Bulletin of Electrical Engineering and Informatics
Vol. 9, No. 5, October 2020, pp. 2170~2177
ISSN: 2302-9285, DOI: 10.11591/eei.v9i5.2268

An improved ant colony optimization algorithm for wire optimization

Ho Joon Jim, Fazida Hanim Hashim
Department of Electrical, Electronic and Systems Engineering, National University of Malaysia, Malaysia

Article Info

Article history:

Received Dec 28, 2019
Revised Mar 7, 2020
Accepted Apr 15, 2020

Keywords:

Ant colony optimization
Circuit routing
Heuristic algorithm
Steiner nodes
Wire optimization

ABSTRACT

Wire optimization has become one of the greatest challenges in today's circuit design. This paper presents a method for wire optimization in circuit routing using an improved ant colony optimization with Steiner nodes (ACOSN) algorithm. Circuit delay and power dissipation are primarily affected by the length of the routed wire. Thus, the main goal of this proposed algorithm is to find the shortest route from one point to another using an algorithm that relies on the artificial behavior of ants. The algorithm is implemented in the JAVA programming language. The proposed ACOSN algorithm is compared with the conventional ant colony optimization (ACO) algorithm in terms of efficiency and routing performance when applied to three types of circuits: emitter-coupled logic, 741 output and a cascode amplifier. The performance of the proposed method is analyzed based on circuit information such as total wire routing, total number of nets, total wire reduction, terminals per net and total terminals. From the simulation analysis, it is shown that the proposed ACOSN algorithm gives the most benefit to complex circuits, where it successfully reduces the wire length by 21.52% for a cascode amplifier circuit, 14.49% for a 741 output circuit, and 10.43% for emitter-coupled logic circuit.

Corresponding Author:

Fazida Hanim Hashim,
Department of Electrical, Electronic and Systems Engineering,
National University of Malaysia,
43600 Bandar Baru Bangi, Selangor, Malaysia.
Email: fazida@ukm.edu.my

1. INTRODUCTION

In recent years, power dissipation has become a primary concern for high speed and high integration circuits. A catch-phrase "high speed, low power" is the most desirable attribute of a microprocessor performance. The trend in the past decades shows tremendous increase in power dissipation. According to the trend chart of Intel Pentium processor [1], the power dissipation increases two-fold every 4 years, similar to Moore's Law. This is due to among others, the increase in the length and number of global and local signaling wires, owing to the growth in circuit complexity, area expansion, and shrinkage in modules [2].

Countless techniques, both proactive and reactive, have been proposed to address this issue. Proactive technique focuses on capitalizing the design stage such as the circuit layout by reducing transition and switching of signals, minimizing capacitance, compression, mass data storage, utilizing clock cycle, high bit-rate digital subscriber loops, etc [3]. Reactive measures look into curbing the heat dissipation by employing advanced packaging material, heat sink and fans.

Consequently, one of the techniques to curb power dissipation is through circuit routing in the design stage, where the routing in the net should be kept as minimum as possible. Due to the high component

density in integrated circuits today, it is imperative to aid the task with the help of a design tool [4]. Previous studies have shown commendable improvements in circuit wiring optimization. Some of these studies besides aiming for minimum wiring length and/or area, perform comparison on other circuit parameters such as routing time [4-13], number of terminals [4, 8, 12], number of channel tracks [4], number of nets [10], etc. Since the advent of optimization in circuit routing, many studies have adopted heuristic approaches for circuit wiring optimization such as the ant colony optimization (ACO) algorithm [5, 7-9] and particle swarm optimization (PSO) [6]. ACO being a robust algorithm, is also adopted in optimization of wireless network routings [13] among other bio-inspired algorithms [14, 15], as well as in various optimization problems [16-21]. Others have adopted the non-heuristic approach such as the left-edge algorithm [7], shortest-path-search algorithm [10], compact metal routing algorithm [11], integer linear programming [12], tree-based algorithm [13] and Steiner minimum tree routing [22]. This study adopts the conventional method of reducing total wire length by invoking the ACO algorithm with further enhancement of the method by insertion of the Steiner nodes.

2. NETWORK TOPOLOGY

2.1. Basic circuit topology

According to [6], a typical circuit routing could be classified into three types of grid routing: uniform grid routing, non-uniform grid routing and grid-free routing. This study focuses on uniform grid routing applications using the Hanan grid. Figure 1 shows an example of an inverter and a diode in a Hanan grid with the suggested optimized routing path by the "ant". Q2, Q1 and D1 are called instances (or components) and under each instances, connections are made to other instances which are called terminals. Each metal-oxide-semiconductor field-effect transistor (MOSFET) has three terminals while the diode has two. Each terminal has its own designated identification and coordinates. In Figure 1, Q2 is identified by three terminals: Q2.1, Q2.2 and Q2.3. Each terminal is associated with a coordinate, for example, terminal Q2.3 is at coordinate (3,6) on the Hanan grid. When all of the instances and terminals have been defined, the ant starts to route a particular node, sometimes referred to as a net list. The dashed connection from Q2.2 to Q1.2 to D1.1 shows the best routed node optimized by the ant. In this paper, the method of routing the nodes is implemented by either of the two algorithms; conventional ACO routing or ACO with Steiner nodes (ACOSN) routing. All the topographical information is user defined and it is stored in the input file. The ant will then predict the shortest route based on the two earlier mentioned algorithms and it will mark the empty node each time it passes one. The marked node will be updated in a topology so that it will be visible to the next subsequent routing ants.

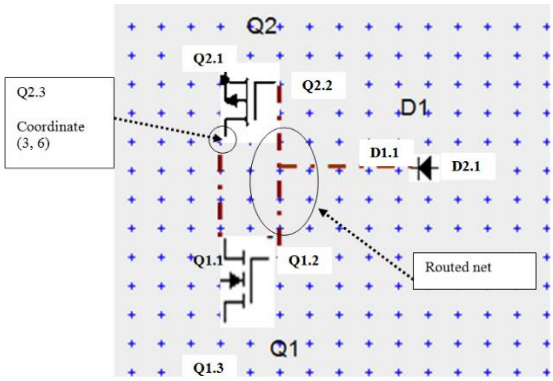

Figure 1. An inverter routed on a hanan grid

2.2. Algorithm topology

There are two types of topologies that could be solved by the algorithm which are line topology and network topology. Figure 2 depicts the two different topologies. Line topology is a basic topology which routes a wire from point A to B, where the number of terminals, *n*, is equal to 2. Network topology on the other hand is a complex topology which has many branches, and *n* greater than 2 (*n>2*). Network topology can be created by ACO algorithms by combining multiple line topologies and controlling it in a loop format.

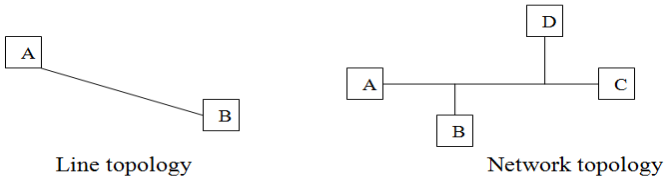

Figure 2. Algorithm topologies

3. RESEARCH METHODS

The point-to-point (P2P) algorithm is primarily used to route line topology while ACO and ACOSN are used to route network topology. Since network topology is actually a combination of multiple line topologies, both ACOSN and ACO exploit routing line topology by invoking the P2P algorithm.

3.1. Point-to-point algorithm (P2P)

The P2P algorithm is the basic sub-algorithm to create both the conventional ACO algorithm and the ACO algorithm with Steiner nodes (ACOSN). The only purpose of this algorithm is to move the ants from point A to point B. If there are more than just point A to point B routing, a network topology needs to be constructed. This algorithm is modified from a research done by Tamana Arora [3]. Figure 3 shows the sub-algorithm of the P2P algorithm.

```
POINT-TO-POINT SUB-ALGORITHM
1.For (every current location) {
2. Deploy all ants at the neighbouring nodes
3. Lay pheromone
4. Determine desirability of each neighbouring nodes
5 Choose best candidates from neighbouring nodes
6. Move the ant to the best candidate node
7. Evaporate pheromone /mark visited node
8. If (current location== destination node) {
9. Terminate at that node }
10. } end
```

Figure 3. Point-to-point sub-algorithm

3.2. Conventional aco algorithm (ACO)

Since conventional ACO algorithm is designed to handle network topology, it exploits P2P algorithm to find the shortest route. Depending on the number of terminals, the ant will route the first path and that existing path acts as an array of destination candidates instead of one static destination so that subsequent ants will "sense" which destination is nearer before invoking a P2P algorithm. In this case the destination coordinates along the route is not fixed by the user but will be determined by the ants. Figure 4 shows the algorithm for a conventional ACO routing system.

```
CONVENTIONAL ACO ALGORITHM
1. Initialize pheromone.
2. Identify all terminals in net list.
3. If(first route) {
4. Invoke point-to-point algorithm
5. Mark visited route }
6. If(route >first) {
7. While(n terminals not connected) {
8. Identify new source terminal
9. From marked route, sense suitable candidate for destination
terminal
10. Choose a destination node
11. Invoke point-to-point algorithm
12. Mark visited route }
13. } end
```

Figure 4. Conventional aco algorithm

3.3. Aco Algorithm with steiner nodes (ASN)

Like its predecessor, ACOSN is a variation of the ACO algorithm which invokes a special algorithm to find the extra nodes to optimize the wire routing. Figure 5 shows the algorithm for an ACO routing system with Steiner nodes. To find the Steiner node, polygon optimization has to be done to select the encroachment area which acts as a pool of potential Steiner candidates. Having done this, all the candidates of the Steiner node could be identified and the desirability of each candidate will be calculated. The Steiner candidate which yields the highest desirability will be chosen as the Steiner node. This methodology employs bubble shrinking heuristics [23], 1-Steiner Algorithm [24] and clustered Hanan [25].

ACO ALGORITHM WITH STEINER NODES

1. Initialize pheromone.
2. Identify all terminals in net list.
3. Polygon optimization
4. Calculate all intermediate nodes desirability by the Hanan Weight
5. Select the candidate with best Hanan Weight as the Steiner node.
6. If(> 1 best candidates/ or in cluster) {
7. Find 2 Steiner points in the cluster by invoking CAA sub-algorithm to find destination node from given terminals
8. Connect a bridge between 2 Steiner points
9. Connect remaining terminals to the bridge using P2P }
10. Else {
11. While(terminals not covered) {
12. invoke point-to-point algorithm
13. Marked visited nodes }
14. } end

Figure 5. Aco algorithm with steiner nodes

4. RESULTS AND DISCUSSION

The simulations were done using JAVA programming language and conducted on a computer running Pentium core I-5 460 MHz processor with 4 GB of RAM. A Microsoft Excel comma separated value (CSV) file is generated to store the data from the circuits. The proposed algorithm and its counterpart were tested on three different circuits. Each circuit has variations in attributes such as total wire length, total terminals, and total nets. Since the only difference between ACOSN and ACO is the improvement in terms of network topology (for $n>2$), the main focus of this simulation is to compare the effectiveness of ACOSN to its counterpart ACO, in weeding out any extra length of network topology.

4.1. Preliminary results

Each individual circuit topology was tested before simulating the entire circuit. The topologies can be categorized as:

- Typical linear topology (n=2)
- Complex linear topology (n=3, n=4)
- Typical network topology (n=3)
- Complex network topology (n=4)

When comparing the performance of ACOSN and ACO for typical linear topology, there is no difference in terms of performance due the fact that both algorithms share the same point-to-point sub-algorithm to route the line topology. The same holds true for complex linear topology. Since the linear topology have all nodes connected in a line, no added Steiner points are necessary and no improvement could be made by ACOSN over ACO.

However, for typical network topology, there is an added advantage for using ACOSN over ACO. Figure 6 shows an example of this topology where an extra Steiner point is added by the ACOSN algorithm as a shortcut path between A, B and C (ACOSN and ACO co-exists in the same graph). This results in the cost of ACOSN to be reduced to 24 units compared to ACO which is 28 units. This gives a total of 14.28% of length improvement.

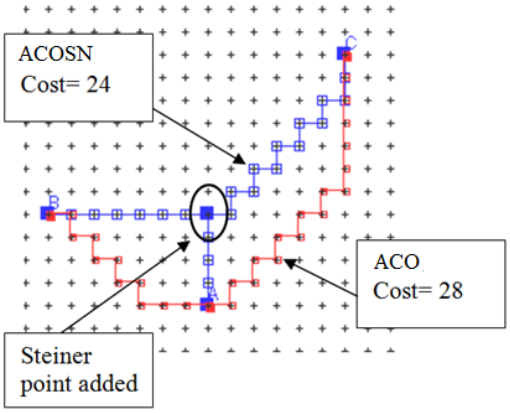

Figure 6. Typical network topology (*n*=3)

Finally, complex network topology gains the most benefit from ACOSN, owing to the capability of utilizing 1-Steiner algorithm to add two extra Steiner points. Maximum cost savings are expected to be observed from this topology. From Figure 7 below, the total cost of ACOSN is 23 units while ACO is 29 units, giving a great improvement of 20.68%. Table 1 summarizes the percentage of improvements of all three network topologies. From the table, it is shown that the proposed ACOSN gives the greatest improvement to the network with the most complex topology.

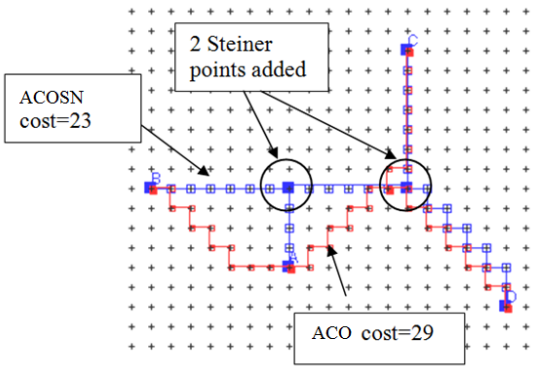

Figure 7. Complex network topology (*n*=4)

Table 1. Relationshipbetween topology and potential length improvement

Topology	*n*	Improvement (%)
Linear	2	0
Network	3	14.28
Network	4	20.69

4.2. Final results

The simulations conducted in this study focus on three key areas of improvement which are:

- Average improvement by terminal per net.
- Total length improvement versus composition of terminal per net.
- Total length improvement versus coverage of net (for *n>2*).

The first key improvement in (a) of the simulation gives us the likelihood of average length improvement or cost savings of net (categorized by *n*) as shown in Table 1, based on the data of three different circuits. Having done this, we would like to know how effective is the contribution of net *n=3* or *n=4* in reducing the overall circuit length by first knowing the quantity as depicted in (b) and quality as quantified by its original wire length versus the entire circuit route shown in (c). Preliminary simulations

were done on the individual topology rather than the whole circuit. Table 1 shows the improvement of ACOSN versus ACO by taking the samples of a few topologies. It is seen that there is a 14.28% potential of length improvement for network topology with *n=2*, and a 20.69% potential of length improvement for *n=3*.

4.3. Average improvement by terminal per net

From the simulations, the total wire lengths covered by the ACOSN and ACO algorithm are computed. The total length is classified by nets having different number of terminals, *n*. The graph in Figure 8 shows the overall improvement. As expected, no improvement was established for line topology, whereas substantial improvements of 18.1% and 29.3% were established for *n=3* and *n=4* respectively.

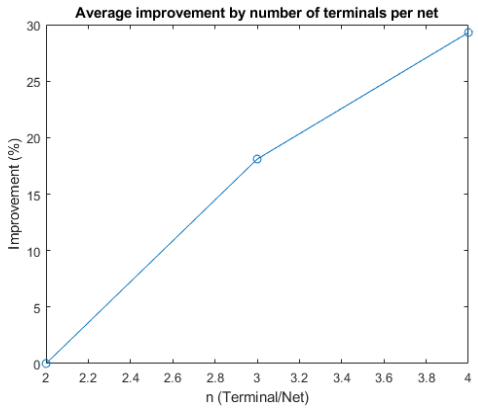

Figure 8. Average improvement by terminal per net

4.4. Total length improvement versus composition of terminal per net

Once the percentage of improvement by the types of net was identified, we can perceive how this data will impact the circuits performance whether in terms of quantity or quality. In this experiment, we would like to prove that circuits which composed of higher number of network topology (*n=3*, *n=4*) will gain the most benefit from ACOSN algorithm. The three circuits chosen for the simulations are [26]:

- Emitter-coupled logic gate (ECL)
- output (partial)
- Cascode amplifier

Figure 9 shows the total length improvement versus composition of terminal per net. From the graph, it can be observed that:

- The ECL circuit has the lowest improvement. The ratio of network topology (*n>2*) to the total net is fair.
- Cascode amplifier circuit shows higher improvement compared to 741 output circuit. The justification would be the ratio of network topology of cascode amplifier versus 741 is higher. Furthermore, the number of net with *n=4* in cascode amplifier is higher than in 741.

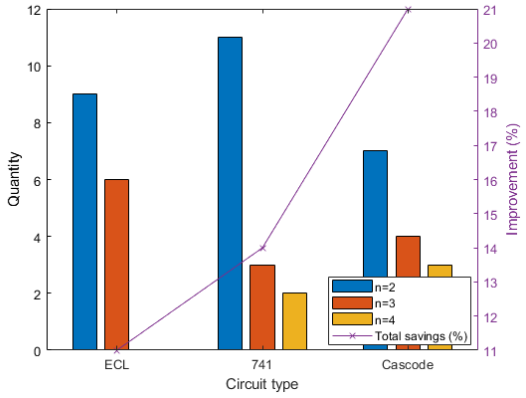

Figure 9. Total length improvement versus composition of terminal per net

In this study, there is not enough data to analyze higher orders of network topology. Emitter-coupled logic has greater network topology to line ratio compared to 741 but the length reduction is lower. Thus, the next analysis would be to find out the efficiency of network topology based on its quality defined by the dominance of the wire length.

4.5. Total length improvement versus coverage of net (*n>2*)

Besides the quantity of nets, the other area to look through is the total length covered by the nets. Figure 10 illustrates the correlation between the total length covered by network topology (*n=3*, *n=4*) and the improvement. From the graph it clearly shows that the more percentage of length covered by the network topology, the more potential it has to reduce the total length. This shows that contrary to earlier justification, 741's superior improvement compared to emitter-coupled logic's is not governed by the quantity ratio between the net, but is due to the dominance of the total length area of network topology. Therefore, it can be concluded that the potential of the circuit routing improvement is dictated by the total length covered by network topology rather than the number of network topology in the circuit.

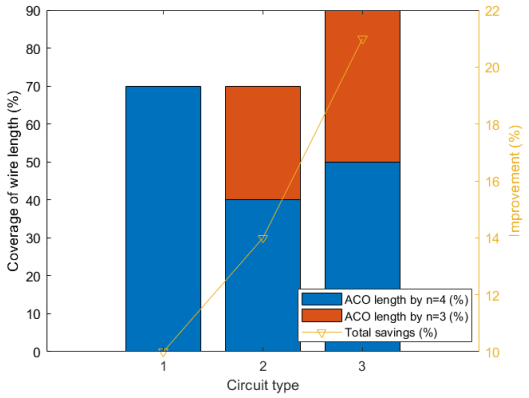

Figure 10. Total length improvement versus coverage of net (*n>2*)

5. CONCLUSION

By adding Steiner points, the wire length of a net constructed by the conventional ACO algorithm can be reduced up to 29.3% depending on the circuit topology. The simulation also shows the circuits benefit in terms of wire length reduction between 10.43% to 21.52% when ACOSN was implemented. Network topologies with four terminals show a two-fold improvement over those with three terminals, due to the insertion of additional Steiner points. Potential length improvement is influenced by the number of complex nets and their total length. In other words, a complex circuit will gain the most benefit from ACOSN algorithm. In conclusion, the main objective of reducing total wire routing by both algorithms which are ACO algorithm and ACOSN are met, and as expected, the algorithm with added Steiner nodes showed better performance. Furthermore, other secondary objectives were also achieved which are automated routing and plotting of circuit using the Excel CSV file. Future work would be adopting or combining machine learning techniques for circuit optimizaiton as it is deemed as the promising solution in reducing the limits in circuit design that previously relies on rule-based technology.

ACKNOWLEDGEMENTS

We are very grateful to Universiti Kebangsaan Malaysia for the financial support under the GUP-2017-074 grant.

REFERENCES

[1] S. H. Gunther, F. Binns, D. M. Carmean & J. C. Hall, "Managing the Impact of Increasing Microprocessor Power Consumption," *In Intel Technology Journal*, vol 1, pp. 1-8, 2001.

[2] P. Kapur, G. Chandra and K. C. Saraswat, "Power estimation in global interconnects and its reduction using a novel repeater optimization methodology," *Proceedings 2002 Design Automation Conference IEEE Cat. No.02CH37324*, New Orleans, LA, USA, pp. 461-466, 2002.

[3] T. Arora and M. E. Moses, "Ant Colony Optimization for power efficient routing in manhattan and non-manhattan VLSI architectures," *2009 IEEE Swarm Intelligence Symposium*, Nashville, TN, pp. 137-144, 2009.
[4] Chia-Jung Liu, Yi-Chen Lin and Jiann-Chyi Rau, "The grid-based two-layer routing algorithm suitable for cell/IP-based circuit design," *2008 15th IEEE International Conference on Electronics, Circuits and Systems*, St. Julien's, pp. 462-465, 2008.
[5] S. Chandra, U. Shrivastava, R. Vaish, S. Dixit and M. Rana, "Improved-AntNet: ACO Routing Algorithm in Practice," *2009 11th International Conference on Computer Modelling and Simulation*, Cambridge, pp. 25-29, 2009.
[6] X. Huang and Z. Liu, "Gridless net routing of integrate circuit with Particle Swarm optimization algorithm," *CICED 2010 Proceedings*, Nanjing, pp. 1-6, 2010.
[7] H. Liying, T. Hongmei, S. Zhijie, Z. Cunshan and Z. Hongdong, "Application of Improved Ant Colony Algorithm in Integrated Circuit Routing," *2009 International Conference on Computer and Communications Security*, Hong Kong, pp. 44-47, 2009.
[8] T. T. Jing, Y. Hu, Z. Feng, X. L. Hong, X. Hu & G. Yan, "A full-scale solution to the rectilinear obstacle-avoiding Steiner problem," in *Integration*, the VLSI Journal, vol. 41, no. 3, pp. 413-425, May 2008.
[9] H. Liying, T. Hongmei, S. Zhijie, Z. Cunshan and Z. Hongdong, "Application of Improved Ant Colony Algorithm in Integrated Circuit Routing," *2009 International Conference on Computer and Communications Security*, Hong Kong, pp. 44-47, 2009.
[10] Changxu Du, Yici Cai, Xianlong Hong and Qiang Zhou, "A shortest-path-search algorithm with symmetric constraints for analog circuit routing," *2005 6th International Conference on ASIC*, Shanghai, pp. 844-847, 2005.
[11] D. Khurge, P. Prashant Kulkarni and V. Sharma, "Compact Automatic Metal Routing Algorithm in DSM technologies," *2017 International Conference on Computing, Communication, Control and Automation ICCUBEA*, Pune, pp. 1-5, 2017.
[12] N. Kito, K. Takagi and N. Takagi, "Automatic Wire-Routing of SFQ Digital Circuits Considering Wire-Length Matching," in *IEEE Transactions on Applied Superconductivity*, vol. 26, no. 3, pp. 1-5, Art no. 1300305, April 2016.
[13] X. Bai, D. Xiao, W. Zhu and J. Chen, "An Initial Detailed Routing Algorithm Considering Advanced Technology Nodes," *2019 China Semiconductor Technology International Conference CSTIC*, Shanghai, China, pp. 1-3, 2019.
[14] A. Sharmin, F. Anwar, S. M. A. Motakabber, "A novel bio-inspired routing algorithm based on ACO for WSNs," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 2, pp. 718-728, 2019.
[15] M. Moza, S. Kumar, "Routing in networks using genetic algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 6, no. 1, pp. 88-98, 2017.
[16] J. Kou, S. Xiong, Z. Fang, X. Zong and F. Bian, "Positive point charge potential field based ACO algorithm for multi-objective evacuation routing optimization problem," *2012 IEEE Congress on Evolutionary Computation*, Brisbane, QLD, pp. 1-8, 2012.
[17] M. Mavrovouniotis and S. Yang, "Ant colony optimization with memory-based immigrants for the dynamic vehicle routing problem," *2012 IEEE Congress on Evolutionary Computation*, Brisbane, QLD, pp. 1-8, 2012.
[18] C. Wang, B. Zhao, W. Peng, C. Wu and Z. Gong, "Routing Algorithm Based on Ant Colony Optimization for DTN Congestion Control," *2012 15th International Conference on Network-Based Information Systems*, Melbourne, VIC, pp. 715-720, 2012.
[19] F. Zhu and H. Wang, "A modified ACO algorithm for multicast state scalability problem based on multicast tree similarity," *13th International Conference on Advanced Communication Technology ICACT2011*, Seoul, pp. 972-976, 2011.
[20] N. Lin and Z. Shao, "Improved Ant Colony Algorithm for Multipath Routing Algorithm Research," *2010 International Symposium on Intelligence Information Processing and Trusted Computing*, Huanggang, pp. 651-655, 2010.
[21] M. Eghbali and M. A. Sharbafi, "Multi agent routing to multi targets via ant colony," *2010 The 2nd International Conference on Computer and Automation Engineering ICCAE*, Singapore, pp. 587-591, 2010.
[22] G. Shyamala, G. R. Prasad, "Obstacle aware delay optimized rectilinear steiner minimum tree routing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 640-652, 2019.
[23] A. B. Kahng, "A Steiner tree construction for VLSI routing," *IJCNN-91-Seattle International Joint Conference on Neural Networks*, Seattle, WA, USA, vol.1, pp. 133-139, 1991.
[24] A. Kahng and G. Robins, "A new class of Steiner tree heuristics with good performance: the iterated 1-Steiner approach," *1990 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, Santa Clara, CA, USA, pp. 428-431, 1990.
[25] T. Barrera, J. Griffith, S. A. McKee, G. Robins and T. Zhang, "Toward a Steiner engine: enhanced serial and parallel implementations of the iterated 1-Steiner MRST algorithm," *Proceedings Third Great Lakes Symposium on VLSI-Design Automation of High Performance VLSI Systems*, Kalamazoo, MI, USA, pp. 90-94, 1993.
[26] A. S. Sedra & Kenneth C. Smith, "Microelectronic Circuits. 5th edition," *New York: Oxford University Press*, pp. 1057, 2004.