Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 8, August 2014, pg.170 – 176

RESEARCH ARTICLE

GSA BASED ROUTING IN MANET

Manisha Kumari[1], Mrs. Sangeeta[2]

[1]M.TECH (CSE), Vaish College of Engg., MDU, Haryana (India)

Mehramanisha22@gmail.com

[2]Assistant Professor, Vaish College of Engg., MDU, Haryana (India)

Sangeeta.phogat@gmail.com

Abstract: A Mobile Ad hoc Network (MANET) is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies. In the mobile ad hoc network, nodes can directly communicate with all the other nodes within their radio ranges; whereas nodes that not in the direct communication range use intermediate node(s) to communicate with each other. GSA is a nature inspired algorithm which is based on the Newton's law of gravity and the law of motion. GSA is grouped under the population based approach and is reported to be more intuitive. The algorithm is intended to improve the performance in the exploration and exploitation capabilities of a population based algorithm, based on gravity rules. This research implements the GSA to improve the routing performance in the mobile Adhoc network. The research compares the results of the ABC based routing and the GSA based routing in MANET. The comparison parameters are PDR and end 2 end delay. The PDR of the proposed algorithm is better than the existing while the e2edelay gets reduced.

Keywords: MANET, Security, Attacks, Swarm Intelligence

I. INTRODUCTION

A Mobile Ad hoc Network (MANET) is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies. People and vehicles can thus be internet worked in areas without a preexisting communication infrastructure or when the use of such infrastructure requires wireless extension [1]. In the mobile ad hoc network, nodes can directly communicate with all the other nodes within their radio ranges; whereas nodes that not in the direct communication range use intermediate node(s) to communicate with each other. In these two situations, all the nodes that have participated in the communication automatically form a wireless network, therefore this kind of wireless network can be viewed as mobile ad hoc network [2].

"Mobile" and "wireless", each of these two words enforces a list of requirements, and the daunting task is to fulfill them to their best. The *mobility* means that nodes in the networks moves at any time, which implies the short duration of neighbor-hood and topology information received at any moment. In order to handle mobility correctly and efficiently, the information need to be updated regularly. The wireless nature of the medium implies the limited bandwidth capacity, which is further reduced by the high bit error rate in radio transmissions. Being such a precious resource, the bandwidth in a wireless network usually requires prudent consumption [3].

Therefore, to reduce the unnecessary use of the bandwidth is one of the main tasks in designing a protocol using wireless links. Moreover, unlike wired communication in traditional networks, wireless communication in MANETs suffers from congestion issues. When an ad hoc node transmits a message, all neighbors within its transmission range receive or overhear the message. If there are concurrent transmissions in the network, it may result in collisions and loss of the messages. The requirements of mobile nature and wireless nature are often opposite to each other and the compromise is to manage the mobility of nodes while using minimum of the bandwidth resources [3].

II. SWARM INTELLIGENCE

Swarm Intelligence appears in biological swarms of certain insect species. It gives rise to complex and often intelligent behavior through complex interaction of thousands of autonomous swarm members. Interaction is based on primitive instincts with no supervision. The end result is accomplishment of very complex forms of social behavior and fulfillment of a number of optimization and other tasks. The main principle behind these interactions is called stigmergy, or communication through the environment [4].

A well-known SI is ant colony optimization (ACO). The ACO technique is inspired by real ant colony observations. It is a multi-agent approach that was originally proposed to solve difficult discrete combinatorial optimization problems, such as the traveling salesman problem (TSP) .In the original ACO meta-heuristic, .artificial ant colonies cooperate to find good solutions for difficult discrete optimization problems. Different ACO models have been applied to FS design problems .In the FS input space was partitioned in grid type with antecedent part parameters of an FS manually assigned in advance. In the FS input space was flexibly partitioned using a fuzzy clustering-like algorithm in order to reduce the total number of rules [5]. The term swarm is used for an aggregation of animals such as fish schools, bird flocks and insect colonies such as ant, termites and bee colonies performing collective behaviour. The individual agents of a swarm behave without supervision and each of these agents has a stochastic behaviour due to her perception in the neighbourhood. Local rules, without any relation to the global pattern, and interactions between self-organized agents lead to the emergence of collective intelligence called swarm intelligence. Swarm intelligence works on two basic principles: self organization, stigmergy.

III. SWARM ALGORITHMS

For all traditional versions of the algorithms discussed in this survey, in their essence, the candidate solutions are encoded as a set of real variables, which represent a point in a multidimensional space. Various algorithms such as artificial bee colony, Ant Colony Optimization, artificial immune system, Gravitational Search Algorithm.

a. Artificial Bee Colony Algorithm

Artificial Bee Colony algorithm (ABC) was proposed for optimization, classification, and NNs problem solution based on the intelligent foraging behavior of honey bee swarm Therefore, ABC is more successful and most robust on multimodal functions included in the set with respect to DE, PSO, and GA [6], ABC algorithm provides solution in organized form by dividing the bee objects into different tasks such as employed bees, onlooker bees, and scout bees. These three bees/tasks determine the objects of problems by sharing information to others bees. The common duties of these artificial bees are as follows:

- Employed bees: Employed bees use multidirectional search space for food source with initialization of the area. They get information and all possibilities to find food source and solution space. Sharing of information with onlooker bees is performed by employee bees. An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new source is higher than that of the previous source, the employed bee memorizes the new source position and forgets the old one.
- Onlooker bees: Onlooker bees evaluate the nectar amount obtained by employed bees and choose a food source depending on the probability values calculated using the fitness values. For this purpose, a

fitness-based selection technique can be used. Onlooker bees watch the dance of hive bees and select the best food source according to the probability proportional to the quality of that food source.

- Scout bees: Scout bees select the food source randomly without experience. If the nectar amount of a food source is higher than that of the previous source in their memory, they memorise the new position and forget the previous position. Whenever employed bees get a food source and use the food source very well again, they become scout bees to find new food source by memorising the best path[7].

The main steps of the algorithm are given below:

1. Initialize
2. REPEAT
3. Move the employed bees onto their food sources and determine their nectar amounts.
4. Move the onlookers onto the food sources and determine their nectar amounts.
5. Move the scouts for searching new food sources.
6. Memorize the best food source found so far. UNTIL (requirements are met)
7. Each cycle of the search consists of three steps: moving the employed and onlooker bees onto the food sources and calculating their nectar amounts and determining the scout bees and then moving them randomly onto the possible food sources.

A food source represents a possible solution to the problem to be optimized. The nectar amount of a food source corresponds to the quality of the solution represented by that food source. Onlookers are placed on the foods by using ‘‘roulette wheel selection’’ method. Every bee colony has scouts that are the colony’s explorers. The explorers do not have any guidance while looking for food. They are primarily concerned with finding any kind of food source. As a result of such behaviour, the scouts are characterized by low search costs and a low average in food source quality. Occasionally, the scouts can accidentally discover rich, entirely unknown food sources. In the case of artificial bees, the artificial scouts could have the fast discovery of the group of feasible solutions as a task. In ABC algorithm, one of the employed bees is selected and classified as the scout bee. The classification is controlled by a control parameter called ‘‘limit’’. If a solution representing a food source is not improved by a predetermined number of trials, then that food source is abandoned by its employed bee and the employed bee associated with that food source becomes a scout. The number of trials for releasing a food source is equal to the value of ‘‘limit’’, which is an important control parameter of ABC algorithm [8].

b. Gravitational Search Algorithm

GSA is a heuristic optimization algorithm which has been gaining interest among the scientific community recently. GSA is a nature inspired algorithm which is based on the Newton’s law of gravity and the law of motion [9]. GSA is grouped under the population based approach and is reported to be more intuitive. The algorithm is intended to improve the performance in the exploration and exploitation capabilities of a population based algorithm, based on gravity rules. However, recently GSA has been criticized for not genuinely based on the law of gravity. GSA is reported to exclude the distance between masses in its formula, whereas mass and distance are both integral parts of the law of gravity. Despite the criticism, the algorithm is still being explored and accepted by the scientific community.

GSA was introduced by Rashedi et al. in 2009 and is intended to solve optimization problems. The population-based heuristic algorithm is based on the law of gravity and mass interactions. The algorithm is comprised of collection of searcher agents that interact with each other through the gravity force. The agents are considered as objects and their performance is measured by their masses. The gravity force causes a global movement where all objects move towards other objects with heavier masses. The slow movement of heavier masses guarantees the exploitation step of the algorithm and corresponds to good solutions. The masses are actually obeying the law of gravity as shown in Equation (1) and the law of motion in Equation (2).

$$F = G(M_1 M_2 / R^2) \quad (1)$$

$$a = F/M \quad (2)$$

Based on Equation (3.1), F represents the magnitude of the gravitational force, G is gravitational constant, M1 and M2 are the mass of the first and second objects and R is the distance between the two objects. Equation (1) shows that in the Newton law of gravity, the gravitational force between two objects is directly proportional to the product of their masses and inversely proportional to the square of the distance between the objects. While for Equation (2), Newton’s second law shows that when a force, F, is applied to an object, its acceleration, a, depends on the force and its mass, M.

In GSA, the agent has four parameters which are position, inertial mass, active gravitational mass, and passive gravitational mass [9]. The position of the mass represents the solution of the problem, where the gravitational and inertial masses are determined using a fitness function. The algorithm is navigated by adjusting the gravitational and inertia masses, whereas each mass presents a solution. Masses are attracted by the heaviest mass. Hence, the heaviest mass presents an optimum solution in the search space. The steps of GSA are as follows:

Step 1: Agents initialization:

The positions of the N number of agents are initialized randomly.

$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n), for\ i = 1,2, \ldots., N.$ (3)

x_i^d represents the positions of the i^th^ agent in the d^th^ dimension, while n is the space dimension.

Step 2: Fitness evolution and best fitness computation:

For minimization or maximization problems, the fitness evolution is performed by evaluating the best and worst fitness for all agents at each iteration.

Minimization problems:

$best(t) = \min_{j\in\{1,\ldots..N\}} fit_j(t)$ (4)

$worst(t) = \max_{j\in\{1,\ldots..N\}} fit_j(t)$ (5)

Maximization problems:

$best(t) = \max_{j\in\{1,\ldots..N\}} fit_j(t)$ (6)

$worst(t) = \min_{j\in\{1,\ldots..N\}} fit_j(t)$ (7)

fit j(t) represents the fitness value of the *jth* agent at iteration *t*, *best(t)* and *worst(t)* represents the best and worst fitness at iteration *t*.

Step 3: Gravitational constant (G) computation:

Gravitational constant G is computed at iteration t.

$G(t) = G_0 e^{(-at/T)}$ (8)

G0 and $-at/T$ are initialized at the beginning and will be reduced with time to control the search accuracy. *T* is the total number of iterations.

Step 4: Masses of the agents' calculation:

Gravitational and inertia masses for each agent are calculated at iteration *t*.

$M_{ai} = M_{pi} = M_{ii} = M_i \quad i = 1,2, \ldots., N$ (9)

$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}$ (10)

$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)}$ (11)

Mai and *Mpi* are the active and passive gravitational masses respectively, while *Mii* is the inertia mass of the ith agent.

Step 5: Accelerations of agents' calculation:

Acceleration of the ith agents at iteration *t* is computed.

$a_i^d(T) = F_i^d(t)/M_{ii}(t)$ (12)

F_i^d is the total force acting on ith agent calculated as:

$F_i^d(t) = \sum_{j\in Kbest, j\neq i)} rand_j F_{ij}^d(t)$ (13)

Kbest is the set of first *K* agents with the best fitness value and biggest mass. *Kbest* will decrease linearly with time and at the end there will be only one agent applying force to the others. $F_{ij}^d(t)$ is computed as the following equation:

$F_{ij}^d(t) = G(t).(M_{pi}(t) \times M_{ai}(t)/R_{ij}(t) + \varepsilon).(x_j^d(t) - x_i^d(t))$ (14)

$F_{ij}^d(t)$ is the force acting on agent *i* from agent *j* at d^{th} dimension and iteration. $R_{ij}(t)$ is the Euclidian distance between two agents *i* and *j* at iteration *t*. *G(t)* is the computed gravitational constant at the same iteration while it is a small constant

Step 6: Velocity and positions of agents:

Velocity and the position of the agents at next iteration (t+1) are computed based on the following equations:

$V_i^d(t+1) = rand_i x V_i^d(t) + a_i^d(t)$ (15)

$x_i^d(t+1) = x_i^d(t) + V_i^d(t+1)$ (16)

Step 7: Repeat steps 2 to 6

Steps 2 to 6 are repeated until the iterations reach their maximum limit. The best fitness value at the final iteration is computed as the global fitness while the position of the corresponding agent at specified dimensions is computed as the global solution of that particular problem. Figure 1 shows the flowchart of GSA.

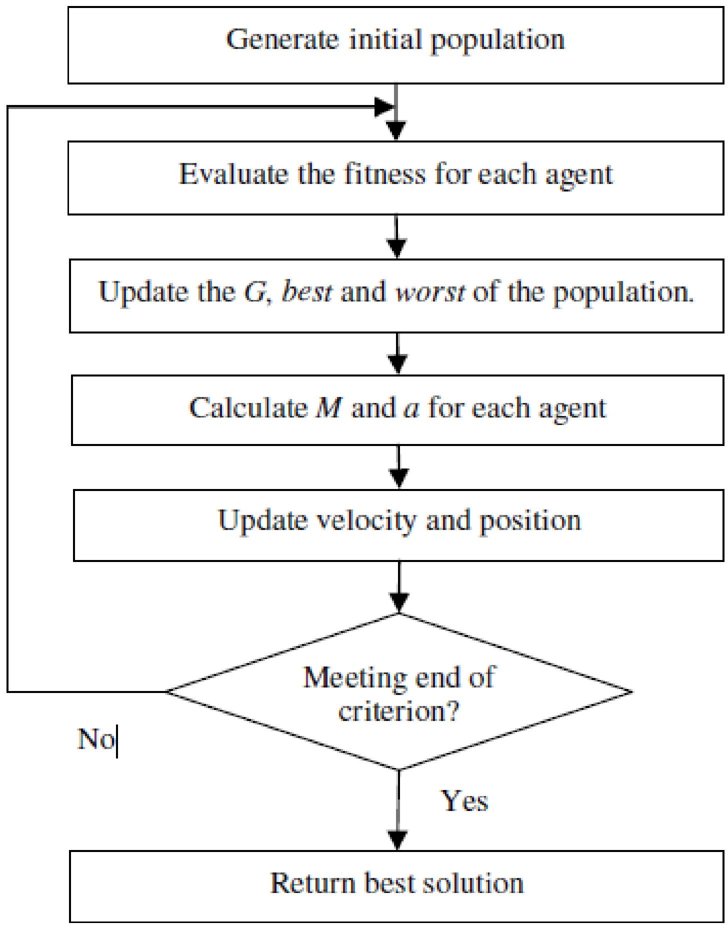

Figure 1: Flowchart of GSA

IV. RESULTS

Parameter Analysis

- **Packet Delivery Ratio (PDR)**

The ratio of the number of delivered data packet to the destination. This illustrates the level of delivered data to the destination.

∑ Number of packet receive / ∑ Number of packet send

- **End-to-end Delay**

The average time taken by a data packet to arrive in the destination. It also includes the delay caused by route discovery process and the queue in data packet transmission. Only the data packets that successfully delivered to destinations that counted.

∑ (arrive time – send time) / ∑ Number of connections

Table 1: Parameter Analysis of Existing Algorithm

Run	PDR	Delay
1	.9975	12.8717
2	.9971	10.3215
3	.9975	12.3534
4	.9972	11.9238
5	.9973	14.4252
6	.9975	13.3325

Table 2: Parameter Analysis Using Proposed Algorithm

Run	PDR	Delay
1	.9993	4.6830
2	.9993	5.4071
3	.9994	6.3206
4	.9994	5.3677
5	.9994	4.8948
6	.9994	5.1281

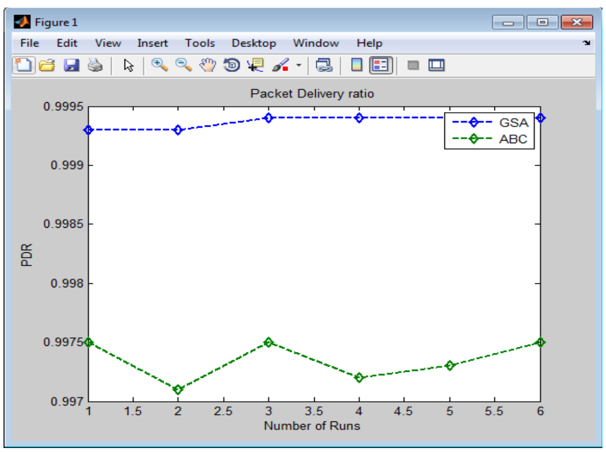

Figure 2: Comparison of PDR between Existing and Proposed

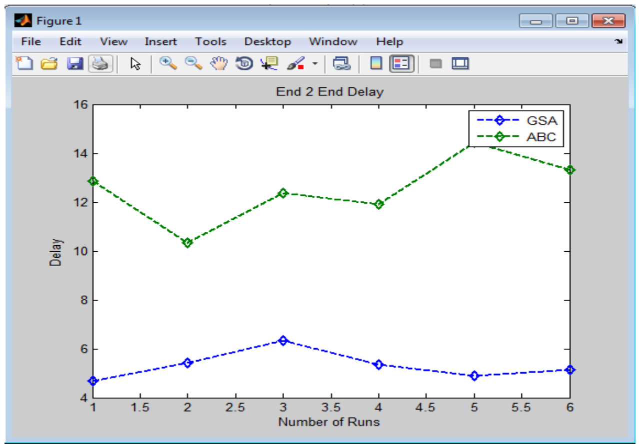

Figure 3: Comparison of Delay between Existing and Proposed

V. CONCLUSION

GSA is a nature inspired algorithm which is based on the Newton's law of gravity and the law of motion. GSA is grouped under the population based approach and is reported to be more intuitive. The algorithm is intended to improve the performance in the exploration and exploitation capabilities of a population based algorithm, based on gravity rules. This dissertation implements the GSA to improve the routing performance in the mobile Adhoc network. The dissertation compares the results of the ABC based routing and the GSA based routing in MANET. The comparison parameters are PDR and end 2 end delay. The PDR of the proposed algorithm is better than the existing while the e2edelay gets reduced. In future following work can be done: The security can be enhanced in the proposed algorithm. The algorithm can be extended to handle the congestion control.

REFERENCES

[1] M.S. Corson, J.P. Maker, and J.H. Cernicione, "Internet-based Mobile Ad Hoc Networking", IEEE Internet Computing, pages 63–70, July-August 1999.

[2] V. Park and S. Corson, "Temporally Ordered Routing Algorithm (TORA) Version 1, Functional specification", IETF Internet draft, http://www.ietf.org/internet-drafts/draftietf-manet-tora-spec-01.txt, 1998.

[3] HuiZeng, Minming Li, Hai Liu, XiaohuaJia, "Efficient Flooding in Mobile Ad Hoc Networks", 2007.

[4] Kassabalidis, Ikas, M. A. El-Sharkawi, R. J. Marks, P. Arabshahi, and A. A. Gray. *" Swarm intelligence for routing in communication networks."* In Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE, vol. 6, pp. 3613-3617. IEEE, 2001.

[5] Chia-Feng Juang, "Combination of Particle Swarm and Ant Colony Optimization Algorithms for Fuzzy Systems Design", ISBN 978-953-7619-92-3, pp. 216, February 2010, INTECH, Croatia, SCIYO.COM.

[6] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer, 2003.

[7] Shah, Habib, Rozaida Ghazali, and Nazri Mohd Nawi. "Using artificial bee colony algorithm for mlp training on earthquake time series data prediction." arXiv preprint arXiv:1112.4628 (2011).

[8] Marco Dorigo , Luca M. Gambardella, Gianni Di Caro "Ant Algorithms for Discrete Optimization" ArtifiŽ cial Life Volum e 5, Number 2.

[9] Sabri, Norlina Mohd, Mazidah Puteh, and Mohamad Rusop Mahmood. "A Review of Gravitational Search Algorithm." *Int. J. Advance. Soft Comput. Appl*5, no. 3 (2013).