ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Improvement of Genetic Algorithm for Optimizing Routing and Power Consumption in Network on Chip

F. Jalali[1,*], A. Ebrahimi[2], S. Alirezazadeh[3], F. Siar[4]

[1, 2] Faculty of Electrical Engineering, Sahand University of Technology (SUT), Tabriz, Iran
[3] Faculty of Computer Science, University of the East, Manila, Philippines
[4] Faculty of Electrical and Computer Engineering, Semnan University, Semnan, Iran

ARTICLE INFO

Article History:
Received 20 May 2020
Received in revised form
10 June 2020
Accepted 22 August 2020
Available online 29 August 2020

Keywords:
ACO; genetic algorithm; mapping; NoC; power consumption.

ABSTRACT

Network on Chip (NoC) is a new structure that has attracted attention of researchers in electronics. Performance of routing in network on chip is an important task. Two problems are considered in this paper. First problem is finding an appropriate order for responsibilities of task graph. Position of tasks should be found on communication kip. Second problem is communicating and finding appropriate direction for NoC. Appropriate direction is a direction that contains minimum distance in mapped tasks and uses network bandwidth optimally. It also has less power consumption. In this paper, a combination of genetic algorithm (GA) and ant colony optimization (ACO) is utilized to map cores on NoC optimally. Also proposed algorithm is applied for finding directions on chip. Results show that proposed algorithm can do routing task in NoC well and decrease power consumption.

1. INTRODUCTION

At first, system on chip (SoC) was appropriate for decreasing power consumption and size of systems. Through time, these systems were not responsive to increasing amount of information transmitted in the chip. The problem is solved by upcoming of network on chip (NoC) [1]. Researches on NoC are categorized into 4 fields including Communication infrastructures, communication infrastructure features, communication unit assessment, and mapping, which is focused in this paper.

The NOC architecture provides the communication infrastructure for resources. Optimizing power consumption is a critical issue in NoC. Delay of routing algorithms affects energy consumption in network. NoC consists of nodes (that refer to tasks) and links, which determines the network connectivity. By joining nodes in the networks, graphs are created. Nodes can be connected using different ways [2].

In NoC, there are many directions from one node to another. By applying routing algorithms, it is possible to find a good direction to the destination node.

* Corresponding Author : f_jalali@sut.ac.ir
Faculty of Electrical Engineering, Sahand University of Technology (SUT), Tabriz, Iran

http://dx.doi.org/10.47176/TMI.2020.14

2. TYPES OF MAPPING

According to the time devoted to process, mapping of tasks is divided into two categories: dynamic and static [3]. In dynamic mapping, allocation and assignment of tasks are done online through running the program [4]. This type of mapping always tries to perform the best operation and task distribution among processors. Dynamic mapping depends on capacity of processors. Computational efforts, delay, and power consumption is high in dynamic mapping.

In static state, mapping is done offline before tasks are executed. For a series of specified tasks and communication infrastructure, static mapping tries to perform the best task assignment, so that mapping is performed only once before the tasks are executed. Static mapping does not change during execution [2]. Static algorithms are divided into two groups: exact mapping and search-based mapping [5]. In exact static mapping, the goal is to find an optimal response based on mathematical programming. Mathematical model should provide a trade-off between runtime, processors, and communication costs to provide an optimal mapping.

Search-based mapping is classified to two categories including deterministic and heuristic mappings, based on the type of search. Branch-and-Bound (BB) search algorithms are examples of this type. The problem with this mapping is that search time grows exponentially with the size of the problem [6].

The heuristic search algorithm is categorized into two groups: constructive and transformative. In the constructive search algorithms, partial responses are created continuously, and ultimately, determined by a final mapping that may be improved with repetition or without improvement. In some cases, these techniques are faster than modified heuristic search techniques. On the other hand, the transformative search algorithms change the responses to gain the best one. Some of these algorithms are genetic algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) [4].

3. TYPES OF MAPPING

NoC is described with $n \times m$ graphs. In each graph, there is an IP and a switch. Parameters which are applied for transmitting bits on chips, are defined as follow:

1. Delay of i'th switch: the time allocated to transfer one bit from initial node to the destination node.

2. Power consumption of i'th switch: power usage in i'th switch for transferring one bit from initial node to the destination node.

3. Delay of i'th Link: delay of i'th link for transferring one bit.

4. Power of i'th Link: power consumption in i'th link for transferring one bit.

5. Number of bits: number of transferred bits from initial node to the destination node.

4. PREVIOUS WORKS

Mapping is known as one of the most complex computational problems, for which innovative solutions have been considered.

Lei et al. [7] used a two-stage genetic algorithm for mapping tasks to NoC and reducing the runtime of the program. In the first step, tasks are assigned to different IPs, with the assumption that the delays are constant and equal to the average latency. Secondly, the IPs are mapped to the chips by considering the actual delay based on the network traffic model, which minimizes the system's total latency. In their mapping strategy, latency factors such as the possibility of sending messages by cores (IPs), packet lengths, and network connections are not considered.

Zhou et al. [8] considered a delayed model for mapping in NoC that took all of factors into account. Their model could map tasks to NoC and optimize the average delay. They considered a population corresponding to the position of the cores for the NoC topology (in genetic algorithm). The initial population was selected randomly and their target function was waiting time. To create a new population, a multi-point intersection with random selection of points is used. Chromosomes with a lower waiting time have more chances to be selected at the intersection. The size of the chromosomes is equal to the number of core in graph. To control the convergence rate, mutations were utilized. This process is repeated until the lowest average waiting time is reached. Best response shows location of cores in NoC.

Catania et al. [9] performed a mapping based on genetic algorithm in NoC, optimized their multi-objective function, and evaluated their performance in terms of power consumption. They run multi-objective searches based on mesh in the network on the chip. Their method helps to obtain the Pareto mapping optimally and accurately and reduces power consumption in the network.

Robindra et al. [10] proposed a genetic algorithm that minimized energy consumption by reducing the number of switching links between the cores. In their research, the IP mapping problem is solved using NoC and mesh. Their proposed method saves more than 15% of the bandwidth energy.

Also, Chadari et al. [11] presented a genetic algorithm based on mapping techniques for reducing communication energy in order to optimize NoC. In their work, genetic algorithm has been used to combine NoC to link with different bandwidths. Optimization goals were to enhance communication and network distribution. Results show that using the grid on an irregular chip performs better than mesh-based methods.

Sepelida et al. [12] considered a multi-purpose matching function that combines a set of features to use the solutions diverse populations, improve local search, and prevent early convergence. In their research, power consumption and latency are considered as multi-purpose objective function. Results show that the use of the MAIA multipurpose function optimizes power consumption and cloaking greatly. Nandhini and Paulene [13] proposed the notion of network information area, which denotes the network information used by each routing method while also incorporating the backward ant mechanism. They assess each linked work using NIR before proposing the ACO-based adaptive routing with pheromone diffusion (ACO-PhD) algorithm.

In addition, the research [14] employs ant colony optimization for the realization of the NoC map in terms of power consumption and delay potential. Ant colony methods were utilized in [15] to locate and optimize paths in a mesh-based NoC with many randomly generated applications mapped. The goal of routing optimization is to reduce total latency in packet delivery between jobs. By comparing it to general purpose algorithms for deadlock free routing, simulation results demonstrated the usefulness of the ant colony inspired routing. Also covered in [16] are the many obstacles and potential solutions for creating energy-efficient many-core circuits allowed by 3D integration.

5. PROPOSED METHOD

In this paper, routing in NoC is improved using a combination of ant colony optimization and genetic algorithm. Fig. 1 shows diagram of proposed method. Ant colony optimization is utilized to create initial population for genetic algorithm.

Ants in ACO algorithm, search in a discrete space, which shows 2D topology of NoC (see Fig. 2 for 2D NoC structure using x-y routing strategy). They go from one node in NoC graph to another and have memory to save directions. Directions between the initial node and the destination node are marked. Pheromone is put in the direction between the initial and the destination nodes.

Pheromone density is more in shorter directions. This feature ultimately leads to choose directions with more pheromones and less length by ants. At first, one node is selected randomly and all ants are moved from this initial node to the destination node. Here, the aim is to move through all nodes in the network with less delay, shorter path, and most speed. Ants move based on the probability between two directions i and j. They select the direction with most probability. (1) shows the probability rule in ACO algorithm.

$$P_{ij}^{k}(t) = \frac{[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}}{\sum_{h \notin tabu_{k}'}[\tau_{ij}]^{\alpha}.[\eta_{ij}]^{\beta}} . \quad if\, j \in allowed\ k \tag{1}$$

In (1) when $j \notin$ allowed k, the probability is considered zero. τ_{ij} shows the amount of pheromone in the direction i and j between initial and destination nodes. η_{ij} represents probability of observing this direction by ants. α and β show Learning coefficients. For these parameters values greater than zero are accepted. The larger α makes to choose a direction that is observed by more ants. Choosing small values for α transforms the algorithm into a greedy one. The directions, selected by ant k, is put in the tabu list. Probability of viewing the direction is calculated for all k ants. The simplest way to calculate this probability is Euclidean distance using (2), in which d_{ij} represents the distance between the two directions i and j.

$$\eta_{ij} = \frac{1}{d_{ij}}. \tag{2}$$

At the end of each iteration, the pheromone of the direction is updated according to (3).

$$\tau_{ij}(new) = (1-\rho)\tau_{ij}(old) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}. \tag{3}$$

In (3), ρ shows the evaporation coefficient of the pheromone and the value of it is between 0 and 1. It can be referred to degree of information reducing with time lapse. Parameter m is the colonial size (the number of ants). $\Delta\tau_{ij}$ shows the pheromone created between directions i and j and is defined according to (4). In each movement of all ants, cost function is calculated using distance. Lower distance makes the cost function minimum that means delay and power consumption are low and optimum.

$$\Delta\tau_{i,j} = \frac{Q}{L_k}. \quad \textit{if (i, j)} \in \textit{the direction described by the ant} \tag{4}$$

In (4) when (i, j) ∉ the direction described by the ant, the probability is considered zero. Where L_k represents the cost of the path observed by the ant k and Q is a constant number greater than zero.

According to the algorithm, the type of switching in this problem is as follow: first, one node (which refers to a task) is randomly selected from NoC graph, that is a space with size n by m. The first selected node has eight nodes in its neighbourhood. The second node is selected from these eight state space neighbourhood nodes in the second stage. If the first node is selected at corners, the second node's state space neighbourhood consists of 5 or 3 nodes, depending on the location. Selecting other nodes follows the same way. In the next step, the distances of these nodes are calculated either directly or in a diagonal state. Distances are summed together and the optimal state space is selected. This cycle is repeated and n × m optimized state space is mutated using the combination ACO and genetic algorithm.

The genetic mapping algorithm is used for mapping the cores onto the mesh. The structure of the chromosomes in this algorithm is similar to the array of switches in the lattice connectivity. Thus, we have a two-dimensional structure for chromosomes. After generating a first-generation randomly, generations of genetic operators (based on the two-dimensional structure) are used to create subsequent generations.

The amount of each gene in the chromosome represents a core of communication graph. The population of this gene is used directly for the next generation, and genes that are more than 0.5 have been transmitted directly from the first parent to the child, and the remaining genes of this child are inherited from the second parent. In this way, the genes, transferred from the first parent to the child, are eliminated in the second parent. Then, the remaining genes in the second parent are transferred to their empty houses in a row from the first child's position (1, 1).

For the second child, only the role of the first and second parents will change with each other. After integration, mutation is performed in order to avoid stopping the algorithm in local responses. A similar condition is also used for mutation, which is likely to be 0.5. In mutation, two genes are randomly selected and replaced with each other.

To create a new generation, if the children are better than their parents, based on calculation, they will be replaced in the population, and the cycle of assessment and generation of the new population will be re-established in the new population to achieve convergence.

6. RESULTS

In this work, ACO and genetic algorithm are combined to improve routing in NoC. The ants move from the initial node to the destination node in NoC. In this step, the aim is to traverse all nodes to find an optimum direction between the initial and destination nodes. The selected direction should contain the least delay, which means the shortest distance and the fastest switching happens.

In our tests, five ants are initially considered (k = 5), so five directions are created. Then the cost function, which is based on distance, is calculated. In ACO algorithm, parameters are $\alpha = 1$, $\beta = 1$, $\rho = 0.05$, and number of nodes in 2D NoC topology is considered 20. Parameters in genetic algorithm are set as follow:

1. Population size = 5 (it is equal to number of ants and obtained directions from ants).
2. Crossover Percentage = 0.8.
3. Mutation Percentage = 0.3.

If the distance is short, the delay time and energy consumption are low and all parameters are optimized. In each iteration of ACO, best responses (that are short directions) are considered as the primary population in the genetic algorithm. The infants will be acquired to the initial population of crossover and mutation. Then, the answers are sorted in the genetic algorithm and the optimal answers are selected. Pheromones are updated at each stage after applying genetic algorithm. This process continues according to number of iterations which is considered 500 in our tests.

In this research, primary population production for genetic algorithm is very important. In NoC, methods with high computational efforts are used to generate primary population, which decrease the speed of the algorithm. The advantage of using ACO to create primary population for genetic algorithm is that the answers are obtained quickly. The ant first moves in order to produce the primary direction and population. By acquiring the primary population, sorting and calculating the genetic algorithm loop, cost is improved. Fig. 3 shows how the move is done to reach the optimal direction. Two nodes have been selected. The ant proceeds different destinations and reaches the destination node.

Fig. 4 and Fig. 5, respectively, show the best cost and delay diagrams in 500 repetitions. The fifth diagram (see Fig. 5) is the delay time that is directly related to the energy consumption. In other words, the delay time and energy consumption graphs are the same. As shown in Fig. 4, the best cost is obtained for the first repetition and it gradually reaches a stable and optimal value which is fixed at 20. In combination of ACO and genetic algorithm, it is concluded that the higher the number of repetitions, the better and the more correct answers are obtained. According to the results, input parameters play an important role in reaching the optimal answer. Reducing the runtime of the algorithm and increasing the accuracy can be obtained by selecting appropriate range for input parameters.

Fig. 6 shows our experiments with different total iterations. Results show that choosing large values for regulation of the pheromone intensity results in rapid convergence toward the optimal overall solution, while low values of this parameter may lead to program divergence. Additionally, selecting the appropriate amount for updating the pheromone matrix is done according to the overall cost of the network, which can be achieved with several test to run into its appropriate range. On the other hand, the more ants in the network, the more accurate the answers will be. The number of iterations plays an important role in solving the problem. Low number of iterations will reduce the accuracy and certainty of the answer. High number of iterations makes the algorithm more accurate, but slow. So, there should be a trade-off between accuracy and speed in choosing appropriate total number of iterations. Also, Table 1 shows comparison of different number of nodes in reaching convergence in proposed algorithm. Second column in Table 1 represents the average iteration that is needed to gain convergence in proposed ACO-GA algorithm. Third column shows average of cost function in ACO algorithm which is a distance-based parameter. The nodes refer to assigned task in the 2D NoC structure.

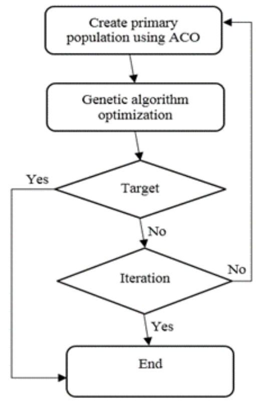


Fig. 1. Diagram of proposed method for routing in NoC.

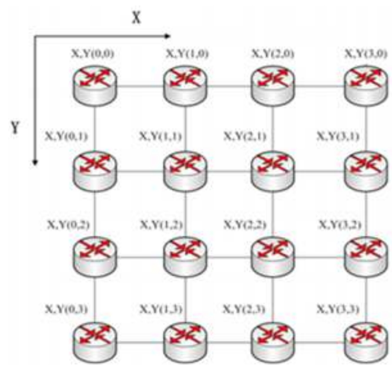


Fig. 2. An example of 2D NoC topology [16].

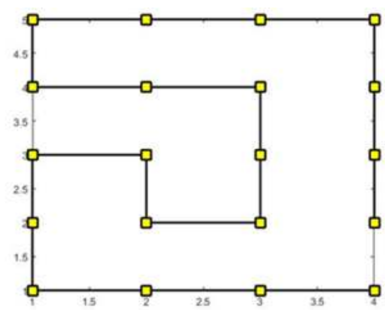


Fig. 3. Finding the best direction by the ant in 2D graph of NoC.

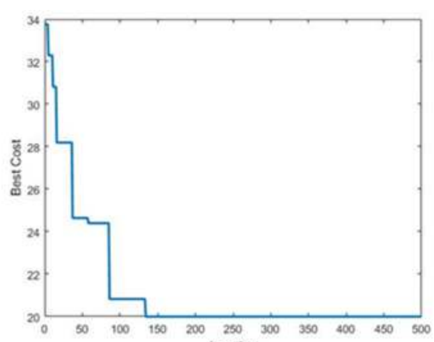


Fig. 4. Diagram of best cost in one run of proposed algorithm with 500 iterations. The vertical axis demonstrates best cost and the horizontal axis shows iteration number (number of nodes = 20).

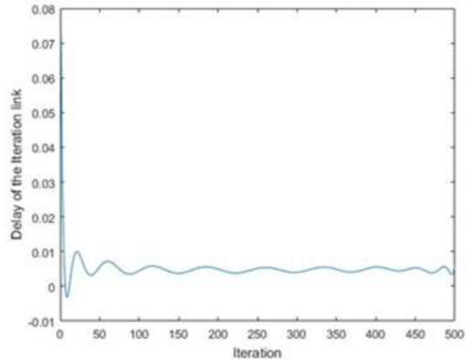
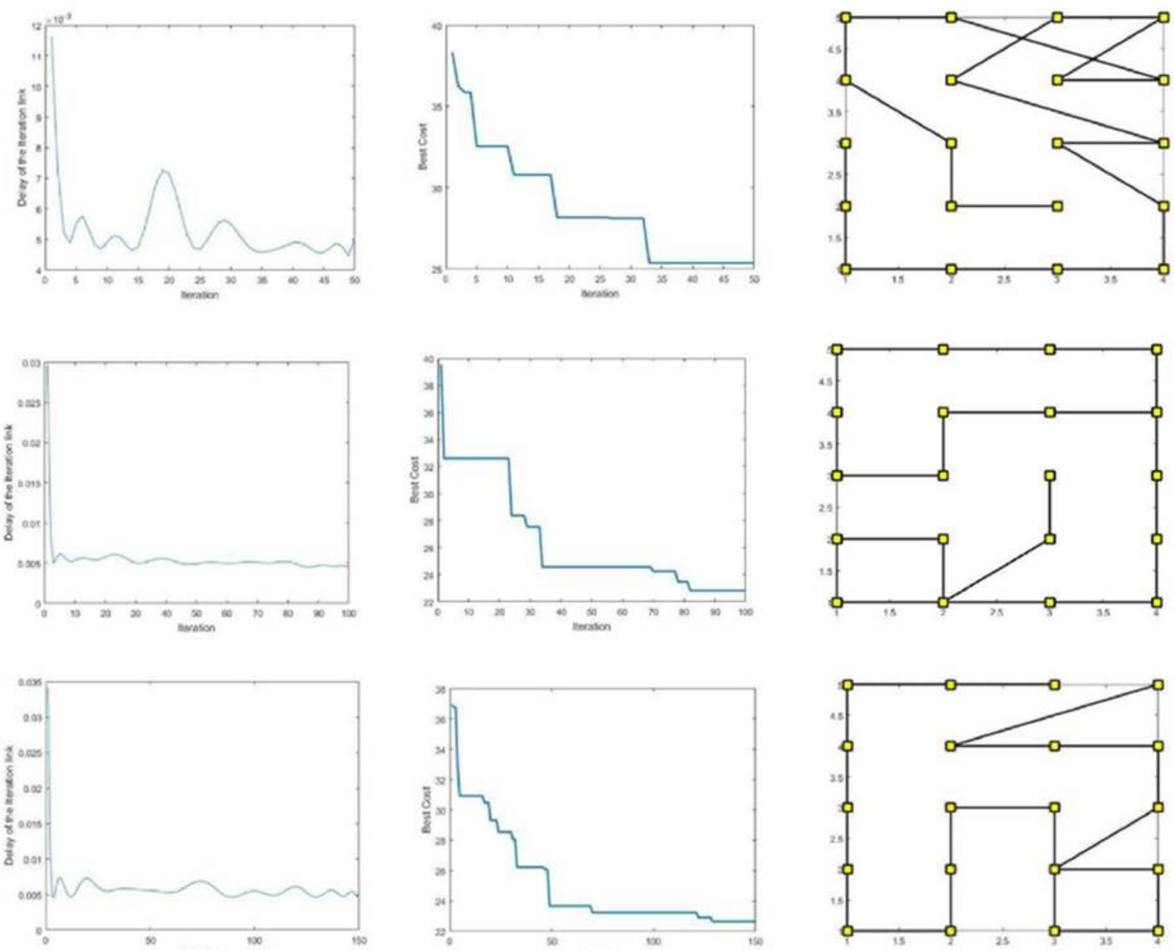


Fig. 5. Diagram of delay of iteration link in one run of the proposed algorithm with 500 iterations. The vertical axis demonstrates delay of iteration link and the horizontal axis shows iteration number (total iterations = 500, number of nodes = 20).



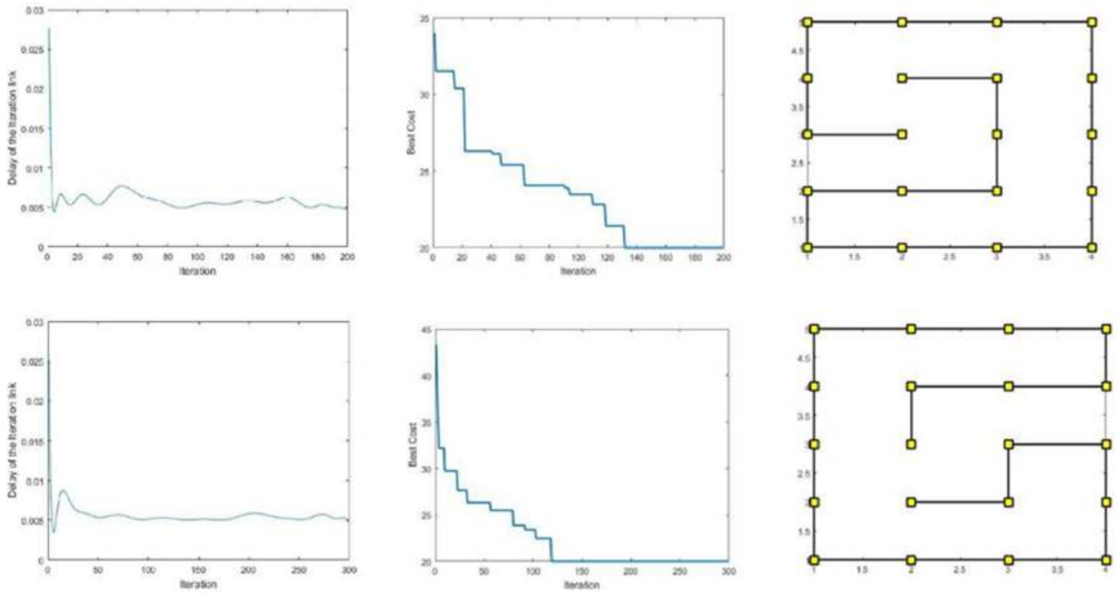


Fig. 6. The experiments with different total iterations. Values of total iteration from top row to the bottom are 50, 100, 150, 200, and 300 respectively. In each row first diagram (from left to right) shows delay of iteration link. Second diagram represents the best cost. Third diagram shows movement of ant and detected optimum direction. Number of nodes is 20, which refers to assigned tasks in NoC.

Table 1. Comparison of different number of nodes in reaching convergence in proposed algorithm. Second column shows the average iteration that is needed to gain convergence in proposed algorithm. Third column shows average of cost function in ACO algorithm which is a distance-base parameter.

Number of Nodes (Tasks in NoC)	Mean Iteration	Mean Cost in ACO
6	5	6.0000
10	40.8000	10.0000
16	182.2000	16.8284
20	248.8000	20.0000
26	280.0000	26.9430
30	400.4000	30.9073
36	354.4000	38.2173
40	419.8000	41.7889
46	375.8000	50.6556
50	423.6000	55.0048
56	341.0000	62.9488
60	393.2000	68.9491
66	362.6000	76.5965
70	344.8000	82.2494
76	455.4000	89.2141
80	372.8000	94.5180
86	311.8000	103.0545

7. CONCLUSIONS

In this paper, a combination of genetic algorithm and ACO is applied for improving power consumption and routing in NoC. A condition is used on genetic algorithm to stop it when there is no optimization in answer. In proposed method, ACO is utilized to create primary population for genetic algorithm. This process decreases power consumption and delay in NoC and increases speed of finding the optimum direction.

REFERENCES

[1] Bjerregaard, T., & Mahadevan, S. (2006). A survey of research and practices of Network-on-chip. ACM Computing Surveys, 38(1), 1. doi:10.1145/1132952.1132953

[2] Sahu, P. K., & Chattopadhyay, S. (2013). A survey on application mapping strategies for Network-on-Chip design. Journal of Systems Architecture, 59(1), 60–76. doi:10.1016/j.sysarc.2012.10.004

[3] Mandelli, M., Amory, A., Ost, L., & Moraes, F. G. (2011). Multi-task dynamic mapping onto NoC-based MPSoCs. Proceedings of the 24th symposium on Integrated circuits and systems design - SBCCI '11. Brazil. doi:10.1145/2020876.2020920

[4] Weichslgartner, A., Wildermann, S., & Teich, J. (2011). Dynamic decentralized mapping of tree-structured applications on NoC architectures. Proceedings of the Fifth ACM/IEEE International Symposium on Networks-on-Chip - NOCS '11. Pittsburgh, Pennsylvania. doi:10.1145/1999946.1999979

[5] Ghosh, P., Sen, A., & Hall, A. (2009). Energy efficient application mapping to NoC processing elements operating at multiple voltage levels. 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip. La Jolla, CA, USA. doi:10.1109/nocs.2009.5071448

[6] Hu, J., & Marculescu, R. (2005). Energy-and performance-aware mapping for regular NoC architectures. IEEE Transactions on computer-aided design of integrated circuits and systems, 24(4), 551–562.

[7] Lei, T., & Kumar, S. (2003). A two-step genetic algorithm for mapping task graphs to a network on chip architecture. Euromicro Symposium on Digital System Design, 2003. Proceedings. Belek-Antalya, Turkey. doi:10.1109/dsd.2003.1231923

[8] Zhou, W., Zhang, Y., & Mao, Z. (2006). An application specific NoC mapping for optimized delay. In Design and Test of Integrated Systems in Nanoscale Technology. DTIS 2006. International Conference on (pp. 184–188). IEEE.

[9] Ascia, G., Catania, V., & Palesi, M. (2004). Multi-objective mapping for mesh-based NoC architectures. Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis - CODES+ISSS '04. Stockholm, Sweden. doi:10.1145/1016720.1016765

[10] Jena, R. K., & Sharma, G. K. (2007). A multiobjective evolutionary algorithm-based optimisation model for network on chip synthesis. International journal of innovative computing and applications, 1(2), 121. doi:10.1504/ijica.2007.016793

[11] Choudhary, N., Gaur, M. S., Laxmi, V., & Singh, V. (2011). GA based congestion aware topology generation for application specific NoC. 2011 Sixth IEEE International Symposium on Electronic Design, Test and Application. Queenstown, New Zealand. doi:10.1109/delta.2011.26

[12] Sepúlveda, M. J., Chau, W. J., Gogniat, G., & Strum, M. (2012). A multi-objective adaptive immune algorithm for multi-application NoC mapping. Analog Integrated Circuits and Signal Processing, 73(3), 851–860. doi:10.1007/s10470-012-9869-9

[13] Huida, L., Bingchen, F., Liheng, Y., & Yanan, L. (2012). RFID-based hazardous waste management platform establishment. Procedia Engineering, 29, 4–8. doi:10.1016/j.proeng.2011.12.658

[14] He, T., & Guo, Y. (2013). Power consumption optimization and delay based on ant colony algorithm in network-on-chip. Engineering Review: Međunarodni časopis namijenjen publiciranju originalnih istraživanja s aspekta analize konstrukcija, materijala i novih tehnologija u području strojarstva, brodogradnje, temeljnih tehničkih znanosti, elektrotehnike, računarstva i građevinarstva, 33(3), 219-225.

[15] Junior, L. S., Nedjah, N., & De Macedo Mourelle, L. (2013). Routing for applications in NoC using ACO-based algorithms. Applied Soft Computing, 13(5), 2224–2231.

[16] Achballah, A. B., Othman, S. B., & Saoud, S. B. (2017). Problems and challenges of emerging technology networks− on− chip: A review. Microprocessors and Microsystems, 53, 1–20.