International Journal of Energy, Information and Communications
Vol.7, Issue 2 (2016), pp.21-40
http://dx.doi.org/10.14257/ijeic.2016.7.2.03

Swarm Intelligence Low Power Routing in Network-on-Chips

Setareh Shafaghi[1], Mohammad Shokouhifar[2] and Reza Sabbaghi-Nadooshan[3]

[1,3]Department of Electrical Engineering, Islamic Azad University of Central Tehran Branch, Tehran, Iran
[2]Department of Electrical Engineering, Shahid Beheshti University G.C., Tehran, Iran
[1]strshafaghi@yahoo.com, [2]m_shokouhifar@sbu.ac.ir, [3]r_sabbaghi@iauctb.ac.ir

Abstract

Network-on-chip (NoC) is an example of modern architecture for a suitable inner connection structure for systems-on-chip (SoC). NoC is currently used to accommodate the ineffective shared bus architecture on SoC chips. NoC is built using routers that regulate traffic between the devices and wires that connect the routers to the devices and each other. Routers are based on a routing algorithm to route packets. To cope with the rapid development of processor cores in integrated circuits, adequate routing to send and receive data packets to accelerate speed is required. The major challenges in NoCs are topology, switching, and routing; the quality of NoC performance depends on their suitability. A routing algorithm is an effective factor for evaluating the performance of a network. This study investigated the best solution for routing and introduces a suitable routing algorithm that uses the heuristic method to reduce power consumption and latency in routing time and increase bandwidth. The results were compared with deterministic routing. The existing algorithms for routing in NoC are discussed and new adaptive routing algorithms for 2-D mesh topology are proposed and simulated. The results showed that the proposed algorithms can improve network parameters such as latency, power consumption, and bandwidth.

Keywords: *Network-on-chip, power, latency, routing, swarm intelligence algorithms*

1. Introduction

Network-on-chip (NoC) is an integrated circuit approach for designing a communication subsystem between cores of a System-on-Chip (SoC). Basically, an on-chip network consists of links and nodes; each node consists of a process element and a router. A graph of a network design specifies the network topology. The most common topologies are mesh, ring and torus. Among them, the simple structure and easy implementation of mesh topology make it the more common choice. Generally, NoC links can reduce the complexity of designing wires for predictable speed, power, noise, reliability [1, 2]. Figure 1 shows an on-chip network where the core in the network is connected to a switch [3]. The cores communicate with each other by sending data packets to all other nodes, and consequently, multiple paths may exist.

A routing method is effective for reducing buffer consumption and the number of hops. Selecting an appropriate routing algorithm is central to the network because it reduces power consumption and improves performance. A deterministic algorithm between a specific source and destination always selects one path. In a fully adaptive algorithm between a specific source and destination, different paths can be chosen and usually will increase performance. The logic of these algorithms is complex and causes increased power consumption. It is appropriate to use methods based on computational intelligence and methods inspired by nature to solve

ISSN: 2093-9655 IJEIC

complex problems. The advantage of natural algorithms is that they can be applied to solving a variety of problems, they have high flexibility and are easy to implement. Although they do not ensure a global optimum, in most cases, they provide an adequate approximate solution. Ant colony systems, neural network, and genetic algorithms are all examples of these methods [2].

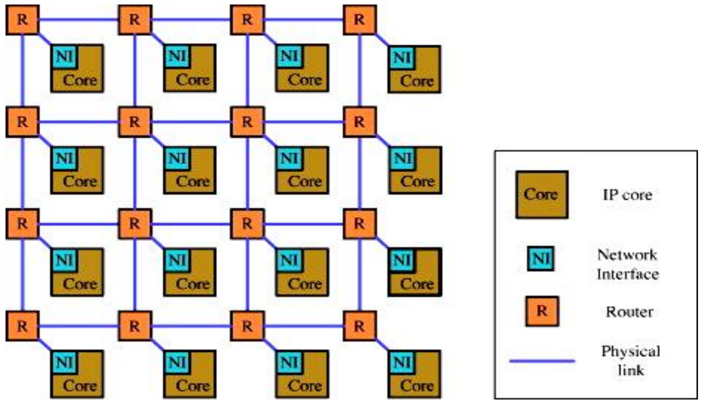


Figure 1. A Sample Structure for NoC [3]

Various algorithms can be used for routing in a NoC; the most popular is the XY algorithm. This is a deterministic algorithm where each packet is routed in one dimension and this action continues until the packet reaches the desired dimension [4, 5]. Li *et al.*, [6] described a new algorithm based on a routing algorithm called dynamic XY (Dy XY) in which adaptive routing is performed based on congestion and the lack of deadlock and live lock are considered simultaneously in the 2D mesh structure. If several short routes exist between the source and the destination, routers choose the path with less traffic to transmit the packets.

One of the most effective factors on the performance of NoC is to reduce the power consumption and load balance of the connections [1]. Modern techniques, such as the heuristic method, and evolutionary algorithms for routing have been used by Chuan-pei *et al.*, [7], who used an evolutionary algorithm and particle swarm optimization for routing. They determined the shortest path between the source and destination. Bandwidth is guaranteed and provides load balancing for the 2D mesh. This algorithm performs better when combined with a genetic algorithm (GA) [1]. The developed genetic algorithm is used to route 2D mesh that is deadlock free along the minimal path effective for balancing the load of the link. Other types of GA have been introduced to successfully reduce power consumption [2].

Ant colony optimization (ACO) was also used in NoC [8]. In [9-11], ACO-based routing is used to balance connection load and reduce power and latency in data transmission. Su *et al.*, [12] introduced an ACO algorithm without deadlocks called ACO-DAR with guarantees and no deadlocks that improves the NoC. Silva *et al.*, [13] studied the performance of ACO-based routing in NoC with 3D mesh topology. The results show that it can optimize paths for packet transmission between nodes. Ant colony algorithms have proven effective in static routing in systems designed to perform a fixed set of tasks or where the communication pattern is known. Hu *et al.*, [14] offered a finite algorithm and sub-algorithm for minimum consumption of energy in a 2D mesh structure on the basis of XY routing. Because this method is limited and constrained, when the chip-on-network is large, it does not guarantee optimal solutions. Also, Lee *et al.*, [14] proposed a GA for this purpose.

2. Swarm Intelligence Algorithms

The advantages and disadvantages of existing algorithms for finding optimal routing require selection of a new algorithm that can choose the shortest path between source and destination. One of the best ways is the use of heuristic intelligence algorithms, that are effective search methods for large spaces. They are oriented toward finding a response in the shortest time and it is appropriate for solving non-deterministic polynomial-time (NP) complete problems. These are a class of problems that cannot be solved using the usual methods. This means that the problem can be solved in Polynomial time using a Non-deterministic algorithm. Basically, a solution has to be testable in poly time. NP-complete problems go back to the roots of complexity theory; a solution to any search problem can be found and verified in polynomial time using a non-deterministic algorithm. Typically, evolutionary and swarm intelligence algorithms can efficiently use to solve NP-complete problems [24-32].

2.1. Genetic Algorithm (GA)

The theory of evolutionary computation was introduced by Richenberg in 1960. This theory was further developed by other researchers and led to the emergence of genetic algorithms in 1975 by Holand *et al.*, [15]. These algorithms use a series of code variables that have the advantage of converting continuance space to discrete space. GA principles are based on random processes. The GA is in a group of optimization methods based on imitation of evolutionary natural selection. In a GA algorithm, the solution in the binary string is coded into a chromosome. Instead of working with a single solution, research is usually initiated by a random group of chromosomes called an initial population. Each chromosome has a fitness rating that is directly associated with the optimization objective function. This approach leads to searches taking very little time [20]. If the components are properly defined, the GA will perform well.

2.2. Ant Colony Optimization (ACO)

Dorigo introduced a stochastic optimization algorithm as a nature-inspired approach to solve combinatorial optimization problems [16, 19]. The ant colony optimization algorithm takes its name from the behaviour of ants looking for food. In a natural ant colony, this arises from remaining pheromones on the ground left by ants looking for food. Ants randomly search for food around their nests; once a food source is found, they return with the food to their nest and leave pheromones on the ground. The shortest path between the nest and the food source contains more pheromones than other paths because of the frequent passage of the ants and lower evaporation. When an ant begins moving in the direction of the nest with higher pheromone intensity levels following the shortest the path to the food source, it leaves pheromones. ACO is a meta-heuristic method in which colonies of artificial ants cooperate in finding good solutions for optimization problems.

2.3. Artificial Bee Colony (ABC)

The artificial bee colony (ABC) algorithm is a new population-based meta-heuristic algorithm inspired by the process of honey bees looking for food as suggested by optimization problems of Karaboga [21]. The algorithm simulates the intelligent behavior of honey bees with a very simple and strong algorithm based on random swarm intelligence optimization [21]. ABC rivals other population-based algorithms because of the simplicity of implementation for a variety of problems.

There are three types of honey bees: workers, scouts and onlookers. Natural honey bees use a complex communication system. The system enables them to obtain information about the location and quality of food resources available outside the hive from the scouts. Communication between bees is accomplished by a bee dance that comprises scout movements that signify information about the quality of the food source, its location and position. The number of rotations represents the distance and its duration indicates the quality of the food source. This information helps the colony to send worker bees toward the food source. Onlooker bees observe a large number of bees dancing and then choose one food source [21, 22].

3. Proposed Methodology

This section presents the simulation methodology used to evaluate and optimize the proposed algorithms for NoC routing application. This study looked for the path that minimizes power consumption in the NoC mesh architecture and maximizes bandwidth in a short period of time. The power consumption of a network is proportional to the number of bit transitions on the network. To evaluate the power consumption of NoC architecture, the model shown in Eq. (1) was used [17, 18, 23]:

$$P_{total} = P_{buff\_rd} + P_{buff\_wr} + P_{crossbar} + P_{link} \tag{1}$$

where P_{buff_rd}, P_{buff_wr}, $P_{crossbar}$, P_{link} represent the average power consumption for read and write operations over time T, power consumption in a crossbar switch and power consumption of link circuitry, respectively. In order to calculate the power consumption, we use following equations [17]:

$$P_{buff\_rd} = \frac{\left(C_{buff\_rd} \times \sum_{i=1}^{N_{buff\_rd}} S(i)_{buff\_rd} \right)}{T} \tag{2}$$

$$P_{buff\_wr} = \frac{\left(C_{buff\_wr} \times \sum_{i=1}^{N_{buff\_wr}} S(i)_{buff\_wr} \right)}{T} \tag{3}$$

$$P_{crossbar} = \frac{\left(C_{crossbar} \times \sum_{i=1}^{N_{crossbar}} S(i)_{crossbar} \right)}{T} \tag{4}$$

$$P_{link} = \frac{\left(C_{link} \times \sum_{i=1}^{N_{link}} S(i)_{link} \right)}{T} \tag{5}$$

where C_{buff_wr}, C_{buff_rd}, $C_{crossbar}$ and C_{link} are power coefficient of write and read operations, power coefficient of switch traversal and power coefficient of link traversal, respectively. Moreover, N represents number of each operation, and S represents number of switching activity in all of the above equations.

3.1. GA-based Routing Methodology

In this paper, a new GA-based method is proposed to increase the convergence speed of the traditional GA, in which, the initial population and crossover operator are modified. One main difference between this method and traditional methods of optimization is that the proposed GA algorithm works with a population or a set of points at a certain moment while previous methods operated just for a certain point. This means the new algorithm allows many projects to be processed at one time by the GA. The algorithm is directed toward the optimal paths and the initial population was not selected randomly. At each step, for each current node, a lateral node selected from four available options can be conducted so that distance to targets is short and creates fewer errors in the cost function. It was assumed that the source calculates the error of each solution having chromosomes and each node having a packet, decides the rest of path, and continues the loop. Each node selects one neighboring node and makes a cost function suitable for acceptance.

Figure 2 shows that the data packet is in node source (0,0) and the destination node source is (4,4). For example, routing node (2,2) is selected and each of the four neighboring nodes can be chosen according to their fitness.

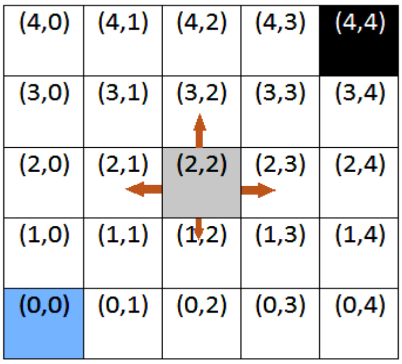


Figure 2. An Example of a 5*5 Mesh

The steps of the proposed GA are described in the following. A GA is an iterative process and each repeating step is called a generation or population. Population size is the number of existing chromosomes. In each loop, the GA started with a set of solutions represented by a set of chromosomes. This set is called the initial population and is usually created randomly.

Encoding is perhaps the most difficult step of solving problems using GAs. The concept is that it continues by offering a good representation for all possible solutions to the next steps; this is important because it depends on how to proceed in this step. Chains of chromosome or a same bit string is built for possible answers. First, a good algorithm can be simulated for the answers in a reasonably good time. After building the structure of each possible answer, the initial population of these structures can be built. In GA different methods for coding are used depending on the nature of the problem. Each gene can have a positive integer, thus each chromosome represents a sequence of integers as shown in Figure 3. For example, one possible path is represented from source node 2 to destination node 9.

2	7	3	1	10	5	6	4	8	9

Figure 3. Permutation Encoding used in the Proposed GA-based Method

Determination of an efficient objective (goal) function is essential to solving problems using GAs. After presenting the algorithm with the best answer for a problem, it is necessary to apply inverse encoding action to the answers or the same decoding until a real answer is clearly obtained. In a GA, for each chromosome, a fitness value is assigned indicating its suitability, ability to survive, and the chromosome products that determine the offspring. Here, the objective function was considered so that it minimized the distance (minimizing the number of intermediate nodes). As mentioned, GA starts with an initial population. After evaluating the fitness of each chromosome for its introduced objective function, some of the fittest chromosomes are selected as the parent population to create the next generation of offspring. The chromosomes having fewer errors are selected as the parents for the next generation. The cost function that should be minimized is as Eq. (6):

$$\mathrm{cos}t = \left(\frac{powerconsumption}{Bandwidth}\right) \times \mathrm{distance} \tag{6}$$

The simplest algorithm to represent a chromosome is a bit string. Typically, numeric parameters can be represented by integers, although it is possible to use floating point representations, which are natural to evolution strategies and evolutionary programming. The basic algorithm performs crossover and mutation at the bit level to respect the data element boundaries. For most data types, specific variation operators can be designed. Different chromosomal data types will work better or worse depending on the specific problem domain.

Gray coding is often employed when bit-string representations of integers are used. In this way, small changes in the integer can be readily affected through mutations or crossovers. This has been found to help prevent premature convergence, in which too many simultaneous mutations (or crossover events) must occur in order to change the chromosome to a better solution.

Results from schemata theory suggest that the smaller the alphabet, the better the performance, but it was initially surprising to researchers that good results were obtained using real-valued chromosomes [23, 10]. In GAs, creating a new generation or a set of new chromosomes from answers requires application of a series of changes for the selected answers or chromosomes. The proposed operators for these changes are crossover and mutation. Crossover is a process in which the old generation chromosomes are combined so that the new generation improves in comparison with the old generation. In other words, chromosomes considered in the selection step as a parent, exchange their genes in this section and create new member chromosomes called offspring that contain genes from both parents. Different methods have been introduced because the crossover operator is not generally useable for such a problem. Mutation is an operator that creates another possible answer. It randomly selects a point on a selected chromosome and changes its form.

At the each step of the GA, after crossover and mutation, the offspring with the best parents of the previous generation are replaced in the new generation and used in the next iterations as a new generation. This operation is called reproduction. Here, the parents of the each iteration are transmitted directly to the next generation and others are created by combining the parents. The overall steps of the proposed GA-based routing algorithm are illustrated in Figure 4.

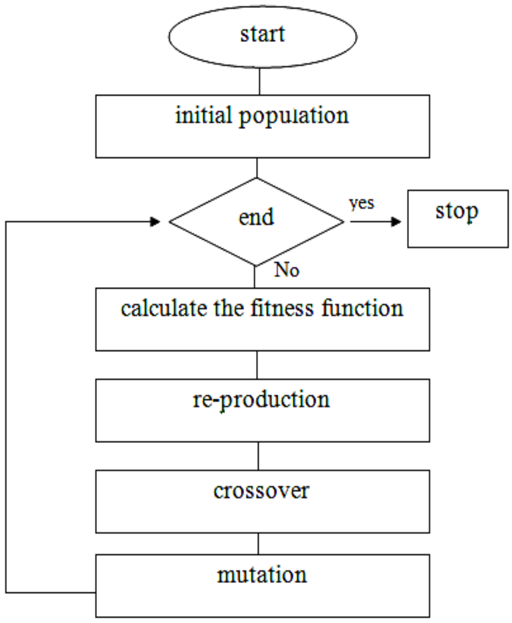


Figure 4. Flowchart of the Proposed GA-based Algorithm

3.2. ACO-based Routing Methodology

The inherent qualities of the proposed ant colony algorithm clearly make it one of the best routing algorithms. This algorithm comprises several steps. First, the number of initial ants must be selected and distributed randomly. Some ants contain the solutions and specifically chosen ants are assumed to deposit pheromones.

At the every iteration of the algorithm, evaporation and pheromone deposition in different directions mean that the probability of selection depends on pheromone intensity. The probability of selection increases as α increases, as shown in Eq. (7). If the value of α is greater than one, it increases the possibility of choosing the same initial strong solutions.

The dependence of probability selection on the strength of the pheromone causes fast convergence and algorithm solution degradation. Eq. (7) shows the probability of selection of a direction for k ants from point i to point j. In this formula, α is the effect of the pheromones.

$$p_{ij}^{k}(t) = \frac{\tau_{ij}^{\alpha}(t)}{\sum_{L} \tau_{ij}^{\alpha}(t)} \quad (7)$$

When each ant has completed its path, it is evaluated using a cost function as defined in Eq. (6) for depositing pheromones. Here, ants having the lowest error are selected for the next generation. It is worth noting that in all proposed algorithms, a blacklist was defined to prevent deadlock. If the packet passes the message to a node that has already passed, that route is deleted.

After evaluating solutions in Eq. (1), the pheromones are updated. Generally, pheromone update is performed in two steps. In the first step, the pheromones of all routes and points are evaporated; ρ is the pheromone evaporation coefficient. In the second stage, pheromones should be deposited. Updating pheromone τ is based on Eq. (8):

$$\tau_{new} = (1-\rho)\times\tau_{old} \quad (8)$$

At the every iteration of the algorithm, a number of new solutions are created and evaluated and the loop repeats until it reaches the stopping criteria. The overall flowchart of the conventional ACO algorithm can be seen in Figure 5.

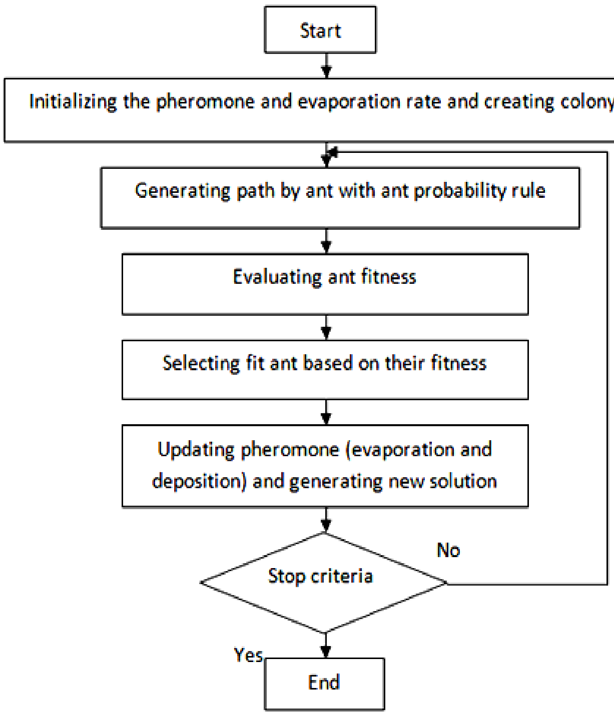


Figure 5. Flowchart of the Conventional ACO

3.3. Modified ACO-based Routing Methodology

The ACO algorithm can achieve the optimal solution to solve the routing problem, but to improve the quality of the solutions the use of parallel algorithms or a combination of algorithms is necessary. To improve the ant colony, create population diversity, avoid local minima for the algorithm and the absence of early convergence, a heuristic ant colony algorithm without mutation and ant colony algorithm with mutation is proposed. These algorithm steps are similar to the previous algorithm; the only difference is that, in the pheromone update, new solutions are produced that produce better random answers.

In the ant colony algorithm with mutation, a new solution is produced such that, after *m* new solution is obtained using the probability rule, for all *m*, a mutation will occur, similar to the type of mutation in the GA after the crossover step. This method leads to the solution of the stochastic process and consequently prevents early convergence of the ant colony and placing the algorithm into the local optima. The proposed algorithm flow chart is shown in Figure 6.

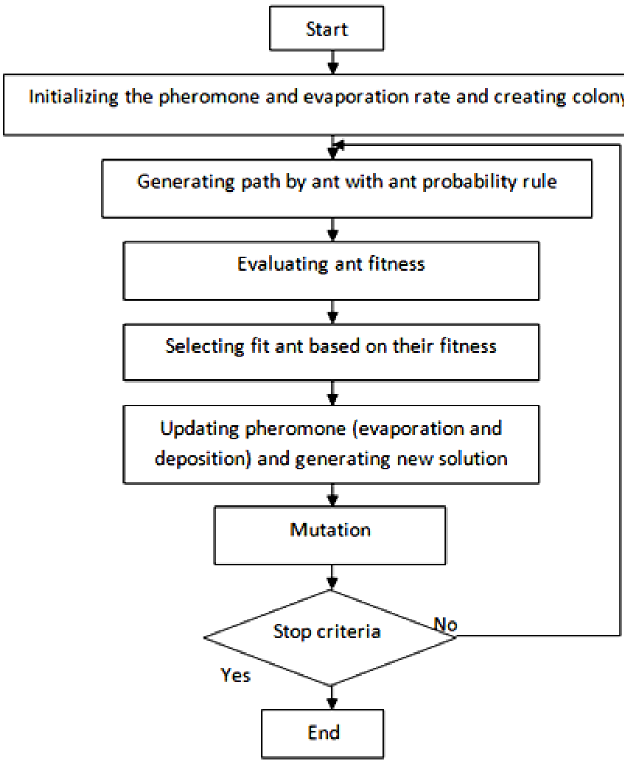


Figure 6. Flowchart of the Modified ACO-based Algorithm

3.4. ABC-based Routing Methodology

The overall stages of the proposed artificial bee colony algorithm are described in the following. First, the total number of bees, including scouts, are selected and randomly distributed. The first worker bees select the initial solutions randomly and these positions are saved. In this algorithm, each bee shows a possible solution for the problem. After a complete search, onlooker bees control the nectar quality and evaluate the resources that worker bees have found. The food resource nectar rate is obtained according to the quality of the given problem solution by that resource as Eq. (9):

$$p_i = \frac{F(nectar_i)}{\sum_{k=1}^{N} F(nectar_K)} \tag{9}$$

where $F(nectar_i)$ is the i^{th} food resource nectar rate obtained in Eq. (6) as the total available nectar around the hive. The cost is calculated from Eq. (3):

$$F(nectar_i) = \frac{1}{cost_i} \tag{10}$$

The bees that produce the least error for scouting will be selected by the scout bees. The process of recruitment is related to whether the worker bees are allocated to each scout bee in a uniform or non-uniform manner. In this study, recruitment is considered to be non-uniform, meaning that the number of worker bees for the scout bees are not equal, but their distribution is equal to initial population.

Neighbor searching is the most important aspect of the ABC algorithm. After recruitment of a scout bee, its soldiers search the neighborhood of the found solution. For any given solution, each variable in its neighborhood changes and a probability parameter P_{change} is defined. To ensure more changes when the algorithm starts and fewer in the next repetitions, P_{max} and P_{min} are defined to work toward an improved solution. The flowchart of the proposed ABC algorithm is shown in Figure 7.

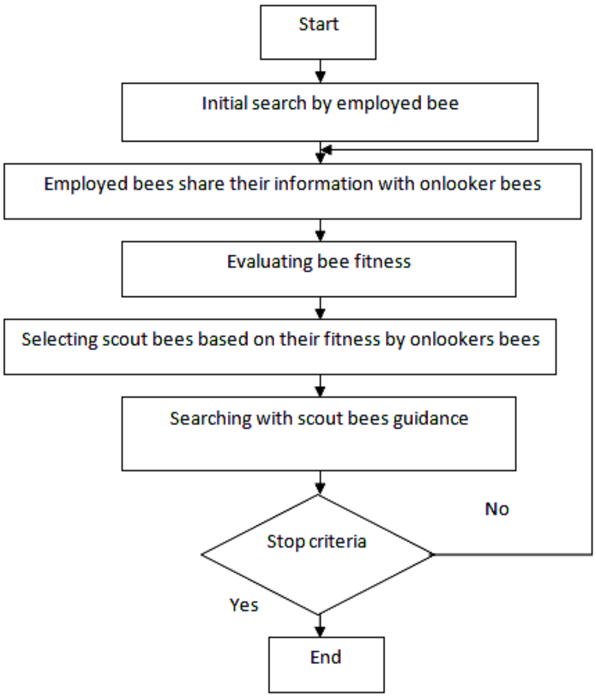


Figure 7. Flowchart of the Proposed ABC-based Routing Algorithm

4. Simulation Results

4.1. Simulation Settings

This section evaluates the proposed algorithms. The simulations were done using MATLAB R2015 software on a computer with an Intel (R) Core (TM) i7 processor with 2.5 GHz speed and 8 GB RAM. Conditions for the proposed algorithms are as follows:

A. The decision making was source based, meaning that the source decides how the data packets reach their destination.
B. Route diversity, as opposed to the deterministic XY algorithm, includes an existing path from source to destination. For the ACO,GA and ABC algorithms, all possible ways to reach the destination from the source are fully adaptive.
C. The number of destinations are uncast, meaning that each source has only one destination.
D. Deadlock often happens in interconnection network when group of packets are unable to act because of waiting each other to release some resource. All of the proposed algorithms are deadlock avoidance based, that means they guarantee that deadlock won't occur. For preventing this problem we create a black list in the program and if a loop is created then it enters the black list and will be omitted and again a new turn is started until it will be accepted.
E. The path length is minimal; the message reaches from source to destination using the minimum number of hops. For all algorithms, uniform traffic is assumed. In the traffic model, the message (packet) is sent from each processor element to

another element in the network processor with equal probability. Table 1 shows the listed items.

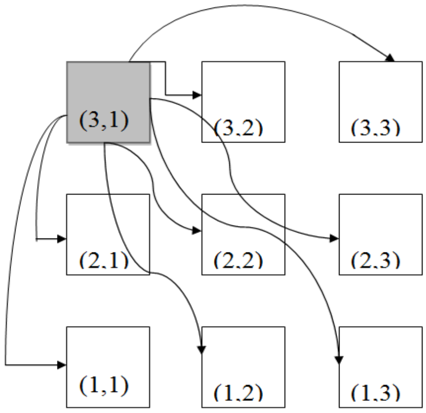


Figure 8. An Example of Uniform Traffic in Mesh 3×3

Simulations were done using 2D mesh topology. Simulations were carried out on 3 meshes having different characteristics. The routing time parameter is the time required to find a shorter and less trafficked path for each packet via each desired source node to each desired destination node with dimension in seconds. As mentioned, the aim was to find the shortest path with minimum power and maximum bandwidth.

All desired nodes are chosen for the source and destination nodes, for example, for 25 × 25 mesh, the source nodes [2, 3] and the destination nodes [16, 24] were chosen. To better compare all algorithms, the same mesh graph was considered and, in each mesh, the farthest node was selected. Each proposed algorithm was run 15 consecutive times. Table 1 shows the simulation parameters. In the following results, all power values are given in mwatts and so are bandwidth in bit per second and time in second.

Table 1. Simulation Parameters

Parameter	Description
Traffic pattern	Uniform
Decision making	Source based
Route diversity	Fully adaptive
Deadlock deadline	Deadlock avoidance based
Path length	Minimal

4.2. Results of GA

In this section, we present the results for the GA-based routing algorithm. Table 2 shows the initialized parameter values for the GA algorithm to find the best answer. The simulation results achieved by GA can be seen in Table 3. Figure 9 shows the best routing for an 18 × 18 mesh topology with a GA algorithm and its error reduction. The error reduction for all figures means percentage of decreasing error in the cost function that describes in Eq. 2 for running of each algorithm.

Table 2. Setting the GA Parameters

Parameter	Description
Population size	30
Crossover rate	0.1
Mutation rate	0.05

Table 3. Simulation Results for GA

Topology	Source Node	Destination Node	Power Consumption	Bandwidth	Routing Time
Mesh 15 × 15	(1,1)	(15,15)	11.67	16.62	0.09
Mesh 18 × 18	(1,1)	(18,18)	15.25	21.08	0.12
Mesh 25 × 25	(1,1)	(25,25)	16.2	22.24	0.13

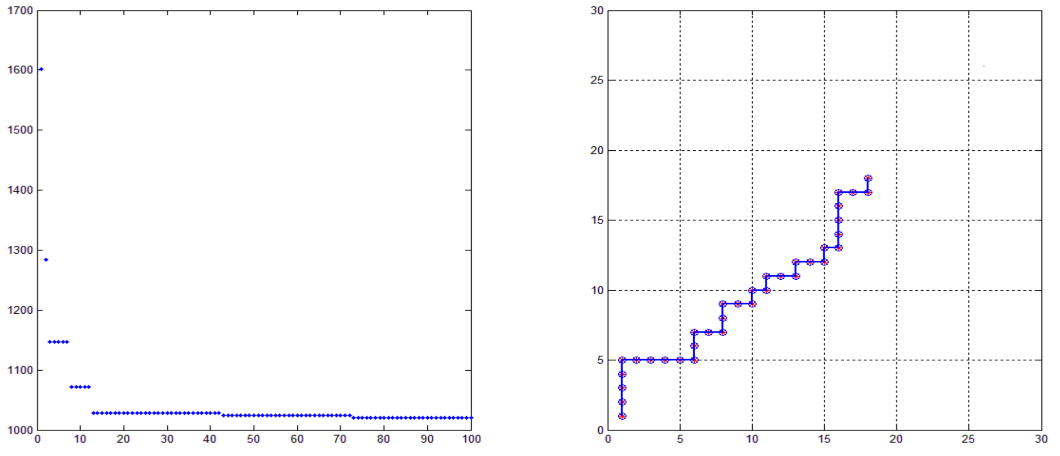


Figure 9. The best Solution found by GA

4.3. Results of Conventional ACO

Table 4 shows the ACO initialized parameter values to find the best answer. Table 5 shows simulation results achieved by the conventional ACO algorithm. Figure 10 shows the best routing for 18 × 18 mesh topology using the conventional ACO algorithm and its error reduction.

Table 4. Setting the ACO Parameters

Parameter	Description
Population size	30
Evaporation rate	0.2
Deposition rate	0.3
α	1
β	2

Table 5. Simulation Results for Conventional ACO

Topology	Source Node	Destination Node	Power Consumption	Bandwidth	Routing Time
Mesh 15 × 15	(1,1)	(15,15)	11.66	16.59	0.15
Mesh 18 × 18	(1,1)	(18,18)	16.08	23.1	0.16
Mesh 25 × 25	(1,1)	(25,25)	20.54	29.22	0.28

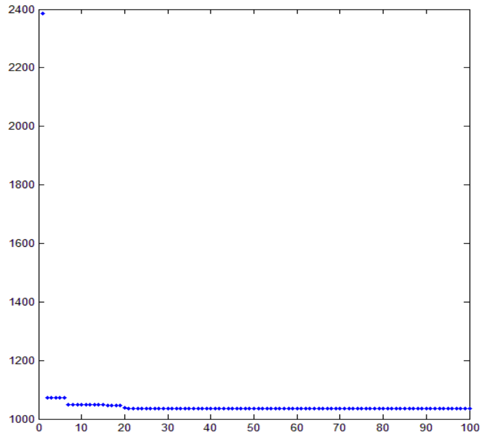


Figure 10. The Best Solution found by Conventional ACO

4.4. Results of Hybrid ACO with Mutation

Table 6 shows simulation results achieved by ACO with mutation algorithm. Figure 11 shows the best routing for the 18 × 18 mesh topology with ACO with mutation algorithm and its error reduction. Mutation, as expected, had a good influence on the results and minimized power consumption and routing time and maximized bandwidth over conventional ACO.

Table 6. Simulation Results for Hybrid ACO with Mutation

Topology	Source Node	Destination Node	Power Consumption	Bandwidth	Routing Time
Mesh 15 × 15	(1,1)	(15,15)	11.76	16.87	0.14
Mesh 18 × 18	(1,1)	(18,18)	14.92	19.81	0.2
Mesh 25 × 25	(1,1)	(25,25)	15.61	22.53	0.24

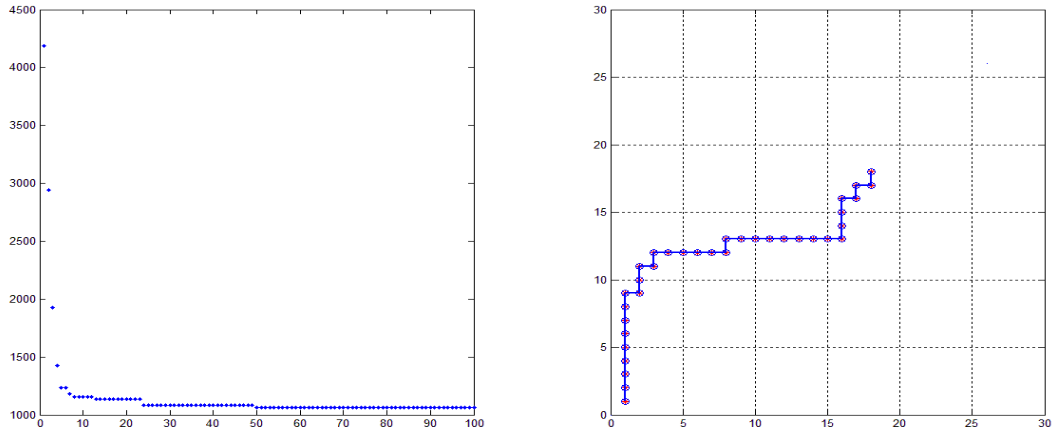


Figure 11. The Best Solution Found by Hybrid ACO with Mutation

4.5. Results of ABC

Table 7 shows the initialized parameter values of the ABC algorithm to find the best answer. Table 8 shows simulation results achieved by the ABC algorithm. Figure 12 shows the best routing for the 18 × 18 mesh topology for the ABC algorithm and its error reduction. Compared with previous algorithms, the ABC algorithm generated results of almost the same quality using less routing time.

Table 7. Setting the ABC Parameters

Parameter	Description
Population size	30
Number of scout bees	3
P_change_{max}	0.04
P_change_{min}	0.01
Level value	25

Table 8. Simulation Results for ABC

Topology	Source Node	Destination Node	Power Consumption	Bandwidth	Routing Time
Mesh 15 × 15	(1,1)	(15,15)	11.66	16.6	0.08
Mesh 18 × 18	(1,1)	(18,18)	14.21	20.11	0.1
Mesh 25 × 25	(1,1)	(25,25)	20.63	28.44	0.13

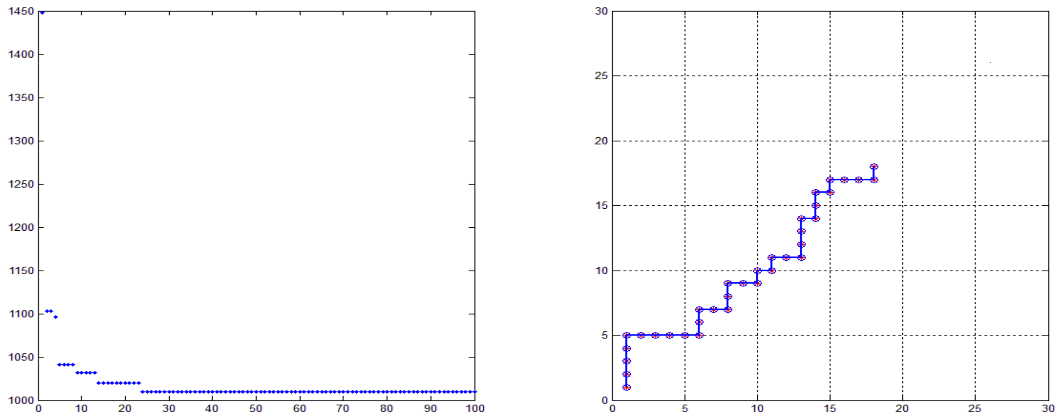


Figure 12. The best Solution Found by ABC

4.6. Results of XY Algorithm

Table 9 shows simulation results achieved by the XY algorithm. Figure 13 shows the best routing for the 18 × 18 mesh topology with the deterministic XY algorithm. It is evident that routing time in the case of deterministic routing is much lower than adaptive routing. As seen, deterministic routing of XY does not consider network congestion and first routes the packet in the X dimension followed by the Y dimension.

Table 9. Simulation Results for XY Algorithm

Topology	Source Node	Destination Node	Power Consumption	Bandwidth
Mesh 15 × 15	(1,1)	(15,15)	14.96	14.13
Mesh 18 × 18	(1,1)	(18,18)	19.91	18.31
Mesh 25 × 25	(1,1)	(25,25)	23.21	20.9

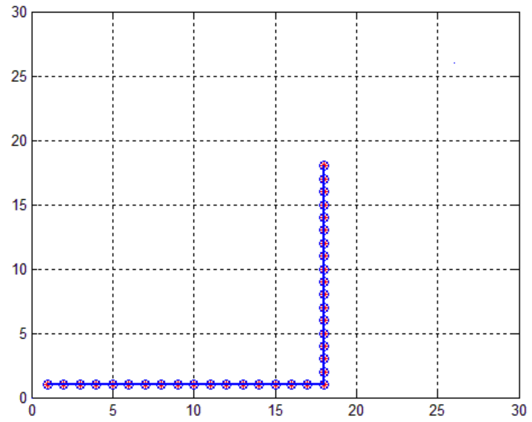


Figure 13. The best Solution Found by XY Algorithm

4.7. Discussion

Figures 14-16 compare the power consumption, bandwidth, and routing time of all proposed algorithms, respectively. All the adaptive algorithms performed better than the deterministic algorithm. Figures 15 and 16 show that all of the proposed heuristic algorithms had better results than the XY deterministic algorithm because these algorithms perform global searches in search space and choose the suitable solution according to the parameters defined in the cost function. The routing time for the XY algorithm in Figure 18 was not considered because this algorithm's routing time is about 1 ms because of the inherent determinism.

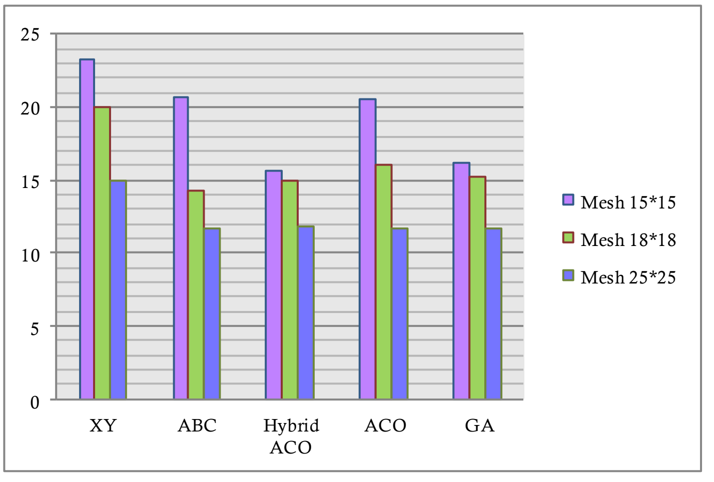


Figure 14. Power Consumption Chart

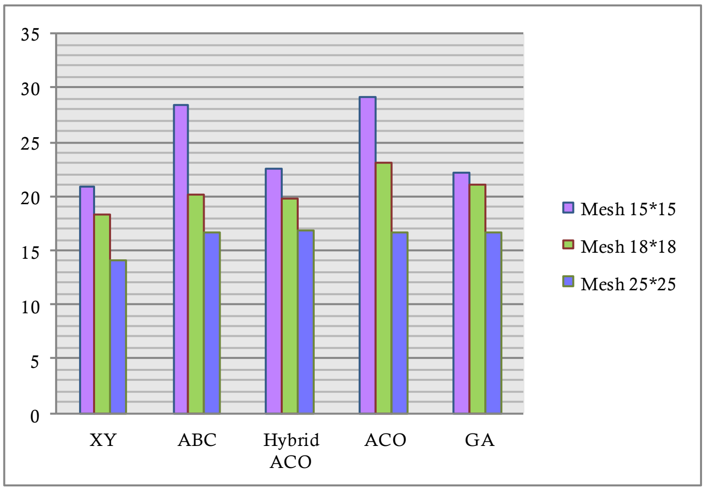


Figure 15. Bandwidth Chart

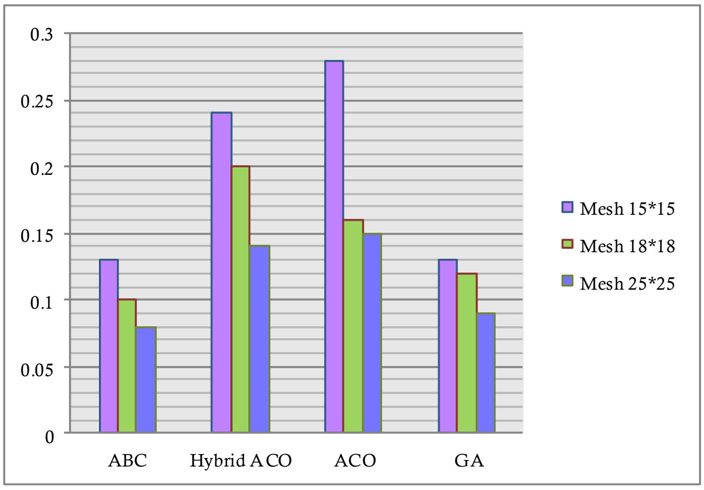


Figure 16. Routing-Time Chart

5. Conclusion

Choosing the proper algorithm for routing affects the transfer of packets in the network. In the present study, evolutionary algorithms for routing on NoC are presented. These algorithms perform better than other algorithms under different traffic conditions because they are customizable, problem independent, and use global search. Optimized routing strongly affects NoC performance. In the proposed algorithms, packets move through the shortest path to the destination node. The error reduction rate, power consumption, bandwidth, and routing were also investigated. The values obtained show that all evolutionary algorithms are adaptive in comparison with the deterministic XY algorithm and show better performance. Simulation results also show better performance for the proposed ACO algorithm with mutation over with the conventional ACO algorithm. It is evident that the artificial bee colony algorithm and the GA can adequately reduce faults in the cost function; however, the ABC algorithm was the most successful routing for time among all the algorithms.

References

[1] Z. Emrani and K. Mohammadi, "A Technique for NoC Routing Based on Extended Compact Genetic Optimization Algorithm", 19th Iranian Conference on Electrical Engineering (ICEE), **(2011)**, pp. 1-4.

[2] S. Shafaghi and R. Sabbaghi-Nadooshan, "Optimal Path Diagnosis by Genetic Algorithm for NoCs", International Journal of Smart Electrical Engineering, vol. 1, no. 2, **(2012)**, pp. 131-136.

[3] N. Roma, "Network-on-chip for Multicore Processors", MSc Thesis Proposal, technical University of Lisbon, Dept. of Electrical and Computer and computer Engineering, **(2013)**.

[4] A. V. de Mello, L. C. Ost, F. G. Moraes and N. L. V Calazans, "Evaluation of routing algorithms on mesh based NoCs, technical report series, 040, Faculdade Informatica Pucrs, Brazil, **(2004)**.

[5] M. Atagoziyev, "Routing algorithms for on chip networks", M. Sc. Thesis in electrical and electronics engineering, Middle East Technical University, **(2007)**.

[6] M. Li, Q. A. Zeng and W. B. Jone, "DyXY: a proximity congestion-aware deadlock-free dynamic routing method for network on chip", Proceedings of the 43rd annual Design Automation Conference ACM., **(2006)**, pp. 849-852.

[7] X. Chuan-pei, Y. Xiao-feng and C. Yu-qian, "A Technique for NoC Routing Based on Hybrid Particle Swarm Optimization Algorithm", In 3rd IEEE International Conference on Genetic and Evolutionary Computing WGEC'09., **(2009)**, pp. 607-610.

[8] H. K. Hsin, E. J. Chang, C. H. Chao and A. Y. Wu, "Regional ACO-based routing for load-balancing in NoC systems", In Nature and Biologically Inspired Computing (NaBIC), Second World Congress on IEEE, **(2010)**, pp. 370-376.

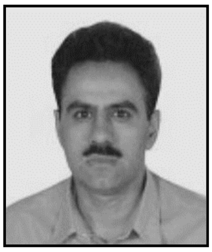
[9] E. J. Chang, C. H. Chao, K. Y. Jheng, H. K. Hsin and A. Y. Wu, "ACO-based Cascaded Adaptive Routing for traffic balancing in NoC systems", In Green Circuits and Systems (ICGCS) International Conference on IEEE, **(2010)**, pp. 317-322.

[10] C. Z. Janikow and Z. Michalewicz, "An Experimental Comparison of Binary and Floating Point Representations in Genetic Algorithms", Proceedings of the Fourth International Conference on Genetic Algorithms, **(2013)**, pp. 31-36.
[11] L. S. Junior, N. Nedjah and L. de Macedo Mourelle, "Routing for applications in NoC using ACO-based algorithms", Applied Soft Computing, **(2013)**, pp. 2224-2231.
[12] K. Y. Su, H. K. Hsin, E. J. Chang and A. Y. A. Wu, "ACO-Based Deadlock-Aware Fully-Adaptive Routing in Network-on-Chip Systems", In Signal Processing Systems (SiPS), IEEE Workshop, **(2012)**, pp. 209-214.
[13] L. S. Junior, N. Nedjah and L. de Macedo Mourelle, "ACO approach in Static Routing forNetwork-on-Chips with 3D Mesh Topology", Circuits and Systems (LASCAS) IEEE Fourth Latin American Symposium, **(2013)**, pp. 1-4.
[14] F. Ge and N. Wu, "Genetic Algorithm Based Mapping and Routing Approach for Network on Chip Architectures", Chinese Journal of Electronics, vol. 19, no. 1, **(2010)**.
[15] W. Hung, C. Addo-Quaye, T. Theocharides, Y. Xie, N. Vijaykrishnan and M. J. Irwin, "Thermal-Aware IP Virtualization and Placement for Networks-on-Chip Architecture", In Computer Design: VLSI in Computers and Processors, ICCD Proceedings IEEE International Conference, **(2004)**, pp. 430-437.
[16] M. Dorigo and T. Stützle, "Ant Colony Optimization", Massachusetts Institute of Technology, Hong Kong. Printed and bound in the United States of America, **(1996)**.
[17] N. Salehi, A. Dana and A. Khademzadeh, "Power Distribution in NoCs through a Fuzzy Based Selection Strategy for Adaptive Routing", In Digital System Design Architectures, Methods and Tools (DSD), 13th Euromicro Conference on IEEE, **(2010)**, pp. 45-52.
[18] S. Tosun, "New heuristic algorithms for energy aware application mapping and routing on mesh-based NoCs", Journal of Systems Architecture, vol. 57, no.1, **(2011)**, pp. 69-78.
[19] V. Chopra and A. Singh, "Solving FPGA Routing using Ant Colony Optimization with Minimum CPU Time", International Journal of Computer Science & Technology, vol.2, no.4, **(2011)**, pp. 223-226.
[20] M. Shokouhifar and A. Jalali, "An evolutionary-based methodology for symbolic simplification of analog circuits using genetic algorithm and simulated annealing", Expert Systems with Applications, vol. 42, no. 3, **(2015)**, pp. 1189-1201.
[21] D. Karaboga and C. Ozturk, "A novel clustering approach: Artificial Bee Colony (ABC) algorithm", Applied Soft Computing, vol. 11, no. 1, **(2011)**, pp. 652-657.
[22] M. Shokouhifar, A. Jalali and H. Torfehnejad, "Optimal routing in traveling salesman problem using artificial bee colony and simulated annealing", In 1st National Road ITS Congress, **(2015)**.
[23] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic", Evolutionary Computation CEC 99. Proceedings of the 1999 Congress, vol. 2, **(1999)**.
[24] M. Shokouhifar and A. Jalali, "A new evolutionary based application specific routing protocol for clustered wireless sensor networks", AEU-International Journal of Electronics and Communications, vol. 69, no. 1, **(2015)**, pp. 432-441.
[25] M. Shokouhifar and S. Sabet, "PMACO: A pheromone-mutation based ant colony optimization for traveling salesman problem", In 2012 International Symposium on Innovations in Intelligent Systems and Applications (INISTA), **(2012)**, pp. 1-5.
[26] M. Shokouhifar and S. Sabet, "A hybrid approach for effective feature selection using neural networks and artificial bee colony optimization", In 3rd international conference on machine vision (ICMV), **(2010)**, pp. 502-506.
[27] M. Shokouhifar and G. S. Abkenar, "An artificial bee colony optimization for MRI fuzzy segmentation of brain tissue", In 2011 International Conference on Management and Artificial Intelligence IPEDR, vol. 6, **(2011)**.
[28] M. Shokouhifar and A. Jalali, "Evolutionary based simplified symbolic PSRR analysis of analog integrated circuits", Analog Integrated Circuits and Signal Processing, vol. 86, no. 2, **(2016)**, pp. 189-205.
[29] M. Shokouhifar and A. Jalali, "Automatic symbolic simplification of analog circuits in MATLAB using ant colony optimization", In 22th Iranian conference on electrical engineering, **(2014)**, pp. 407-412.
[30] M. Shokouhifar and A. Jalali, "Automatic simplified symbolic analysis of analog circuits using modified nodal analysis and genetic algorithm", Journal of Circuits, Systems and Computers, vol. 24, no. 4, **(2015)**, pp. 1-20.
[31] M. Shokouhifar and A. Jalali, "Simplified symbolic gain, CMRR and PSRR analysis of analog amplifiers using simulated annealing", Journal of Circuits, Systems and Computers, vol. 25, no. 7, **(2016)**, pp. 1-23.
[32] Z. M. Zahedi, R. Akbari, M. Shokouhifar, F. Safaei and A. Jalali, "Swarm intelligence based fuzzy routing protocol for clustered wireless sensor networks", Expert Systems with Applications, vol. 55, **(2016)**, pp. 313-328.

Authors

Setareh Shafaghi received her B.Sc. degree in Electronic Engineering from Islamic Azad University of Lahijan, Iran, in 2009. She received her M.Sc. degrees in Electronic Engineering from Islamic Azad University of Central Tehran Branch, Tehran, Iran, in 2013. She is currently with SAMA University, Guilan, Iran. Her research interests include Swarm Intelligence Algorithms, Digital Image Processing and Digital Circuits.

Mohammad Shokouhifar received his B.S. from Islamic Azad University, Dezfoul, Iran, in 2008. He received two M.S. degrees from Islamic Azad University and Shahid Beheshti University, Tehran, Iran, in 2011 and 2013, respectively. He is currently a Ph.D. candidate in Electronic Engineering at Shahid Beheshti University, Tehran, Iran. His research interests include Symbolic Circuit Analysis, Wireless Sensor Networks, Fuzzy Logic and Swarm Intelligence Algorithms.



Reza Sabbaghi-Nadooshan is an associate professor and an academic member of Department of Technical and Engineering in Islamic Azad University of Central Tehran Branch, Tehran, Iran. His research interests include Networks, Hardware and Nano Computing.