Asian Journal of Information Technology 13 (4): 207-214, 2014
ISSN: 1682-3915

Bactria Foraging Optimization Algorithm with Hybrid Immigrants for Dynamic Shortest Path Routing Problem in Mobile Ad hoc Network

[1]S. Mangaiarkarasi and [2]M. Karnan
[1]Department of CSE, Shanmuganathan Engineering College, Pudukkottai, Tamilnadu, India
[2]Tamilnadu College of Engineering, Coimbatore, India

Abstract: In internet computing, swarm intelligence has shown growing interest in study dynamic optimization problems. Many approaches are developed for SI to enhance the diversity of the population and improve the performance of the algorithm for DOPs. Out of these approaches, immigrants schemes are found useful for SIs in DOPs. In this study, random, elitism based and hybrid based immigrants schemes are applied to Bacteria Foraging Optimization Algorithm (BFOA) for the Dynamic Shortest Path Routing Problem (DSPRP). The simulation results show that random immigrants are useful for BFOA in quick changing environments, whereas elitism-based immigrants are useful for BFOA in gradually ever changing environments. The BFOA algorithm with a hybrid based immigrants scheme combines the merits of the random and elitism immigrants schemes. Moreover, the simulation results show that the proposed algorithms outperform in almost all dynamic test cases and immigrant based BFOA schemes enhance the performance efficiently in DSPRP.

Key words: MANET, BFOA, immigrants schemes, dynamic optimization problem, merits

INTRODUCTION

A Mobile Ad hoc Network (MANET) (Perkins, 2001) is a self-organizing and self-configuring multi-hop wireless network, comprised of a set of Mobile Hosts (MHs) that can move around freely and cooperate in relaying packets on behalf of each other. In this study, researchers tend to investigate the Shortest Path (SP) routing that issues with finding the shortest path from a specific source to a specific destination in a given network while minimizing the total cost associated with the path. The SP problem has been investigated comprehensively. It involves a classical combinatorial optimization problem arising in many designs and planning contexts (Ahn *et al*., 2001; Ali and Kamoun, 1993).

There are many search algorithms for the SP problem: the breadth-first, the Dijkstra's search algorithm and the Bellman-Ford algorithm, etc. All of these algorithms have polynomial time complexity. Therefore, they are going to be effective in fixed infrastructure wireless or wired networks. But, they exhibit intolerably high computational complexity of real-time communications involving rapidly dynamic network topologies (Ali and Kamoun, 1993; Ahn and Ramakrishna, 2002). Since, the algorithms with polynomial time complexity are not suitable for the real-time computation of shortest paths, quite a few research works have been conducted to solve SP problems using artificial intelligence techniques, e.g., Artificial Neural Networks (ANNs) (Ahn *et al*., 2001), Genetic Algorithms (GA s) (Ahn and Ramakrishna, 2002), Ant Colony Optimization, Particle Swarm Optimization (PSO) (Mohemmed *et al*., 2008) and Bacteria Foraging Optimization (Kevin, 2010).

However, up to now of these algorithms primarily address the static SP problem. When the network topology changes, they're going to regard it as a new network and restart the solving process over the new topology. As is standard that the topology changes rapidly in MANETs due to the characteristics of wireless networks, e.g., battery exhaustion and mobility of node. Therefore for the dynamic SP problem in MANETs, these algorithms are not smart choices since they require frequent restart and cannot meet the real-time requirement. Therefore, for the dynamic SP problem in a changing network environment, researchers need to employ new suitable approaches.

In recent years, studying swarm intelligence for DOPs has attracted a growing interest due to its importance in swarm intelligence real world applications (Yang and Yao, 2008). In these Dynamic Optimization Problems (DOPs), the problem-specific fitness evaluation function and constraints of the problem such as design variables and environmental conditions may change over

Corresponding Author: S. Mangaiarkarasi, Department of CSE, Shanmuganathan Engineering College, Pudukkottai, Tamilnadu, India

time. This poses severe challenges to conventional BFOAs due to the convergence problem because once converged BFOAs cannot adapt well to the changing environment. Several approaches have been developed into BFOAs to address dynamic optimization problems such as maintaining diversity during the run via immigrants (Yang, 2008), increasing diversity after a change (Yang and Yao, 2008) using memory schemes to reuse old good solutions (Mavrovouniotis and Yang, 2011) and multi-population approaches (Branke *et al*., 2000).

In this study, various immigrants schemes are proposed and applied to solve the dynamic SP problem for BFOA in dynamically changing environment. With this approach best solution is obtained by creating immigrants to replace the worst individuals in the current population. In this way, not only can diversity be maintained but it is done more efficiently adapt the BFOA to the cyclic changing environment. Whenever, the topology of the network is changed, the optimal solutions in the new environment can be investigated using this algorithm. By simulation experiments, researchers evaluate their performance on the dynamic SP problem. The results show that Hybrid Based Immigrants (HIBFOA) significantly outperform than the other BFOA Methods (RIBFOA, EIBFOA).

LITERATURE REVIEW

Several search algorithms were formulated for SP routing problem. In (Ahn and Ramakrishna, 2002) a Genetic algorithm approach was presented for solving SP routing problem. Simulation studies show that the algorithm is indeed intensive to network topologies in respect of both route optimality and convergence. The quality of solution found to be better than other Deterministic algorithms.

Cobb and Grefenstette (1993) several modifications are applied to the standard GA on track in a changing environment. An experiment shows that the algorithm exhibits difficulties in tracking continuously changing environment. An Ant Colony Optimization (ACO) was proposed for solving SP routing problem (Kaur and Mundra, 2012; Guns *et al*., 2002). Observation shows that ants can find the shortest path between food sources and their nest. But it does not always find the optimal solution. A PSO based algorithm was presented for solving SP problems (Mohemmed *et al*., 2008). The PSO based algorithm is superior to GA (Ali and Kamoun, 1993; Branke *et al*., 2000). Hopfield neural network was proposed (Ahn *et al*., 2001). This algorithm produces a faster convergence and better route optimality than other HNN based algorithms. However, the above said algorithms are not suitable alternative for solving DSPRP in MANETs; here researchers implement the Bacteria Foraging Optimization Algorithm (BFOA) to obtain the optimal solution for DSPRP in MANETs.

MODEL FOR DYNAMIC SHORTEST PATH ROUTING PROBLEM

In this study, let us consider the ad hoc network model and then devise the DSPRP (Yang *et al*., 2010). Researchers model an ad hoc network operating within a fixed environmental region. It can be represented by an undirected and connected topology graph G_0 (N_0, E_0). Where, N_0 specifies the set of wireless nodes and E_0 specifies the set of its links (edges) connecting two adjacent nodes falling into the radio transmission range. If there exists a packet transmission in the link (i, j) then both nodes i and node j should have a radio interface, each with a universal channel. The parameters used in the study:

- G (N_0, E_0) initial Adhoc network topology graph
- G_i (N_i, E_j) Ad hoc network topology after ith chance
- S source node
- T sink node
- P_i (s, t) path from s to t in graph G_i
- C_l cost on communication link l

Ad hoc network can be represented as follows: initially given a network of wireless nodes, a delay upper bound, a source node, a sink node, researchers wish to find a delay bounded least cost loop free path on the undirected topology graph. In mobile Adhoc networks, the topology changes from time to time. The objective of the problem (DSPRP) is discovering the optimal path after every topology change.

BACTERIA FORAGING OPTIMIZATION ALGORITHM FOR SP ROUTING PROBLEM

Bacteria Foraging Optimization Algorithm (BFOA), first introduced by Kevin M. Passion in 2002, it is one of the bio-inspired optimization algorithm based on the principle of social foraging behavior of *Escherichia coli* (*E. coli*) bacteria and the natural selection which has been quite effective, applied in machine learning and optimization problems. To solve a problem, a BFOA maintains a population of bacteria and probabilistically modifies the population by reproduction and elimination and dispersal operator with the intent of seeking a near optimal solution to the problem. The BFOA design is governed by representation of bacteria, chemotaxis, swarming, reproduction and elimination and dispersal (Brownlee, 2011).

Representation of bacteria: In the proposed algorithm, any path from the source node to destination node is a feasible solution. The optimal solution is the shortest one. At the beginning a random population of strings is generated which represents feasible or unfeasible solutions. Unfeasible solutions are strings that cannot reach the destination. A bacterium corresponds to the possible solution of the problem of the optimization problem. Thus, each bacteria represent a path which consists of sequences of positive integers that represent the IDs of nodes through which a routing path passes with the source node followed by an intermediate node (via nodes) and the last node indicating the destination which is the goal. The default maximum bacteria length is equal to the number of nodes.

Chemotaxis: A chemotaxis step is a set of consequence swim steps followed by tumble. When the bacterium meets favorable environment (rich in nutrients and noxious free), it's continuous swimming in the same direction. Decrease in the cost function represents favorable environment while the increase in the cost function represents an unfavorable environment when it meets unfavorable environment it tumbles (changes direction). In this algorithm, fitness of bacterium is evaluated which further decides next movement of the bacterium. Fitness of ith bacterium is represented by cost function:

$$P^i(j+1, k, l) = P^i(j, k, l)z + c(i)\frac{\Delta(i)}{\sqrt{\Delta^T(t)\Delta(i)}} \quad (1)$$

Where:

c (i) = The basic swimming length
P^i (j+1, k, l) = Bacterium at J-represent chemotatic step
k = Reproduction
l = Elimination dispersal loop
$\Delta(i)$ = A vector in the arbitrary direction

Swarming: Every bacterium in the population organized in a group, they travel to the rich nutrition gradient. The groups in the cells have two kinds of behavior, either it may be attractant or repellant. The attractant behavior used to swarm with high fitness value when moving to nutrition gradient. The cell to cell signaling measured using the following equation:

$$J_{cc}\left(P(i,j,k,l)\right) = \sum_{i=1}^{s} p(i,j,k,ell) \quad (2)$$

Where:

S = Number of bacteria
Jcc(P(i, j, k, l)) = ith bacteria, at the jth chemotatic step, kth reproduction step and lth elimination event

Reproduction: Health status (fitness) of each bacterium is calculated after each complete chemotaxis process. It is overall sum of the cost function:

$$J^i_{health} = \sum_{j=1}^{Nc+1} J(i, j, k, l) \quad (3)$$

Where:

j^i_{health} = The health of the ith bacterium
NC = The number of chemotatic steps

The smaller j^i_{health} is the healthier the bacterium. To simulate the reproduction character in nature and to accelerate the swarming speed, all the bacteria are sorted according to their health values in an ascending order and each of the first bacteria splits into two bacteria. The characters including position and step length of the mother bacterium are reproduced to the children bacteria. Through this selection process, the remaining unhealthier bacteria are eliminated and discarded. To simplify the algorithm, the number of the bacteria keeps constant in the whole process.

Elimination and dispersal: For the purpose of improving the global search ability, elimination-dispersal event is defined after reproductive steps. The bacteria are eliminated and dispersed to random positions in the optimization domain according to the elimination-dispersal probability. This elimination dispersal event helps the bacterium avoid being trapped into local optima.

INVESTIGATED BFOA FOR DSPRP

Traditional BFOA: For the DSPRP, the traditional BFOA solutions are not precise and also produce infeasible solutions. In due course of time, it is very difficult to find the solution for DSPRP by traditional BFOA. The results produced by the traditional BFOA are not accurate due to changes in the environment.

Random based immigrants BFOA: The Random Based Immigrants BFOA (RIBFOA) uses an immigrants scheme where bacterias are generated randomly and replace the unhealthier bacteria (worst ones) of the current population for every iteration. It is understood that “the continuous modification of such algorithms is smart only if environmental changes of a problem are small to medium” (Kevin, 2010). This may be considered as come to an end result of specific verified proven fact that the previous environment has lots of chance to be same as the new one. After a change occurs, transferring information from the old environment may provide a good solution efficiently. Considering this argument, RIBFOA may be suitable when changes are not slight since it provides diversity without considering any knowledge from the old environment. Moreover, it may be suitable in fast changing environments where information from the past may not be useful, since the algorithm does not have adequate time to converge onto a high-quality solution in order to gain knowledge.

The random based immigrants approach is a quite natural and simple way around the convergence problem (Branke *et al*., 2000). It maintains the diversity level of the population through substituting some individuals of the current population with random individuals every generation. As to which individuals within the population should be substituted, usually there are two strategies: replacing random individuals or replacing the worst ones (Mavrovouniotis and Yang, 2011). In order to avoid that random immigrants disrupt the ongoing search movement too much, particularly during the period when the environment does not change, the ratio ri of the number of random immigrants to the population size n.

Elitism based immigrants BFOA: The Elitism-Based Immigrants BFOA (EIBFOA) uses an immigrants scheme where bacteria are generated by reproducing the best bacteria of the previous iteration. These immigrants also replace the unhealthier bacteria (worst ones) for every iteration as in RIBFOA. This immigrants scheme transfers knowledge from old environments and thus may be beneficial when changes are small to medium. Furthermore, it may be suitable in slowly changing environments since it needs sufficient time to locate a good optimum which can be useful to the new environment for the global optimum may be similar. Within EIBFOA, for each generation t, after the reproduction operations, the elite E (t-1) from the previous generation is used as the base to create immigrants. From E(t-1), a set of rei×n individuals are iteratively generated by E(t-1) with a probability Ped where n is the population size and rei is the ratio of the number of elitism-based immigrants to the population size. The generated individuals then act as new immigrants and replace the worst individuals within the current population. It may be seen that the elitism-based immigrants scheme combines the idea of elitism with the traditional random immigrants scheme. It uses the elite from previous population to guide the immigrants toward the current environment which is expected to improve BFOA performance in dynamic environments.

In this implementation of EIBFOA, if the elimination and the dispersal probability Ped is satisfied, the elite E (t-1) will be used to generate the new immigrants by the reproduction operation, otherwise E (t-1) itself will be directly used as the new immigrants.

Hybrid based immigrants: The Hybrid Immigrant BFOA (HIBFOA) algorithm uses an immigrant's scheme that combines both random and elitism-based immigrants. The replacing policy is the same as in RIBFOA and EIBFOA algorithms. However, half of the immigrants are randomly generated and the other half are generated by reproducing the best bacteria. HIBFOA attempts to combine the merits of both RIBFOA and EIBFOA where one is good on slowly and slightly changing environments and the other on fast and significantly changing environments.

Within HIBFOA for each generation, after the reproduction operations, the elite E (t-1) from previous generation P (t-1) is taken as the base to create immigrants in the current generation. From elite, a set of rei×n individuals are iteratively generated by reproducing E (t-1) bitwise with a elimination and dispersal probability P_{ed}, where rei is the ratio of the number of elitism-based immigrants and n is the population size and to the population size. The generated individuals then act as new immigrants and replace the worst individuals in the current population. It uses the elite from previous population to guide the immigrants toward the current environment; this way an extremely fit bacteria is never lost from the bacteria pool which is expected to improve BFOA's performance in dynamic environments.

In the algorithm implementation, after reproduction is applied on E (t-1) to generate new immigrants only if the elimination and dispersal probability is satisfied otherwise elite from the previous generation itself is used as a new immigrant. Therefore, HIBFOA may possibly be suitable under all environmental conditions.

Experimental setup: The simulation parameters that have been used for simulation are shown in the Table 1. The network model used in the simulation is composed of mobile nodes and wireless links that are considered bidirectional. The mobility model uses the Random Waypoint Model (RWP) to create the movement patterns of independent nodes for the simulation scenarios needed. RWP is one of the most widely used random-based synthetic mobility model in performance analysis of ad hoc networks. In this model, the mobile nodes start their journey from a random location and move to a random destination without any restrictions, the velocity with which the nodes move is randomly

Table 1: Parameters used during simulation

Parameters	Values
Transmitter range	250 m
Environmental size	1000×1000 m
Simulation time	50 sec
Number of nodes	20
Packet rate	2 packets sec^{-1}
Maximum speed	5 m sec^{-1}

selected from a uniform velocity distribution. After reaching a random destination the node will pause (wait) before moving to the next destination. Several scenarios were obtained from RWP by varying the velocity of the nodes and the pause times.

Simulation results: In this study, researchers present a comprehensive simulation based evaluation of routing metrics using the popular NS2 simulator. For evaluating the routing performance researchers proposed three schemes in this study: Random Based Immigrants BFOA (RIBFOA) and Elitism Based Immigrants BFOA (EIBFOA) Hybrid Based Immigrants BFOA (HIBFOA). Researchers conduct two sets of experiments. In the first set of simulations, researchers demonstrate the node adaptation in a dynamic changing environment by considering the impact of data traffic on different metrics. In the second set of simulation researchers evaluate the impact of node mobility on the overall performance of the proposed schemes. This enables us to investigate which schemes contribute to the performance more significantly. Researchers use a set of metrics to evaluate the impact of proposed schemes on routing performance. These include: packet delivery ratio, throughput, end to end delay, jitter, routing overhead and path optimality.

Impact of data traffic: Researchers first evaluate the impact of varying number of connections with different metrics for the performance of the proposed schemes. The different node density levels are obtained by keeping the area size constant and increasing the number of nodes. The results presented here are averaged over 20 runs. The results of these tests are reported in Fig. 1-6. HIBFOA

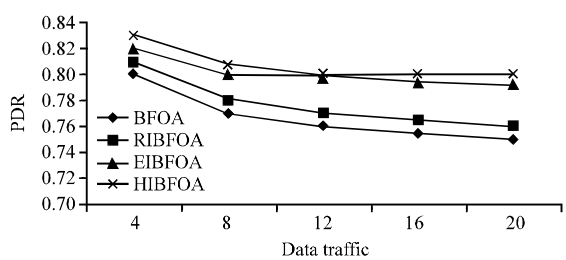

Fig. 1: Data traffic vs. PDR

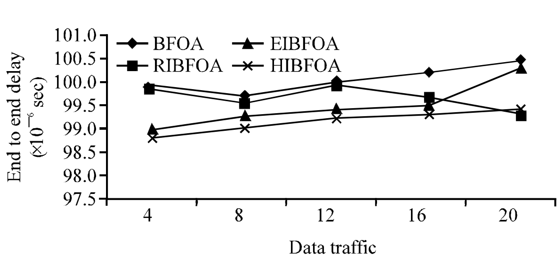

Fig. 2: Data traffic vs. end to end delay

performs better than RIBFOA and HIBFOA in terms of the packet delivery ratio, end to end delay, jitter, routing overhead, throughput and path optimality with increase the difference with the density. The numbers of connections indicate the number of nodes between which the data is transmitted or a data communication has been set. The number is incremented in steps of 4 from 4-20. The network is configured for 20 nodes; the nodes are set to move at a maximum speed of 5 msec^{-1} pausing for every 50 sec (pause time is set to 50).

Figure 1 shows that the packet delivery ratio of proposed schemes as a function of the number of nodes. It is seen from the graph that HIBFOA achieves better

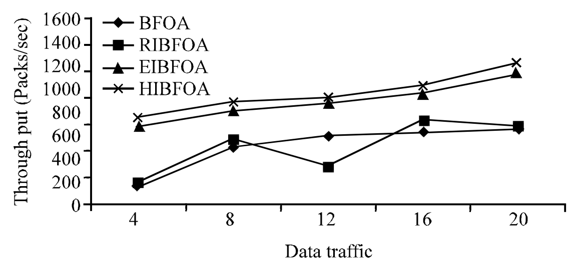

Fig. 3: Data traffic vs. throughput

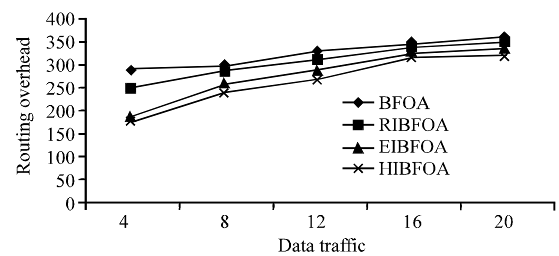

Fig. 4: Data traffic vs. routing overhead

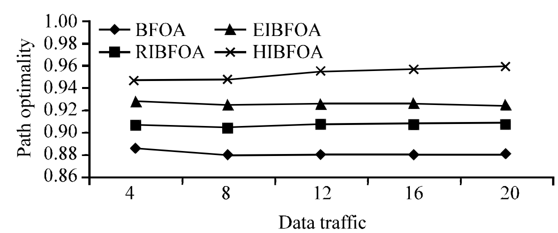

Fig. 5: Data traffic vs. path optimality

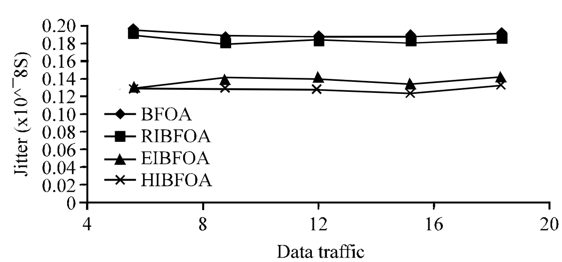

Fig. 6: Data traffic vs. jitter

packet delivery ratio than all other schemes, due to that HIBFOA can maintain a more accurate network topologies for the nodes around routing paths. Note that packet delivery ratio in BFOA, RIBFOA, EIBFOA and HIBFOA increases slightly with traffic load. This is because that the larger number of data packet forwarding which leads to a more accurate network topology and therefore better packer delivery ratio. However, it is expected that the packet delivery ratio will fall if the traffic load high enough to saturate the network. The packet delivery rate could be decreased with an increase in the data traffic on the other hand find the optimal path when a data packet arrives and thus it was able to deliver the data packets even under dynamic conditions.

As shown in Fig. 2, the average end-to-end delay has measured in varying number of data connections. At the higher node mobility HIBFOA outperformed than BFOA, RIBFOA and EIBFOA. HIBFOA showed better results in case of higher mobile conditions (20 traffic connections). As shown in Fig. 2, the delay decreases as the number of nodes increases particularly in dynamic changing environments.

Figure 3 shows the throughput of proposed schemes as number of nodes. As the number of nodes increases, the total throughput increases accordingly. The results in Fig. 3 show that HIBFOA has better throughput than all other schemes even if in high mobility scenarios. As the number of nodes increases and the mobility model applied, HIBFOA has a higher throughput up to 1245 packet sec^{-1} whereas EIBFOA has a maximum throughput up to 1123 packets sec^{-1} , all other schemes has minimum throughout as shown in Fig. 7.

Figure 4 plots the variations of routing load with the number of data traffic connections for the proposed scheme. As the routing load for proposed algorithm remains independent of the node mobility, data traffic, etc. The routing load in the HIBFOA has also remains varying because the network topology changes in this experiment. As shown in the graphs, HIBFOA has a lower routing load than all other schemes in this simulation scenario. However, the routing overhead of HIBFOA is still lower

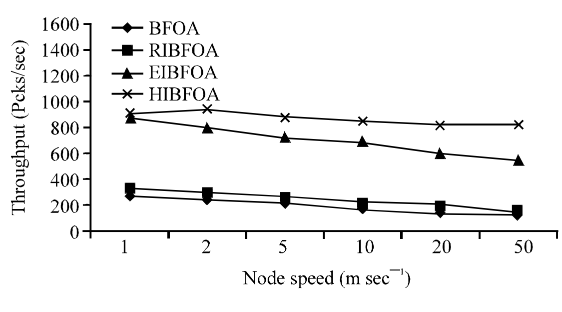

Fig. 7: Node speed vs. throughput

than that of other schemes. For low traffic, the routing overhead of HIBFOA is lower than that of BFOA, RIBFOA and EIBFOA. However, when the data traffic is high, HIBFOA outperform than all other schemes.

Figure 5 shows the variation of path optimality with the number of data traffic connections for the proposed schemes. As shown in Fig. 8, the path optimality for HIBFOA is better than the all other schemes with increase of the number of data connections from 16-20. And also shows that the path optimality of 20 MANET mobile node network scenario. Path optimality depends on total number of packets transmitting on the optimal path. HIBFOA transmits more number of packets, since packets are more likely to follow optimal paths than all other schemes.

Figure 6 shows that the jitter varies with the number of nodes in the network. It is seen that the jitter decreases as the number of nodes increases in the network. This is because, HIBFOA requires less time for packets arriving than other schemes. It is observed from the graph that the jitter value decreased from 4-20 (node topology).

Impact of node speed: In the second set of simulation, researchers evaluate the impact of node speed with different metrics for the performance of the proposed schemes. In this experiment, researchers use the same metrics as in the first set of simulation. The node mobility is changed by increasing the maximum node speed in the random waypoint model. Note that faster the node moves, the more frequently it changes its mobility parameters (speed and direction). The speed is variable from 1 m sec^{-1} which corresponds to a leisurely walk to 50 m sec^{-1}, the speed of a fast moving car. Increase in speed of the nodes results in an increase in the complexity of the network for obvious reasons. The experiment is carried out with six different node speeds: 1, 2, 5, 10, 20 and 50.

As shown in Fig. 7, the throughput of proposed algorithms as function of node speed. Recall the fact that total number of packets successfully received at the destination. As a result, HIBFOA receives the number of packets as compared with all other schemes (BFOA, RIBFOA and EIBFOA). From the outcome of the results, researchers can see that HIBFOA has better throughput than all other schemes. The results indicate that HIBFOA is more suitable than other three schemes to perform packet routing in wireless adhoc network environments with the dynamic nature of the nodes.

Figure 8 shows that variation of routing overhead as a function of node mobility (node speed). Despite the fact that HIBFOA significantly less routing overhead. Recall, when the network topology of node frequently changes

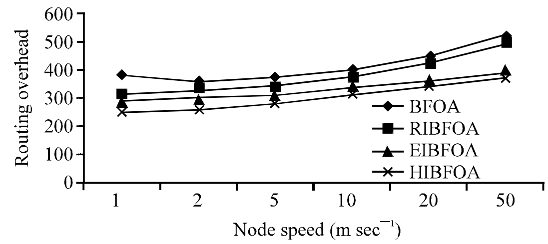

Fig. 8: Node speed vs. routing overhead

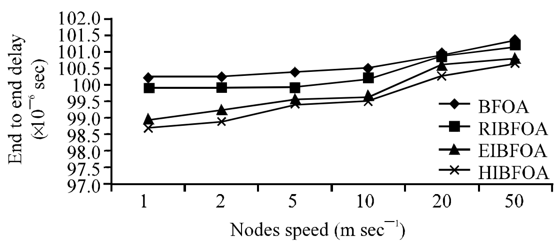

Fig. 9: Node speed vs. end to end delay

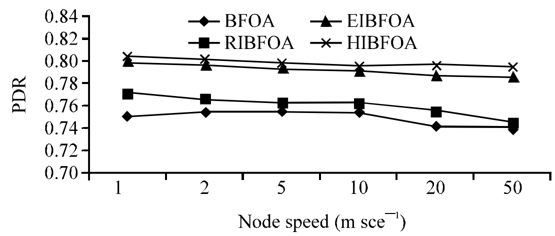

Fig. 10: Node speed vs. PDR

with several new neighbors entering the radio range. As a result, HIBFOA generates less routing overhead in order to keep up with the frequent changes of topologies. Researchers observe a similar linear increase with all other schemes. This is expected because the network topology changes in this experiment.

As shown in Fig. 9, the end-to-end delay is measured in varying speed of the nodes. At the higher node mobility HIBFOA outperformed than all other schemes. Though EIBFOA started out with better results at lower speeds, HIBFOA finishes with lower delay at higher speeds. HIBFOA showed better results in case of higher mobile conditions in this experiment.

Figure 10 shows that the packet delivery ratio of proposed algorithms as a function of node speed. Researchers observe that proposed HIBFOA can achieve comparable packet delivery ratios as the optimal scheme. However, the routing overhead generated by HIBFOA is considerably lower than that of other scheme (Fig. 8). Since, in HIBFOA most packets are forwarded along the optimal path than other schemes, HIBFOA achieves lowest end to end delay, as seen in Fig. 9. In comparison, all the other three schemes (BFOA, RIBFOA and EIBFOA) exhibit a decrease in their packet delivery radio as the speed of node increases (Fig. 10).

Overall, simulation results show that HIBFOA is significantly better at adapting to network topology and traffic load as compared to BFOA, RIBFOA and EIBFOA. The fundamental reason for this is that the packets generated in HIBFOA are expected to improve the routing performance in dynamic environment.

CONCLUSION

Different types of immigrants schemes have been successfully applied to SIS to address DOPs efficiently. In this study, researchers apply random based, elitism-based and hybrid based immigrants schemes into BFOA for the DSPRP, resulting in the RIBFOA, EIBFOA and HIBFOA algorithm, respectively. The distinction of these algorithms lies within the way immigrant bacterias are generated. The immigrant bacterias are generated randomly for RIBFOA and are generated by reproducing the best bacteria of the previous iteration for EIBFOA, respectively. For HIBFOA, half of the immigrant bacteria are generated randomly and the other half are generated using the elitism-based scheme. All immigrants replace the worst bacterias of the population on every iteration in order to gain sufficient diversity within the population which can be useful for the DSPRP. As compared to HIBFOA, the RIBFOA and EIBFOA is easy to implement and there are few parameters to adjust. Therefore, RIBFOA, EIBFOA and HIFOA have been successfully applied in the areas of MANETs. From the outcome of the results, it is shown that the proposed HIBFOA is very effective in giving the optimal solution for dynamic SP problems in cyclic dynamic environments.

REFERENCES

Ahn, C.W. and R.S. Ramakrishna, 2002. A genetic algorithm for shortest path routing problem and the sizing of populations. IEEE Trans. Evolut. Comput., 6: 566-579.

Ahn, C.W., R.S. Ramakrishna, C.G. Kang and I.C. Choi, 2001. Shortest path routing algorithm using Hopfield neural network. Electron. Lett., 37: 1176-1178.

Ali, M.M. and F. Kamoun, 1993. Neural networks for shortest path computation and routing in computer networks. IEEE Trans. Neural Netw., 4: 941-954.

Branke, J., T. Kaussler, C. Schmidth and H. Schmuck, 2000. A multi-population approach to dynamic optimization problems. Proceedings of the 4th International Conference on Adaptive Computing in Design and Manufacture, July 8-12, 2000, Las Vegas, USA., pp: 299-308.

Brownlee, J., 2011. Clever Algorithms: Nature-Inspired Programming Recipes. Jason Brownlee, USA., pp: 257-264.

Cobb, H.G. and J.J. Grefenstette, 1993. Genetic algorithms for tracking changing environments. Proceedings of 5th International Conference on Genetic Algorithms, June 1993, Urbana-Champaign, IL., USA., pp: 523-530.

Guns, M., U. Sorges and I. Bouazizi, 2002. The ant colony based routing algorithm for Manets. Proceedings of the International Conference on Parallel Processing Workshops, August 2002, Germany, pp: 79-85.

Kaur, D. and P.S. Mundra, 2012. Ant colony optimization: A technique used in finding shortest path. Int. J. Eng. Innovative Technol., 1: 122-124.

Kevin, M.P., 2010. Bacterial Foraging optimization. Int. J. Swarm Intell. Res., 1: 1-16.

Mavrovouniotis, M. and S. Yang, 2011. Memory-based immigrants for ant colony optimization in changing environments. Proceedings of the International Conference on Applications of Evolutionary Computation, April 27-29, 2011, Torino, Italy, pp: 324-333.

Mohemmed, A.W., N.C. Sahoo and T.K. Geok, 2008. Solving shortest path problem using particle swarm optimization. Applied Soft Comput., 8: 1643-1653.

Perkins, C.E., 2001. Ad Hoc Networking. Addison-Wesly, Boston, MA., USA.

Yang, S. and X. Yao, 2008. Population-based incremental learning with associative memory for dynamic environments. IEEE Trans. Evol. Comput., 12: 542-561.

Yang, S., 2008. Genetic algorithms with memory and estimed-based immigrants in dynamic environments. Evolutionary Computation, 16: 385-416.

Yang, S., C. Hui and F. Wang, 2010. Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile Ad Hoc network. IEEE Trans. Syst. Man Cybernetics Part C: Appl. Rev., 40: 52-63.