dots, d_n, Dt = d_{n+1}]$$

$$Minimize: Q_{cost} = w_1 \cdot Q_L + w_2 \cdot \frac{1}{Q_S} \quad (1)$$

$$St, d_k d_{k+1} \in semifree\ workspace,\ 0 \le k \le n$$

Where $Q_L$and Q_S denote path length and path safety, respectively, and w_1 and w_2 are two weighting parameters indicating the relative importance of $Q_L$and Q_S (w_1=w_2= 1).). To generate a point-to-point path, $Q_L$is calculated as;

$$Q_L = \sum_{k=0}^{n} Eud(d_k, d_{k+1}) \quad (2)$$

Where $Eud(d_k, d_{k+1})$ denotes the Euclidean distance between waypoint $d_k$and waypoint d_{k+1}. $Q_S$is the summation of the minimum distance of each path segment from the nearest obstacle along the path, which is calculated as;

$$Q_S = \sum_{k=0}^{n} \min_{1 \le l \le N_{ob}} \{min\ Eud(d_k, d_{k+1}, Ob_l)\} \quad (3)$$

Where N_{ob} denotes the number of obstacles.$min\ Eud(d_k, d_{k+1}, Ob_l)$ represents the minimum distance between path segment d_k, d_{k+1} and obstacle Ob_l.

2.3. Path Encoding, Evaluation and Constraints Handling

A path is constructed by a set of waypoints $d_1, d_2, \dots, d_n$ while a set of lines $l_1, l_2, \dots, l_n$ determines the x-axis values of these points. Since the set of lines $l_1, l_2, \dots, l_n$ was given beforehand during the construction of the workspace, values of $d_1, d_2, \dots, d_n$lie only with the y-axis values decided by $l_1, l_2, \dots, l_n$. These y -axis values, denoted as $(y_{d_1}, y_{d_2}, \dots, y_{d_n})$, are applied in order to encode the path. The constraint of the path planning mobile agent migration problem is to generate an obstacle-free path. It is reasonable to evaluate the constraint violation degree of each candidate path by counting its collision times with obstacles.

Given N_{ob} obstacles, the total constraint violation degree of candidate path Cp_1 is calculated as:

$$CV_{c_1} = \frac{1}{N_{ob}} \sum_{l=1}^{N_{ob}} V_{Cp_{1l}} \quad (4)$$

$$V_{c_{1l}} = \begin{cases} 1 & if\ Cp_1\ collides\ with\ Obstacle\ i \\ 0 & otherwise \end{cases} \quad (5)$$

Where CV_{ci} represesent Constraint Violation and V_{c1l} represents fitness value. After calculating the constraint violation degree and the fitness value of each candidate path, the feasibility-based rule is used to evaluate and select the elite path between any two candidate paths.

2.4. Performance Evaluation Metrics

The performance of the PSO, ACO, GA and SA is analyzed, compared and evaluated based on the following metrics;

i. Halstead Software Complexity metrics (program volume (V), program effort (E), programming difficult (D), Intelligent Content of the program (I)).
ii. Lines of code
iii. Program size
iv. Execution/Simulation Time (Speed of Operation).

2.5. Discussion of Results

The Table 1 represents the computational complexities of PSO, ACO, SA and GA with their respective parameters and values. PSO, ACO, SA and GA have the same distinct operators (n_1=14). GA used highest number of distinct operands (n_2= 83), therefore, having highest complexity than the others while SA used least number of distinct operands (n_2=44) with lowest complexity as illustrated in Figure 2.

With respect to Halstead measure, the results obtained show that the value of Program Volume (V) for PSO, ACO, SA and GA are 6951.88, 9841.49, 5588.51 and 11972.24 respectively. Similarly, in terms of Program Effort (E), PSO, ACO, SA and GA have a value of 24753.99, 58228.95, 13391.42 and 103016.47 respectively. The Programming Difficulty (D) for PSO, ACO, SA and GA reads 1.95, 1.66, 2.33 and 1.39 respectively. In terms of Intelligent Content of the program (I), PSO, ACO, SA and GA were found to have value of

35607.00, 591668.00, 239624.00 and 860461.00 respectively. The results show that GA has highest values for V, E, I but least value for D to confirm that GA has highest complexity while SA with least values of V, E, I but high value for D confirmed to have lowest complexity.

Table 1: Computational Complexities of PSO, ACO, SA and GA

Base Measures	PSO	ACO	SA	GA
No. of Distinct Operators (n_1)	14.00	14.00	14.00	14.00
No. of Distinct Operands (n_2)	54.00	68.00	44.00	83.00
Total Number of Operators (N_1)	200.00	305.00	176.00	333.00
Total Number of Operands (N_2)	942.00	1243.00	778.00	1481.00
N (N_1+N_2)	1142.00	1548.00	954.00	1814.00
n (n_1+n_2)	68.00	82.00	58.00	97.00

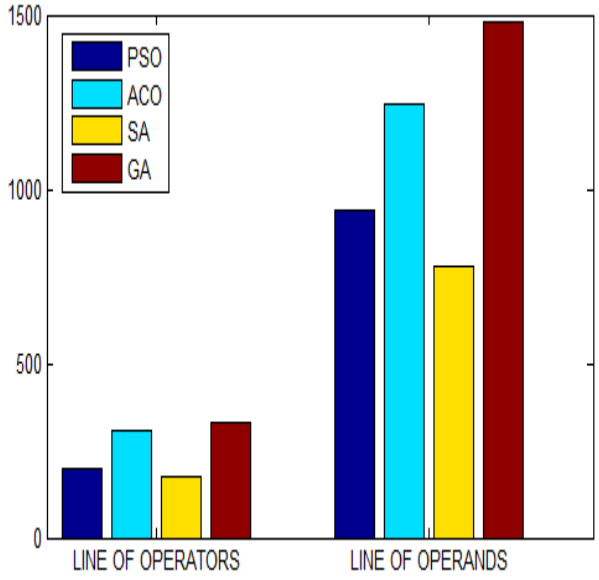

Figure 2: Barchart showing line of operators and line of operands for PSO, ACO, SA and GA

In terms of lines of code and program size, SA has more lines of code than the other while PSO has least lines of code. ACO has best space complexity while PSO has the worst space complexity as evident in Figure 3.

In terms of path length, PSO, ACO, SA and GA have a path length of 26.39, 7.00, 43.00 and 27.09 respectively. Therefore, ACO achieved the best shortest distance between the source and the destination of the mobile agent. Similarly, PSO, ACO, SA and GA have simulation time of 295.83, 8.69, 22.73 and 9.22 seconds respectively. ACO has the least simulation time. Furthermore, the cost value of PSO, ACO, SA and GA are 22.62, 5.80, 4.60 and 0.52 respectively. GA has the least cost value.

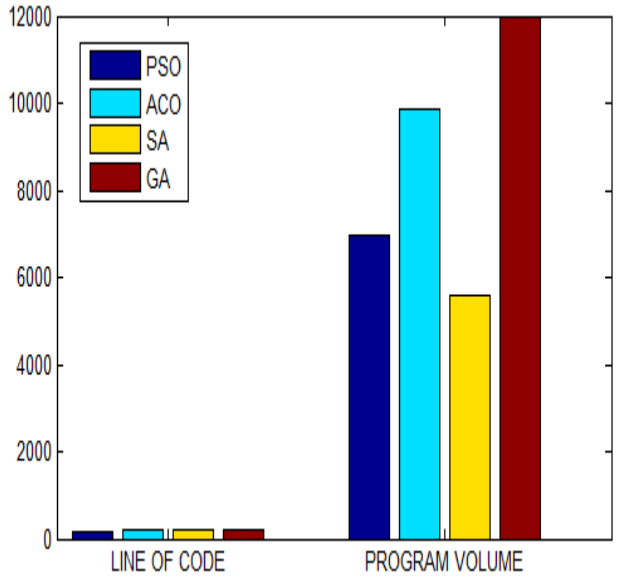

Figure 3: Barchart showing lines of code and program volume for PSO, ACO, SA and GA

Based on this study, ACO algorithm proved to be the most efficient for path planning migration problem because it gets out of local minima and converges in less time. While in terms of analyzing implementation complexity of the four algorithms, PSO and GA yielded better solutions in terms of their cost value when compared with ACO and SA.

3. CONCLUSION AND RECOMMENDATIONS

Based on the results obtained, it was concluded that ACO performed better than PSO, GA and SA in terms of time and acceleration of convergence, which gives better solution in both simulation mode using MATLAB and minimum execution time to locate the shortest path. It is highly recommended that improvement should be sought for ACO in terms of cost value and intelligent content of the program.

REFERENCES

[AG12] **Ahmadzadeh S., Ghanavati M.** – *Navigation of Mobile Robot using the Particle Swarm Optimization*, Journal of Academic and AppliedStudies (JAAS), vol. 2, pp. 32-38.

[BU08] **Bayar University** – *Reordering Triple Patterns of SPARQL Queries using Ant Colony Optimization*, Computer Engineering Department, 45140, Manissa, Turkiye, 2008.

[DD99] **Dorigo M., Di Caro G.** – *Ant colony optimization: a new meta-heuristic*. In

Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on (Vol. 2). IEEE, 1999.

[F+04] Felner A., Shoshani Y., Altshuler Y., Bruckstein A. M. – Multi agents Physical A* with Large Pheromones, Journal of Artificial Intelligence Research, Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems, Bologna, Italy, 2004.

[McG00] **McGrath S.** – *Finding the Shortest Path with A* in an Unknown Physical Environment*, 5th International Conference, MA 2001 Atlanta, GA, USA, 2000.

[M+13] **Mohajer B., Kiani K., Sameiei E., Sharifi M.** – *A New Online Random Particles Optimization Algorithm for Mobile Robot Path planning in Dynamic environments*, Hindwai Journal of Mathematical Problems in Engineering, vol. 2, pp. 1-9, 2013.

[SD08] **Sivanandam S., Deepa S.** – *Introduction to Particle Swarm Optimization and Ant Colony Optimization*. In: Introduction to Genetic Algorithms. Springer, Berlin, Heidelberg, 2008.

[SS05] **Soh B., Saxeena A.** – *Authenticating Mobile Agents Platforms using Signature Chaining without Trusted Third Parties*, IEEE International Conference on E-Technology, E-Commerce and E-Service, Hongkong, China, 2005.