Improved ant colony optimization for quantum cost reduction

ShavetaThakral, Dipali Bansal
Department of Electronics and Communication Engineering, Faculty of Engineering and Technology
Manav Rachna International Institute of Research and Studies, India

Article Info

Article history:

Received Jun 11, 2019
Revised Sep 26, 2019
Accepted Feb 5, 2020

Keywords:

Ant colony optimization
Complexity
Heuristic
Quantum cost
Synthesis

ABSTRACT

Heuristic algorithms play a significant role in synthesize and optimization of digital circuits based on reversible logic yet suffer with multiple disadvantages for multiqubit functions like scalability, run time and memory space. Synthesis of reversible logic circuit ends up with trade off between number of gates, quantum cost, ancillary inputs and garbage outputs. Research on optimization of quantum cost seems intractable. Therefore post synthesis optimization needs to be done for reduction of quantum cost. Many researchers have proposed exact synthesis approaches in reversible logic but focussed on reduction of number of gates yet quantum cost remains undefined. The main goal of this paper is to propose improved ant colony optimization (ACO) algorithm for quantum cost reduction. The research efforts reported in this paper represent a significant contribution towards synthesis and optimization of high complexity reversible function via swarm intelligence based approach. The improved ACO algorithm provides low quantum cost based toffoli synthesis of reversible logic function without long computation overhead.

Corresponding Author:

ShavetaThakral,
Department of Electronics and Communication Engineering,
Faculty of Engineering and Technology,
Manav Rachna International Institute of Research and Studies, Faridabad, India.
Email: Shaveta.fet@mriu.edu.in

1. INTRODUCTION

Reversible logic circuit based synthesis and optimization methods have been broadly classified into two categories. Most of the approaches are based on heuristic algorithms like cycle based, binary decision diagram based, exclusive sum of product based, rule based, transformation based, search based or non search based. Each approach proves itself better in terms of some of the point but simultaneously lacking in terms of other. Practical application based reversible circuits are multi qubit based and search space is large. For such applications heuristic algorithm based approaches suffers in terms of scalability, run time, memory space etc and many researchers decided to blaze into evolutionary algorithm based approaches to achieve optimal or near optimal results with saving of computation time and memory space.

Ant colony optimization can be applied with guarantee of convergence and optimal solutions are obtained [1]. Basic algorithm always provides systematic transformation results with convergence [2]. Positive polarity Reed Muller based synthesis method attained good results as far as scalability and run time are concerned [3]. Cycle based synthesis can be used for reversible logic circuits but does not seem good on scalability and there is scope of improvement of run time also; circuit dependency need to be removed [4]. Search based synthesis provides good results but method suffers with limited scalability [5]. Non search based synthesis is found to be ancillary free method and proves improved performance in terms of scalability but circuit suffers with high quantum cost [6]. Exclusive sum of product based synthesis gives encouraging results as far as scalability is concerned but at the cost of more ancillary inputs [7]. Binary decision diagram based synthesis provides good performance with large functions but at the cost of ancillary inputs, garbage outputs and there is circuit dependency also [8]. Transformation based synthesis is ancillary input free but circuit suffers with limited scalability [9]. Rule based synthesis can be applied without consuming any ancillary input but there is high circuit dependency [10]. A hybrid technique for synthesis is used to achieve optimum balance in terms of convergence time, memory space occupied and quantum cost [11]. Cycle based synthesis approach [12] based on principle of decomposition is proposed with assurance of optimum balance of quantum cost and run time and guarantee of convergence. There is no requirement of ancillary input but method suffers with circuit dependency. Ant colony optimization (ACO) based approach [13] is used for synthesis of reversible logic circuits and results are compared with earlier proposed heuristic methods. The performance evaluation shows saving of gates with their proposed method. There is saving of run time and memory space also but quantum cost is not optimized in every solution. BSSSN and its variant can be applied for transformation but quantum cost becomes very high [14, 15] and need to be optimized. Particle swarm optimization based synthesis leads to optimal or near optimal results with saving of synthesis time [16]. A stochastic search based technique can be used to synthesize reversible circuit using simulated annealing combined with ACO. The results are found to be very satisfying [17]. Reversible logic circuit synthesis is proposed using adaptive genetic algorithm and achieves good results in terms of gate count as compare to existing work [18]. Reversible logic synthesis using mixed polarity Toffoli gate is presented and proved efficient as compare to other existing techniques [19]. Binary decision diagram based synthesis approach is proved to be ancilla free and can be used for large number of variables [20]. Optimization is an intractable problem and there is always scope of improvement. Recently different models of ALU have been proposed [21-24] and can be optimized by applying existing techniques or investigated new technique.

Several approaches have been proposed by various authors for reversible logic based design. Yet there is lot of scope to improve quantum cost of reversible function. In this paper improved ant colony optimization is proposed and applied on high complexity reversible function for its synthesis with reduction of its quantum cost. The proposed methodology is given in section 2. The reversible logic function optimization using ACO is given in section 3. Conclusion is given in section 4.

2. RESEARCH METHOD

Ant colony optimization is applied to minimize the quantum cost in reversible logic based ALU circuit. Terminology used for ant colony can be mapped to any reversible logic function by considering nest as output of function and food as input of function. Application of ant colony optimization to reversible logic function leads to the generation of a low quantum cost based circuit by applying Toffoli gates to bring reversible function to its identity function. The proposed methodology is based on depth first search (DFS) in breadth first search (BFS). The synthesis algorithm not only shows all toffoli gates used for synthesis but also identifies reduced quantum cost for synthesis. All shortest possible paths with low quantum cost can be identified with this algorithm. Steps used for implementation are given below:

1. Reversible function is calculated for complexity by cumulatively adding hamming distance between all input states (ABC) and their corresponding output states ($A^0B^0C^0$).
2. Output State with highest hamming distance from input state is identified.
3. If there are multiple output states coming with same hamming distance from their corresponding input states, then randomly choose any one of them.
4. Record all possible state transitions from chosen output state and delete state transition if that leads to corresponding input state.
5. Investigate all possible toffoli gates from chosen library for synthesis that will participate in recorded state transitions.
6. Apply investigated toffoli gates to output function and calculate complexity of each transformed function.
7. Choose toffoli gate which provides lowest complexity transformation and obtained transformation is now considered as current output state ($A^1B^1C^1$). Repeat steps 2 to 7 and transformations are updated as $A^2B^2C^2$, $A^3B^3C^3$ and so on till complexity becomes zero; indicating synthesis process is complete. Add all applied toffoli gates to gate sequence i.e. g1, g2, g3 etc which describe synthesis flow direction from output to input.

This algorithm can be understood with the help of an example. Let output of reversible function is $A^0B^0C^0$={1, 0, 3, 2, 5, 7, 4, 6}. Complexity of this output function is calculated as 8 with respect to input function. As all output state are with equal hamming distance from their respective input state. Therefore state 1 is chosen which can be moved to state 0 or 3 or 5. Toffoli gates selected for these three transitions (1-0), (1-3), (1-5) are T1(C), T1 (B)/T2(C: B), T1 (A)/T2 (C: A) respectively. All these selected toffoli gates are applied to $A^0B^0C^0$ and complexity is calculated for each transformation. After applying **T1(C),** complexity obtained is 4 which is lowest among all obtained transformations. Hence $A^1B^1C^1$={0, 1, 2, 3, 4, 6, 5, 7}. Now all states of $A^1B^1C^1$ are examined for hamming distance and two states 6 and 5 are identified with highest hamming distance of 2 from their corresponding input states. Now depth first search algorithm can be applied to state 6. State 6 can be further moved to state 2 or 4 or 7. Toffoli gates selected for these three transitions (6-2), (6-4), (6-7) are T1 (A)/T2 (B: A), T1 (B)/T2 (A: B), T1(C)/T2 (A: C)/T2 (B: C)/T3 (A, B: C) respectively. All these selected toffoli gates are applied to $A^1B^1C^1$ and complexity is calculated for each transformation. After applying T2 (A: B)/T3 (A, C: B)/T2 (A: C)/T3 (A, B: C) complexity obtained is 4 which is lowest among all obtained transformations. Hence $A^2B^2C^2$=case 1[**T2(A:B)** {$A^1B^1C^1$}]= {0, 1, 2, 3, 6, 4, 7, 5} or case 2{0,1,2,3,4,6,7,5} or case 3{0,1,2,3,5,7,4,6} or case 4{0,1,2,3,4,7,5,6}. Now randomly first two cases are chosen for analysis. Now depth first search can be further applied to case 1. In case 1, all states of $A^2B^2C^2$ are examined for hamming distance and all four states 6, 4, 7 and 5 are identified with highest hamming distance of 1 from their corresponding input states. Now state 6 can be further moved to state 2 or 4 or 7. Transition (6-7) via **T3 (A, B: C)** results in lowest complexity of 4. Hence $A^3B^3C^3$={0, 1, 2, 3, 7, 4, 6, 5}. Now all states of $A^3B^3C^3$ are examined for hamming distance. State 7 is found to be with highest hamming distance of 2. Now state 7 can be moved to state 3 or 5 or 6. Transition (7-5) via **T3 (A, C: B)** results in lowest complexity of 2. Hence $A^4B^4C^4$={0, 1, 2, 3, 5, 4, 6, 7}. Now all states of $A^4B^4C^4$ are examined for hamming distance. Further Transition (6-7) via **T3 (A, B: C)** results in lowest complexity of 2 and hence $A^5B^5C^5$={0, 1, 2, 3, 5, 4, 7, 6}. Finally state transition (5-4) via **T2 (A: C)** results in identity function $A^5B^5C^5$={0, 1, 2, 3, 4, 5, 6, 7} where complexity is zero. Sequence of Toffoli gates followed is [T1(C), T2(A: B), T3(A, B:C), T3 (A,C:B), T3(A,B:C), T2(A:C)]. Similarly toffoli sequence for case 2 can be calculated as [T1(C), T3 (A, B: C), T3 (A, C: B), T3 (A, B: C),]. Path followed ABCDEFG and ABC'DG in Figure 1 represents case 1 and 2 respectively. Case 1 leads to reversible function synthesis with the help of 6 gates and quantum cost calculated is 18. Case 2 leads to reversible function synthesis with the help of 4 gates and quantum cost calculated is 16. Case 3 and Case 4 leads to implementation with the help of 6 gates with quantum cost of 18 and 4 gates with quantum cost of 16 respectively.

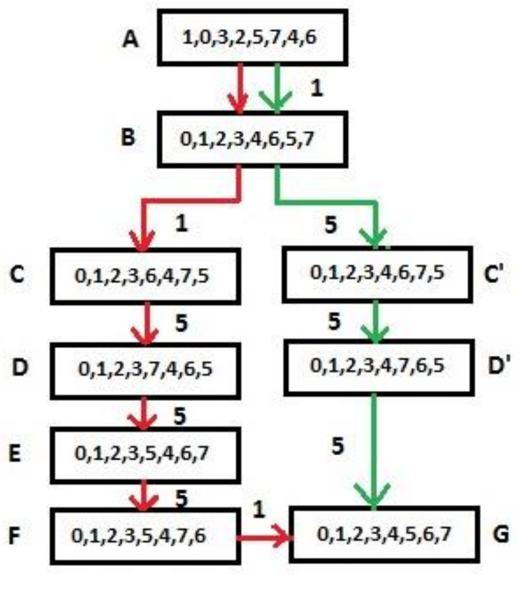

Figure 1. Reversible logic synthesis implementation

2.1. Procedure for probabilistic construction of ant solution

Initially i ants are positioned on output of reversible function. Each ant progressively builds a route by choosing a gate randomly from NCT library but biased partly on pheromone on edges (τ) and heuristic (η). The pheromone on edge i.e. trail strength is multiplied by problem specific heuristic information. Ant 'a' chooses a gate from NCT library in such a way that complexity of function decreases progressively as well as gate chosen should have low quantum cost. In this way ant chooses path with shorter edges having less quantum cost and denser trail strength. Exponent of pheromone function and heuristic function are specified as α and β respectively. There is trade off between these two parameters. For simplicity α and β are put equal to one. Any gate belongs to NCT library functions according to two types of bits i.e. control bit and target bit. The information on target bit is changed according to information on control bit/bits. When an ant a is at current state CS_a in K^{th} iteration, then probability of choosing t^{th}as target bit is given in (1).

$$P_t(CS_a, K) = \frac{[\tau_t(CSa, K)]^{\alpha}[\eta_t]^{\beta}}{\Sigma_{\text{all legal b}}(\tau_b)^{\alpha}[\eta_b]^{\beta}} \quad (1)$$

Where $\tau_t$is cumulative trail strength to choose t^{th} bit as target bit for all edges between current state (CS_a) and final state (identity function).The $\eta_t$is weighing function to choose t^{th} bit as target bit in such a way that complexity of function should decrease progressively along with reduction of quantum cost if there is tie between two states with same complexity.Similarly probability of choosing control bit is given in (2).

$$P_c(CS_a, K) = \frac{[\tau_c(CSa, K)]^{\alpha}[\eta_c]^{\beta}}{\Sigma_{\text{all legal b}}(\tau_b)^{\alpha}[\eta_b]^{\beta}} \quad (2)$$

2.2. Procedure for pheromone updation

After stochastic construction of route by all ants, pheromone Updation rule is applied as per (3). As per first part of rule, pheromone evaporation is done to evaporate and henceforth decrease pheromone values so that unnecessary accumulation of pheromone is avoided that may otherwise lead to rapid convergence of algorithm. As per second part of rule, each ant is allowed to add pheromone on edges it has visited. In this way denser trails with better solution get constructed and leads to higher reinforcement.

$$\tau_{xy}(g, K+1) = (1-\rho)\,\tau_{xy}(g,k) + \Sigma^{i}_{a=1}\Delta\tau^{a}_{xy}(g,K) \quad (3)$$

Where ρ is pheromone evaporation rate and usually taken greater than 0 and less than equal to 1. $\Delta\tau^{a}_{xy}(g, K)$ is the amount of pheromone that ant a deposits at K^{th} iteration. Its mathematical value is given in (4).

$$\Delta\tau^{a}_{xy}(g, K) = \begin{bmatrix} \dfrac{Q * Dist(g)}{QC(route_a)} & \text{if } g\epsilon\ route_a \\ 0 & \text{otherwise} \end{bmatrix} \quad (4)$$

Where Q is total amount of pheromone released by an ant a. QC is total quantum cost of route. If a gate has less quantum cost, then amount of pheromone deposited will be more. Dist (g) indicates position of gate from start of route.

3. REVERSIBLE LOGIC FUNCTION OPTIMIZATION

The ACO algorithm employed for reversible function optimization starts with implementation of pheromone table. Path updation as per implemented pheromone table and reinforcement enables finding shortest path with reduced quantum cost. An example is illustrated to understand reversible logic function optimization procedure. Let the reversible function be *f(n)*={5,2,0,4,7,3,1,6}, the improved ant colony optimization algorithm transforms this function to an identity function with a sequence of Toffoli gates in terms of TOF1, TOF2 and TOF3. TOF1 (T1) is called as NOT gate, TOF2 (T2) is called as CNOT gate and TOF3 (T3) is called as Toffoli gate. Table 1 represents the truth table for the chosen reversible function on which proposed ant colony optimization algorithm is applied.

For 3 bit reversible function, initial pheromone graph can be constructed by taking eight vertices and 64 edges. Among 64 edges, 32 edges are significant and rests 32 are not significant. A state transition can only takes place if hamming distance between input and output vector is either 0 or 1 bit. Significant 32 edges indicate state transition possibility with hamming distance 0 or 1. Non significant 32 edges indicate no possibility of state transition as hamming distance becomes greater than 1. Among 32 significant edges, 8 edges return to same vertex from which they initiated and need to be trimmed. Rests 24 are significant on practical basis. Switching from one state to other should be done in such a way that hamming distance between two states should be one. If initial state is 0, then next state can be 1, 2 or 4. If initial state is 1, then next state can be 3, 5 or 0. If initial state is 2, then next state can be 3, 6 or 0. If initial state is 3, then next state can be 2, 7 or 1. If initial state is 4, then next state can be 0, 6 or 5. If initial state is 5, then next state can be 1, 7 or 4. If initial state is 6, then next state can be 2, 4 or 7. If initial state is 7, then next state can be 3, 6 or 5. The hamming distance between vertices (0-1), (0-2), (0-4), (1-0), (1-3), (1-5), (2-0), (2-3), (2-6), (3-1), (3-2), (3-7), (4-0), (4-5), (4-6), (5-1), (5-4), (5-7), (6-2), (6-4), (6-7), (7-3), (7-5). (7-6) is 1; hence these represent significant 24 edges which will participate in state transition and gate choosing. The hamming distance between states (0,7), (1,6), (2,5), (3,4), (4,3), (5,2), (6,1), (7,0) is 3. The hamming distance from one vertex to each other possible vertex is given in Table 2. It can be logically calculated by applying XOR bit by bit between two vertices which need to be used for calculation of hamming distance. If bits are alike; then XOR logic provides 0, otherwise XOR logic provides 1 corresponding to those two bits. Then the XOR logic result of all three comparisons are passed to full adder, the output of full adder provides hamming distance in binary form and can be converted to decimal as per requirement. Complexity of a reversible function can be calculated by calculating state by state cumulative hamming distance between input and output function. The highest worst case complexity can be seen with reversible function f={7, 6, 5, 4, 3, 2, 1, 0} and it is calculated as 24.

Table 1. Truth table

c	b	a	c^0	b^0	a^0
0	0	0	1	0	1
0	0	1	0	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	1	0	0	0	1
1	1	1	1	1	0

Table 2. Hamming distance calculation

Vertex	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	0	1	1	2	1	2	2	3
1(001)	1	0	2	1	2	1	3	2
2(010)	1	2	0	1	2	3	1	2
3(011)	2	1	1	0	3	2	2	1
4(100)	1	2	2	3	0	1	1	2
5(101)	2	1	3	2	1	0	2	1
6(110)	2	3	1	2	1	2	0	1
7(111)	3	2	2	1	2	1	1	0

For 3 bit reversible function, there exists 12 Toffoli gates : T1(A), T1(B), T1(C), T2(A:B), T2(A:C), T2(B:A), T2(B:C), T2(C:A), T2(C:B), T3(A,B:C), T3(B,C:A), T3(C,A:B). As all these gates are reversible, therefore state transition in both directions [(0,1) (1,0)], [(0,2) (2,0)], [(0,4) (4,0)], [(1,3) (3,1)], [(1,5) (5,1)], [(2,3) (3,2)], [(2,6) (6,2)], [(3,7) (7,3)], [(4,5) (5,4)], [(4,6) (6,4)], [(5,7) (7,5)], [(6,7) (7,6)] select same Toffoli gate. State transitions with possibility of Toffoli gate selection is shown in Table 3. Table summary shows that some state transitions select only NOT gate (T1) and other may choose between NOT (T1) and CNOT (T2) gate and Toffoli gate (T3). Selection of a right gate leads to optimum synthesis and usually a gate is selected in such a way that complexity of function should decrease progressively or if same complexity falls with any of two gate options, then gate with lower quantum cost should be given priority. In this way optimum reversible circuit synthesis can be done in terms of less gate count as well as low quantum cost.

Table 3 can be seen from another prospective where it will be possible to see requirement of a toffoli gate for all possible transitions. Table 4 shows 12 toffoli gates against all possible transitions. Toffoli gates T3(A,B:C), T3(A,C:B) and T3(B,C:A) are used for state transitions [(6,7) (7,6)],[(5,7) (7,5)], [(3,7) (7,3)] respectively. It means these transitions are costly as compare to other transitions if moved through toffoli gates with quantum cost 5. Rest other transitions make use of NOT gate or CNOT gate with unity quantum cost.

Table 3. Toffoli gate selection

State Transition 3 Bit Reversible Function (ABC)	Toffoli Gate Selection		
	NOT Gate (T1)	CNOT Gate (T2)	Toffoli Gate (T3)
0 $\rightleftarrows$ **1**	T1(C)	-	-
0 $\rightleftarrows$ **2**	T1(B)	-	-
0 $\rightleftarrows$ **4**	T1(A)	-	-
1 $\rightleftarrows$ **3**	T1(B)	T2(C:B)	-
1 $\rightleftarrows$ **5**	T1(A)	T2(C:A)	-
2 $\rightleftarrows$ **3**	T1(C)	T2(B:C)	-
2 $\rightleftarrows$ **6**	T1(A)	T2(B:A)	-
3 $\rightleftarrows$ **7**	T1(A)	T2(B:A) or T2(C:A)	T3(B,C:A)
4 $\rightleftarrows$ **5**	T1(C)	T2(A:C)	-
4 $\rightleftarrows$ **6**	T1(B)	T2(A:B)	-
5 $\rightleftarrows$ **7**	T1(B)	T2(A:B) or T2(C:B)	T3(A,C:B)
6 $\rightleftarrows$ **7**	T1(C)	T2(A:C) or T2(B:C)	T3(A,B:C)

Table 4. State transition selection

Toffoli Gate	State Transition Selection			
T1(A)	0 $\rightleftarrows$ 4	1 $\rightleftarrows$ 5	2 $\rightleftarrows$ 6	3 $\rightleftarrows$ 7
T1(B)	0 $\rightleftarrows$ 2	1 $\rightleftarrows$ 3	4 $\rightleftarrows$ 6	5 $\rightleftarrows$ 7
T1(C)	0 $\rightleftarrows$ 1	2 $\rightleftarrows$ 3	4 $\rightleftarrows$ 5	6 $\rightleftarrows$ 7
T2(A:B)	4 $\rightleftarrows$ 6	5 $\rightleftarrows$ 7		
T2(A:C)	4 $\rightleftarrows$ 5	6 $\rightleftarrows$ 7		
T2(B:A)	2 $\rightleftarrows$ 6	3 $\rightleftarrows$ 7		
T2(B:C)	2 $\rightleftarrows$ 3	6 $\rightleftarrows$ 7		
T2(C:A)	1 $\rightleftarrows$ 5	3 $\rightleftarrows$ 7		
T2(C:B)	1 $\rightleftarrows$ 3	5 $\rightleftarrows$ 7		
T3(A,B:C)	6 $\rightleftarrows$ 7			
T3(A,C:B)	5 $\rightleftarrows$ 7			
T3(B,C:A)	3 $\rightleftarrows$ 7			

Selection of toffoli gate as per state transition in 3 bit reversible function can be applied to multi qubit function. Number of transitions and number of Toffoli gate options increase as per reversible logic function. The Ant system graph is constructed for given reversible function as shown in Figure 2. The quantum cost is highlighted on every edge. The quantum cost 1 is shown for family of gates belonging to T1 and T2. The quantum cost 5 is shown for T3 with two positive control lines or one negative and one positive control line. The quantum cost 6 is shown for T3 with two negative control lines. The shortest optimal path with lowest quantum cost is highlighted with green colour. Table 5 shows comparative analysis of toffoli network built through basic algorithm as per literature, improved basic algorithm, bidirectional algorithm, bit string swap sorting network (BSSSN) and variant of BSSSN with proposed ACO.

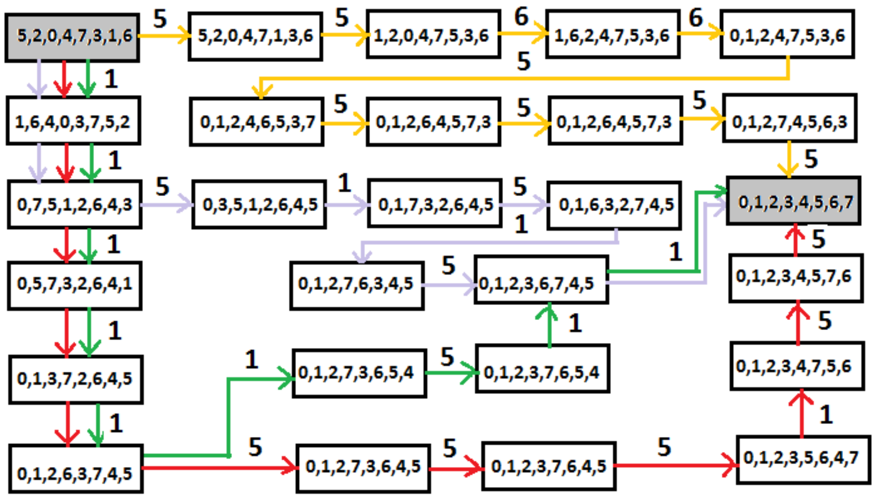

Figure 2. Ant system graph for reversible function

Table 5. Comparative analysis

Algorithm	Total Gates	Quantum cost
Basic algorithm [2]	11	31
Improved Basic algorithm [25]	8	20
Bidirectional algorithm [2]	8	20
BSSSN [14]	21	57
Variant of BSSSN [14]	19	63
Proposed ACO	9	13

4. CONCLUSION

In this paper, ant colony optimization algorithm is proposed with goal of reduction of quantum cost. The proposed ant colony optimization algorithm has been applied on high complexity reversible function. and result shows that after applying ACO, quantum cost gets reduced in comparison with other existing methods. It is also observed that it is not necessary that reduction of gates always find optimum solution. Improved basic algorithm and bidirectional algorithm synthesized reversible function using 8 gates with quantum cost 20 yet proposed ACO method synthesized reversible function using 9 gates with quantum cost 13.

REFERENCES

[1] W. J. Gutjahr, “ACO Algorithms with Guaranteed Convergence to the Optimal Solution,” *Information processing letters*, vol. 82, no. 3, pp. 145-153, 2002.
[2] D. M. Miller, et al., “A transformation Based Algorithm for Reversible Logic Synthesis,” *In Proceedings 2003. Design Automation Conference* (IEEE Cat. No. 03CH37451), IEEE, pp. 318-323, 2003.
[3] P. Gupta, et al., “An Algorithm for Synthesis of Reversible Logic Circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 11, pp.2317-2330, 2006.
[4] V. V. Shende, et al., “Synthesis of Reversible Logic Circuits,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 6, pp.710-722, 2003.
[5] M. Saeedi, et al. “On the Behavior of Substitution-Based Reversible Circuit Synthesis Algorithms: Investigation and Improvement,” *In IEEE Computer Society Annual Symposium on VLSI (ISVLSI'07), IEEE*, pp. 428-436, 2007.
[6] M. Saeedi, et al.,”A Novel Synthesis Algorithm for Reversible Circuits,” *In 2007 IEEE/ACM International Conference on Computer-Aided Design, IEEE*, pp. 65-68, 2007.
[7] K. Fazel, et al., “ESOP-Based Toffoli Gate Cascade Generation,” in *2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 206-209, 2007.
[8] R. Wille and R.Drechsler, “BDD-Based Synthesis of Reversible Logic for Large Functions,” in *Proceedings of the 46th Annual Design Automation Conference, ACM*, pp. 270-275, 2009.
[9] N. Alhagi, et al., “Synthesis of Reversible Circuits with no Ancilla Bits for Large Reversible Functions Specified with Bit Equations,” In *2010 40th IEEE International Symposium on Multiple-Valued Logic, IEEE*, pp. 39-45, 2010.
[10] M. Arabzadeh, et al., “Rule-Based Optimization of Reversible Circuits,” in *Proceedings of the 2010 Asia and South Pacific Design Automation Conference, IEEE press*, pp. 849-854, 2010.
[11] D. K. Kole, et al., “Optimal Reversible Logic Circuit Synthesis Based on a Hybrid DFS-BFS Technique,” *In 2010 International Symposium on Electronic System Design, IEEE*, pp. 208-212, 2010.

[12] M. Saeedi, et al., “A Library-Based Synthesis Methodology for Reversible Logic,” *Microelectronics Journal*, vol. 41, no. 4, pp.185-194, 2010

[13] M. Li, et al., “Reversible Logic Synthesis through Ant Colony Optimization,” *in Proceedings of the Conference on Design, Automation and Test in Europe,* pp. 307-310, 2010.

[14] M. Islam, “BSSSN: Bit string swapping sorting network for reversible logic synthesis,”arXiv preprint arXiv: 1008.4668, 2010.

[15] M. Islam and A. A. Mahmud, “Sorting Network for Reversible Logic Synthesis,” *arXiv preprint arXiv: 1008.3694*, 2010.

[16] K. Datta, et al., “Particle Swarm Optimization Based Circuit Synthesis of Reversible Logic,” *in 2012 International Symposium on Electronic System Design (ISED), IEEE,* pp. 226-230, 2012.

[17] M. Sarkar,et al., “Reversible Circuit Synthesis using ACO and SA Based Quine-McCluskey Method,” *in IEEE 56th International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE,* pp. 416-419, 2013.

[18] T. N. Sasamal, et al.,”Reversible Logic Circuit Synthesis and Optimization using Adaptive Genetic Algorithm,” *Procedia Computer Science*, vol. 70, pp.407-413, 2015.

[19] C. S. Cheng, A.K.Singh and L.Gopal, “Efficient Three Variables Reversible Logic Synthesis Using Mixed-Polarity Toffoli Gate,” *Procedia Computer Science*, vol. 70, pp.362-368, 2015.

[21] M. Soeken, et al., “Ancilla-Free Synthesis of Large Reversible Functions using Binary Decision Diagrams,” *Journal of Symbolic Computation*, vol. 73, pp.1-26, 2016.

[22] S. Thakral and D. Bansal, “Fault Tolerant Arithmetic Logic Unit,” *International Conference on Emerging Current Trends in Computing and Expert Technology*, pp. 226-233, 2019.

[23] S. Thakral and D. Bansal, “Improved Fault Tolerant ALU Architecture,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 6, pp. 1477-1484, 2019.

[24] A. Kamaraj and P. Marichamy, “Design of Integrated Reversible Fault-Tolerant Arithmetic and Logic Unit,” *Microprocessors and Microsystems*, vol. 69, pp. 16-23, 2019.

[25] S. Thakral, et al., “Review of Truth Table Based Reversible Logic Synthesis Methods,” in *International Conference on Soft Computing Techniques and Implementations (ICSCTI), IEEE*, pp. 164-169, 2015.

BIOGRAPHIES OF AUTHORS

Shaveta Thakral is presently working as an Associate Professor in Electronics & communication department, Faculty of Engineering and technology, MRIIRS, Faridabad. She obtained her BE in Electronics and communication from Lingayas Institute of management and Technology, Faridabad; MTECH from IASE Deemed University, Rajasthan. Currently she is pursuing PhD from MRIIRS, Faridabad. Her current research area includes Analog and Digital circuits, VLSI and Microprocessor. She has work experience of 14 years. She has published 30 research papers in prestigious indexed journals and conferences.

Dr. Dipali Bansal is presently Professor & associate Dean Academics, MRIIRS, Faridabad. She did her Bachelors in Electronics & Communication Engineering from BIT Sindri, a renowned and sought after learning hub and has also earned a PhD degree from JamiaMiliaIslamia, New Delhi where she worked on Digital Signal processing and its applications in home health care. She is a part of curiosity driven research group working in the field of bio-signal processing that brings together experimental and theoretical techniques and approaches in acquiring and analyzing human physiological parameters viz. ECG, EMG, EEG signals using professional tools like MATLAB and LabVIEW. Dr Bansal has over 70 publications in prestigious indexed journals and conferences, is mentor to 08 PhD scholars. She is also Reviewer of many international journals. Dr. Bansal is a member of various advisory boards at the University and is a motivational speaker at various forums.