

The problem of communication design has been defined as one of the problems that belong to the category of NP-hard problem, and the aim of the topological communication network design is to identify component placement locations and connectivity aspects. On the other hand, the Reliable Communication Network Design (RCND) is a popular optimization problem used for maximizing network reliability. In addition, finding an accurate calculation of RCND explains the problem of NP-hard problem. To this end, literature studies suggested various metaheuristic algorithms that have been used as approximation methods to find the best solution to this problem. Some of these algorithms belong to the Evolutionary Algorithms (EAs) category, such as Genetic Algorithms (GAs), and some belong to the Swarm Intelligence Algorithms (SIAs) category, such as Ant Colony Optimization (ACO). However, to the best of our knowledge, the Ant Colony System (ACS) algorithm, which is considered an updated version of ACO, has not yet been used to design reliability-constrained communication network topologies. Therefore, this study aims to apply the updated version of the ACS algorithm for solving RCND in small, medium, and large networks. The proposed algorithm was benchmarked against present state-of-the-art techniques that address this challenge. The research findings show that the proposed algorithm is an optimal solution for a fully connected small network size ( $n=6, 7, 8$ , and  $9$ ) and it has been achieved as an optimal solution for all not fully connected sets ( $n=14, 16$ , and  $20$ ). In each case, the results for medium-sized networks were better than the benchmark results

**Keywords:** Ant colony system algorithm, reliable communication network design, Monte Carlo method

UDC 519  
DOI: 10.15587/1729-4061.2022.259693

# IMPLEMENTATION OF THE ENHANCED ANT COLONY SYSTEM ALGORITHM TO SOLVE RELIABLE COMMUNICATION NETWORK DESIGN

**Abeer A. Abdul-Razaq**

PhD, Associate Professor  
Department of Mathematics  
College of Computer Science and Mathematics  
Thi-Qar University  
Park str., Nasiriyah, Iraq, 64001

**Huda Karem Nasser**

MSc, Teacher\*

**Asaad Shakir Hameed**

Corresponding author

PhD, Teacher\*

Data Mining and Optimization Group, Centre of Artificial Intelligence  
National University of Malaysia  
Bangi, Selangor, Malaysia, 43600  
E-mail: asaadshakir@utq.edu.iq

**Modhi Lafta Mutar**

PhD, Teacher\*

Department of Medical Instruments Engineering Techniques  
Al-Turath University College  
Mansour str., Baghdad, Iraq, 12013

**Haiffa Muhsan B. Alrikabi**

PhD

Department of Mathematics  
College of Education for Pure Sciences  
Thi-Qar University  
Near Victory Bridge, Nasiriyah, Iraq, 64001

**Mohammed F. AL-Rifaie**

Master of Computer Networking  
Basrah University College of Science and Technology  
Wofood str., Basra, Iraq, 61001

**Mustafa Musa Jaber**

PhD, Lecturer

Department of Medical Instrumentation Techniques Engineering  
Dijlah University College  
Al Masafi str., Baghdad, Iraq, 00964

Department of Medical Instruments Engineering Techniques  
Al-Farahidi University  
Jadreja str., Baghdad, Iraq, 10021

\*Department of Mathematics

General Directorate of Thi-Qar Education, Ministry of Education  
Al-Haboubistr., Nasiriyah, Iraq, 64001

Received date 15.04.2022

**How to Cite:** Abdul-Razaq, A. A., Nasser, H. K., Hameed, A. S., Mutar, M. L., Alrikabi, H. M. B., AL-Rifaie, M. F., Jaber, M. M. (2022).

Accepted date 16.06.2022

Implementation of the enhanced ant colony system algorithm to solve reliable communication network design. Eastern-European Journal of Enterprise Technologies, 3 (9 (117)), 44–52. doi: <https://doi.org/10.15587/1729-4061.2022.259693>

Published date 30.06.2022

## 1. Introduction

The Reliable communication network design (RCND) intends to identify a communication network topology that fulfils specific reliability criteria at minimum possible cost. The RCND is classified as NP-hard [1].

The literature specifies numerous widely used techniques to solve RCND problems, and these methods can be classified as exact and approximate approaches; the latter comprises metaheuristic algorithms. Exact approaches such as Branch and Bound (B&B) are extremely time-intensive, and those approaches looked for an accurate solution to the

problem that all links in the network had the same probability of operation and were successful with a small size of problems [2, 3].

Hence, academicians and others focusing on this domain emphasise the development of approximate approaches to solve more significant RCND problems within an acceptable computational period [3, 4]. In the metaheuristic context, researchers [1] suggested using the Improved Ant Colony Optimization (ACO) approach to identify network possibilities that reduce cost but fulfil reliability requirements. The literature [5] studied a Monte Carlo (MC) simulation and suggested using particle swarm optimization to identify a topology that reduces cost but adheres to reliability requirements.

Furthermore, the ACO can be used for solving the path problems for determining the best route where all components of the algorithm were compatible with the properties of the problems wherein the ant nests are regarded as depots, artificial ants are considered as vehicles, food acts as the customer, trails form the routes, and pheromone concentration can present the best route for optimising the distance [6].

Many algorithms are included in the ACO family of algorithms [7] like the Ant System (AS), Ant Colony System (ACS), etc. The ACS algorithm is inspired by nature, wherein the ants' scout for food in their neighborhood. Thus, this algorithm is a probabilistic technique and belongs to the swarm intelligence process. Recently, researchers the study [8] have compared all types of ACO algorithms and noted that the ACS algorithm developed properties of its components through an amended transition rule, the introduction of a local update of pheromone, and the global pheromone update is only the best solution. Moreover, they have proposed an efficient improvement of the ACS algorithm for handling capacity vehicle routing problem (CVRP) based on the recent concept which is called sub-paths.

On the other hand, the previous studies have succeeded in optimizing the solutions to other problems in the field of combinatorial optimization problems (COPs) by applying the ACS algorithm such as the traveling salesman problem (TSP) by the study [9] and the vehicle guidance problem (VRP) by using the literature [10]. However, as far as we know, the design of reliability-constrained communication network topologies using the ACS algorithm has not yet been used. On the opposite hand, the character of the RCND problem is complicated as it belongs to a class of NP-hard problems. Therefore, the studies within side the literature committed their efforts to suggesting metaheuristic algorithms to find the best solutions for RCND data that included small, medium, and large network sizes. However, the solutions of those networks still need to be improved, in particular the networks of medium and large size, therefore, this study has been proposed.

---

## 2. Literature review and problem statement

---

This section presents a discussion and analysis of those relevant studies that dealt with solving the RCND problems in the literature by proposed heuristic and metaheuristic algorithms as follows. The researchers in the reference [11] have proposed a hybrid approach based on Ant Colony Optimization (ACO) and Simulated Annealing (SA), called ACO\_SA for solving the RCND problems. This hybrid is featured by the possibility of obtaining an effective solution

through the ACO algorithm while the SA algorithm enabled this hybrid to escape from the local optimal to find better solutions. Regarding their computational results, they obtained the best solutions at the time compared to those algorithms that were compared with them, however, they did not reach the optimal results, specifically in the small networks that have optimal solutions.

The main goal of the related studies is to find the minimum cost network topology with reliability constraints. Therefore, the study [2] has proposed two metaheuristics (Simulated Annealing (SA) and Genetic Algorithm (GA)) to achieve this goal. With respect to Algorithm SA, they proposed adding a penalty cost to the suitability function of the candidate who could not meet the reliability requirements. While their suggestion was included in the GA generating a new crossover operator to protect the two connections of all its children. In order to evaluate the performance of their algorithms (SA and GA), they compared their performance with that of the approach (ACO\_SA) that was proposed in the study [11]. The results obtained by the study [11] have been improved where some optimal solutions were obtained for small-sized networks, in addition to the results related to other medium and large sizes better than those resulting by the algorithm ACO\_AS.

Another study [12] was interested in solving the RCND problems, and this study applied a new non-parameter-tuning approach called self-tuning heuristics (STH). The results show that STH and ACO\_SA [11] have similar solution quality for small test problems. On the other hand, STH's solution quality improves as the size of the problem grows.

When designing metaheuristic algorithms, the researchers need to consider two conflicting criteria called diversification (exploration) and intensification (exploitation). Therefore, a recent study [3] examined these two major issues by proposing a hybrid approach to balance exploration and development. They proposed two new hybrid metaheuristic algorithms, namely, GABB and SABB, by integrating either a Genetic Algorithm (GA) with the Branch and Bound method (B&B) and present two versions of this hybrid (GABB1 and GABB2) or Simulated Annealing (SA) with B&B. Although the proposed hybrid techniques having superior to the studies [11, 12] in a three-stage experiment using small, moderate, and large networks. Moreover, the hybrid algorithm GABB2 has improved the current literature solution by 8–35 % in terms of computational results. However, they are more compute-intensive compared to the other techniques.

In conclusion, let's conclude that there are two problems worth discussing in this research work as follows:

1. From a technical point of view, researchers in these related studies have proposed various metaheuristic algorithms that can be used as approximations to find the best solution to this problem. Some of these algorithms belong to the Evolutionary Algorithms (EA) category, such as Genetic Algorithms (GAs), and some belong to the Group Swarm Intelligence Algorithms (SIAs) category, such as Ant Colony Optimization (ACO). After analysis and criticism of these literary studies, a new direction to existing solutions to the RCND problems by using the latest improved version of the ACS algorithm to design reliability-constrained communication network topologies.

2. From a domain perspective, RCND problems data has the optimal solutions for fully connected small network sizes ( $n=6, 7, 9, 10$ ) and not fully connected sets ( $n=14, 16$ , and 20) while other sizes solutions for medium network

sizes ( $n=15, 20, 25$ ) and large network sizes ( $n=30, 40, 50$ ) have the best solutions in the literature, but do not give the optimal solution yet. Therefore, these cases are considered open research. And therefore, this has brought this issue to the attention of researchers and scholars.

### 3. The aim and objectives of the study

The aim of the study is to find the best solution to the RCDN problems compared to the current solution in the literature by finding the network topology at the lowest cost with reliability constraints.

To achieve this aim, the following objectives are accomplished:

- to propose to apply the updated version of the ACS algorithm that has used sub-paths for solving RCND problems;
- to implement the enhanced ant colony system (EACS) algorithm to solve the RCND problems that include networks of several sizes (small, medium, and large). In addition to evaluating its solutions with those solutions obtained by the state-of-the-art approaches in the literature.

### 4. Materials and methods

#### 4.1. Concepts and tools adopted in this work

This work used explaining the formulation of the communication network model, algorithm generating initial network topology, Monte Carlo (MC) simulation, and Ant Colony System (ACS) algorithm. In addition, MATLAB R2018b (9.5.0.9444) 64-bit software was applied to encode the theoretical steps and implement the proposed algorithm for solving problem data by running the algorithm 10 times with 10000 iterations.

#### 4.2. Communication network modelling

A non-directed graph  $G=(N, L)$  is used to specify a communications system, where  $N$  depicts a group comprising  $n$  vertices (or nodes), while  $L$  depicts a collection of  $k$  edges (or links). Every edge is associated with a reliability and cost number. Link reliability is the degree to which it performs as expected for a specified period. Network cost comprises cables, terminal creation, connections, and installation [13]. The following paragraph specific the mathematical representation of the system used for this research:

$$\min f(G_x) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N c_{ij} x_{ij}. \quad (1)$$

Subject to:

$$R^*(G_x) \geq R_0, \quad (2)$$

$$\sum_{j=1, j \neq i}^N x_{ij} \geq 2, \quad \forall i \in N, \quad (3)$$

$$x_{ij} = 0 \text{ or } 1, \quad \forall i \in N. \quad (4)$$

(1) specifies the objective function that should be minimized to reduce overall network cost and satisfy re-

quirements (2), (3). Equation (2) describes the reliability requirement.

$c_{ij}$  denotes link  $(i, j)$  cost,  $x_{ij}$  denotes the decision variable,  $R^*(G_x)$  is an estimate of the  $G_x$  network structure based on all terminals, and  $R_0$  denotes the lowest permissible all-terminal reliability value.

Fig. 1 shows example of a network topology graph with 5 nodes as follows.

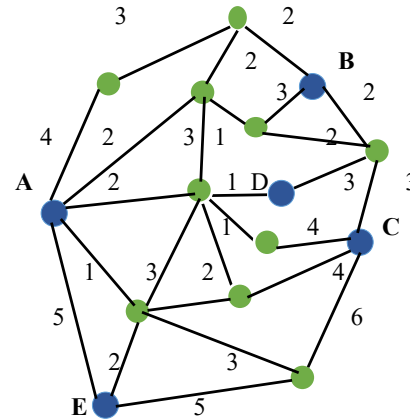


Fig. 1. An example of a network topology graph with five nodes (cost 71) [4]

Another example of a network topology graph is shown in Fig. 2 as follows.

The previous Fig. 1, 2 show diagrams of network topology for instance has 5 nodes and another instance has 7 nodes. Also, the best cost value obtained by the objective function was 71 for 5 nodes and 53 for 7 nodes.

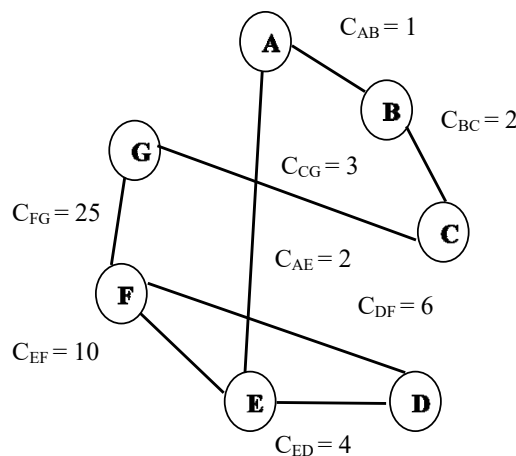


Fig. 2. An example of a network topology graph with seven nodes (cost 53) [3]

#### 4.3. Ant Colony System (ACS) algorithm

This research uses a recently updated version of the ACS algorithm proposed by the study [8] after refinement of some of its steps based on the properties of the RCND problem, small, medium, and large networks. Thereafter let's implement it to find the best topology for those networks that provides the best solutions in a reasonable amount of time.

**Step 1. Initialization.** The system begins by initializing the primary pheromone  $\tau_0$  and  $\tau_0 > 0$  for every link  $(i, j)$  using the expression  $\tau_0 = \frac{1}{f(G_x)}$ , and representing the starting parameter representing pheromone influence, leading to the output cost  $f(G_x)$ . The study [11] has presented a two-phase initiation process to create the network structure that has been used for the initial pheromone trails in our proposed study as follows:

1. Creating a ring topology. Initially, the least cost connection link  $(i, j)$  is identified from the beginning of the list and value change  $i \leftarrow j$  is processed. Subsequently, a greedy process is used to connect node  $i$  with the new node  $j$ ; this process is iterated until all network nodes are processed. The initial and final path nodes constituting the path are created to construct the ring structure. The precise value of the ring structure all-terminal reliability is computed using Equation (5) [14] as follows.

$$R(G_x) = p^{|N|+|N|} p^{|N-1|} q. \quad (5)$$

2. Creating the network topology. Suppose the ring structure does not fulfil the all-terminal reliability threshold specified as  $R(G_x) < R_0$ , the ring is extended by including new links until the reliability condition is satisfied. Here, let's use the greedy process to identify a link  $(i, j)$  from  $L \setminus L_x$ . It is assumed that the network structure has more than  $|N|+1$  links.

Moreover, one shared process while computing the  $R_{UB}$  (the upper limit for all-terminal reliability concerning  $G_x$ ) of potential networks is identifying minimum edge count (i. e.,  $l_{\min}$ ) that fulfils the reliability criterion. The upper limit computation was proposed by the study [15], is used for this study. The degree of node  $i$  is represented using  $d_i$ , while the connections are represented as:

$$R_{UB}(G_x) \leq 1 - \left[ \sum_{i=1}^{|N|} q^{d_i} \prod_{k=1}^{m_i} (1 - q^{d_{i-1}}) \prod_{k=m_i+1}^{i-1} (1 - q^{d_k}) \right]. \quad (6)$$

If  $R_0$  exceeds  $R_{UB}(G_x)$ , the greedy link creation process is iterated until  $R_{UB}(G_x)$  equals or exceeds  $R_0$ . Subsequently, network reliability  $R^*(G_x)$  is precisely forecasted using Monte Carlo (MC) simulation, first suggested by [16]. On other hand, in our research work uses the following algorithm for creating the first network structure [11].

Algorithm 1: Generating initial network topology [11]

**Step 1.**  $N_r \leftarrow \{\}$  and  $L_r \leftarrow \{\}$ .

**Step 2.** Select the link  $(i, j)$  with minimum cost from  $L$  and update  $N_r$ ,  $L_r$  and  $i$  as follows:

$N_r \leftarrow N_r \cup \{i\} \cup \{j\}$ ;  $L_r \leftarrow L_r \cup \{i, j\}$ ;  $i' \leftarrow i$ ;  $i \leftarrow j$ .

**Step 3.** Perform following steps until  $N_r = N$ :

**3.1.** Choose node  $j$  such that link cost between  $i$  and  $j$  is minimum in  $L \setminus L_r$ :  $j \leftarrow \arg \min \{c_{ij} | (i, j) \in L \setminus L_r\}$ .

**3.2.**  $N_r \leftarrow N_r \cup \{j\}$ ,  $L_r \leftarrow L_r \cup \{i, j\}$  and  $i \leftarrow j$ .

**Step 4.** Obtain the ring topology by connecting  $i$  and  $i'$ ,  $L_r \leftarrow L_r \cup \{i', i\}$  and  $L_x \leftarrow L_r$ .

**Step 5.** Calculate  $R(G_x)$  (Equation (5)).

**Step 6.** If  $R(G_x) < R_0$ , apply greedy link addition procedure until obtaining a reliable network, i.e.,  $R_{UB}(G_x) \geq R_0$ .

**Step 7.** Estimate the all-terminal network reliability,  $R^*(G_x)$ , by MC simulation.

**Step 8.** If  $R^*(G_x) < R_0$ , apply the greedy link addition procedure until  $R^*(G_x) \geq R_0$  and calculate  $f(G_x)$  (Equation (1)).

**Step 9.** Return network topology,  $G_x = (N, L_x)$ .

Where

–  $L_x$  – set of selected links,  $\{L_x = x_{ij} | x_{ij} = 1, i = 1, 2, \dots, |N-1|, j = i+1, \dots, |N|\}$ ;

–  $N_x$  – node set of ring topology;

–  $N_r$  – sample size in Monte Carlo simulation;

–  $L_r$  – link set of ring topology.

**Step II:**

1) Solution determination. The conventional transition process is crucial for algorithm structure to enhance solutions since this equation is based on moving an ant across nodes; therefore, such movements must be towards better solutions to avoid moving to sub-optimal nodes, which increases cost and solution time. The objective of using sub-paths for the transition system is to create a diversified solution set by traversing across nodes and adding a new value that regulates the exploration process since every term from the initial transition rule is aligned to the exploitation process. Every sub-path is recorded in a table, while repetitions are moved to an exclusion table to enhance diversity.

$$S_c(j) = \begin{cases} \arg \max_{l \in U} \left[ \tau_{ij}^\alpha [\eta_{ij}]^\beta [\phi_{ij}^k]^\delta \times \right. \\ \left. \times \left[ \frac{(\max D + 1)}{(d_i + 1)} \right]^{\phi_1} \right] \times & \text{if } q \leq q_0, \\ \left. \times \left[ \frac{(\max D + 1)}{(d_i + 1)} \right]^{\phi_2} \right] & \text{otherwise} \end{cases} \quad (7)$$

where:

–  $U$  – the set of unvisited nodes;

–  $\tau_{ij}$  – the amount of pheromone on the edge  $E_{ij}$ ;

–  $\eta_{ij}$  – inverse to the distance between nodes  $i, j$ ;

–  $\alpha, \beta, \gamma, \delta$  – control parameters,  $0 < \alpha < 1$  random value;

–  $d_i$  – degree of a node;

–  $\max D$  – maximum degree of nodes in the topology;

–  $\phi_1$  and  $\phi_2$  – the degrees of node coefficient;

–  $\phi_{il}^k = \frac{1}{\Psi_{il}^k}$ ,  $\Psi_{il}^k$  counts the maximum nodes traversed

starting at solution  $k$  and ending at  $l$  which are present in part or entirely on a previously recorded sub-path.

Hence, selecting additional nodes increases the selected sub-path  $\Psi_{il}^k$ , decreasing  $\phi_{il}^k$ . Hence, the likelihood of selecting later reduces, while the probability of traversing additional sub-paths increases. Considering edge  $E_{ij}$  and unvisited not  $j$  for ant  $k, J$  represents the random parameter provided based on the probability transition expression to a subsequent node  $p_{ij}^k$  [17], applied using the specified sub-paths:

$$J: p_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta [\phi_{ij}^k]^\delta \left[ \frac{(\max D + 1)}{(d_i + 1)} \right]^{\phi_1} \left[ \frac{(\max D + 1)}{(d_i + 1)} \right]^{\phi_2}}{\sum_{l \in N} [\tau_{il}]^\alpha [\eta_{il}]^\beta [\phi_{il}^k]^\delta \left[ \frac{(\max D + 1)}{(d_i + 1)} \right]^{\phi_1} \left[ \frac{(\max D + 1)}{(d_i + 1)} \right]^{\phi_2}} & \text{if } j \in U \\ 0 & \text{otherwise} \end{cases} \quad (8)$$



2) Updating local pheromone values. This part concerns updating the local pheromone values to regulate concentration to traverse new paths to provide solutions and avoid an optimal local trap; this process causes other ants to not favour this road. This stage is expressed using (9):

$$\tau_{ij} = (1-\rho) \cdot \tau_{ij} + \rho \cdot \tau_0, \quad (9)$$

where  $0 \leq \rho \leq 1$  is a user-specified variable designated evaporation coefficient; its value is specified in the parameter specifications table.

*Step III:* Updating global pheromone values. This portion concludes an iteration and updates pheromone values using solution quality identified during the present iteration. The global update method ensures that the most performing ant changes pheromone trails and associated values are within required limits. The update process is specified below:

$$\tau_{ij} = \left[ (1-\rho) \cdot \tau_{ij} + \Delta \tau_{ij}^{best} \right]_{\tau_{min}}^{\tau_{max}}, \quad (10)$$

where

$\tau_{min}$  – minimum boundary value specified for the pheromone;

$\tau_{max}$  – maximum boundary value specified for the pheromone;

$[x]_b^a$  – performs a mathematical operation specified below:

$$[x]_b^a = \begin{cases} a & \text{if } x > a, \\ b & \text{if } x < b, \\ x & \text{otherwise;} \end{cases} \quad (11)$$

and  $\Delta \tau_{ij}^{best}$  is

$$\Delta \tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \text{if edge}(i, j) \text{ is part of the best tour;} \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

where  $L_{best}$  is the tour length corresponding to the best ant. Based on algorithm design, this might represent the most optimal trip identified during the present iteration iteration-best,  $L_{ib}$  – or the most optimal solution identified during the entire execution process best-so-far, or a combination of these two aspects [7].

Pheromone bound  $\tau_{min}$  and  $\tau_{max}$  are usually identified experimentally and regulated to suit the problem [18]. Nevertheless, specific recommendations are given for specifying  $\tau_{min}$  and  $\tau_{max}$  using analytical aspects [19]. The pseudo-code of the formulated EACS approach is specified below:

Algorithm 2: Steps of EACS algorithm

**Input:**  $\alpha, \beta, \rho, \tau, \eta, \varphi_1, \varphi_2$

**Output:** Best Solution

Build initial solution using algorithm 1

Assign  $\tau_{ij} = \tau_0 = 1/f(G_x)$

**For**  $t=1$ : max iteration

**For**  $n=1$ : max ants

topology = {}

select random starting node

**Loop**

**If**  $q < q_0$

Use Eq. (7) to select add new edge  $e$

Topology  $\leftarrow e$

**Else**

Use Eq. (6) to select add new edge

Topology  $\leftarrow e$

**End**

**If** topology satisfy constrain (3)

Calculate  $R_{UB}$  using Eq. (6)

**If**  $R_{UB} > R_0$

Calculate  $R^*(G_x)$  using MC

**If**  $R^*(G_x) > R_0$

Save solution and continue to next ant

**End**

**End**

**End**

**End**

Update local pheromone using Eq. (9)

Apply global update using Eq. (10), (11), (12)

Update best sub-path table

**End**

**End**

Report Best Solution

## 5. Results of Reliable Communication Network Design (RCND) Problems

### 5. 1. Proposing to apply the updated version of the ACS algorithm that has used sub-paths for solving RCND problems

As indicated by the results of the analysis and critique of the related studies in the literature conducted in this study, the utility of the ACS algorithm became selected as a new trend to address RCND problems. The ACS algorithm is distinguished from the rest of the algorithms in the ACO family through the developed properties of its components through an amended transition rule, the introduction of a local update of pheromone, and the global pheromone update is only the best solution. In addition, a recent study [8] has suggested enhancing the design of this algorithm with the usage of a brand new idea known as sub-paths for increasing diversity. All these features in addition to the match its components with the nature of the problem have led to increasing the efficiency of its performance in achieving the best solutions.

### 5. 2. Implement the enhanced ant colony system (EACS) algorithm to solve the RCND problem and evaluate its solutions with other algorithms

This section discusses the performance assessment of the proposed approach using two settings. The first setting concerns computational experiments based on three sets comprising eighteen small, three medium, and three large networks. The second setting discusses the proposed approach and the current state-of-the-art approaches:

a) Implement the enhance ant colony system (EACS) algorithm on small-sized networks.

This section discusses the performance assessment of the proposed technique using eighteen small networks. The data are specified in Table 1.

An optimal solution exists for the small-size network data specified above; hence, performance assessment of the proposed approach is based on Average Percentage Deviation (APD), which is specified as follows:

$$APD = 100 * (\text{Solution provided by the proposed approach} - \text{Optimal solution value}) / \text{Optimal solution value}. \quad (13)$$

Table 1

Data of small size problems

No.	Networks						
	Size	$N$	$L$	$p$	$R_0$	Optimal solutions	Network type
1	Small	6	15	0.9	0.9	231	FC
2	Small	6	15	0.9	0.95	254	FC
3	Small	6	15	0.95	0.95	227	FC
4	Small	7	21	0.9	0.9	189	FC
5	Small	7	21	0.9	0.95	205	FC
6	Small	7	21	0.95	0.95	185	FC
7	Small	8	28	0.9	0.9	208	FC
8	Small	8	28	0.9	0.95	247	FC
9	Small	8	28	0.95	0.95	179	FC
10	Small	9	36	0.9	0.9	239	FC
11	Small	9	36	0.9	0.95	286	FC
12	Small	9	36	0.95	0.95	209	FC
13	Small	10	45	0.9	0.9	131	FC
14	Small	10	45	0.9	0.95	153	FC
15	Small	10	45	0.95	0.95	121	FC
16	Small	14	21	0.9	0.9	1063	NFC
17	Small	16	24	0.95	0.95	1022	NFC
18	Small	20	30	0.95	0.9	596	NFC

Note: \* – FC, Fully Connected, \*\* – NFC, Not Fully Connected

Outcomes from small-size networks generated by the EACS approach are specified below (Table 2).

Table 2

Outcomes for small-size networks

Networks		APD
$N$	$L$	Proposed algorithm EACS
6	15	0
7	21	0
8	28	0
9	36	0
10	45	0.002
NFC problems		0
Ave. APD		0.0003
Ave. CPU (s.)		57.780

Note: \* – APD and CPU time for 3 problems (NFC), \*\* – APD and CPU time for 15 problems (FC)

The MC simulation sample  $nr$  is set to 3000 to fulfil the 95 % confidence level. The values in the table indicate that the proposed approach scored an average APD value of 0.0003 % (solutions comprised 15 FC and 3 NFC problems) and required 57.780 seconds of average CPU time.

b) Implement the enhance ant colony system (EACS) algorithm on medium-sized networks.

This section describes the performance assessment of the proposed approach determined for three medium-sized networks. The problems are specified in Table 3.

The proposed EACS approach provides the best topologies for medium-sized networks, as specified (Table 4).

The above table comprises the outcomes associated with medium-sized problems since these results comprise identifying the optimal topology, the overall value associated with the objective function  $f(G_x)$ , along with  $R^*(G_x)$  estimation

for a 95 % confidence interval concerning the all-terminal reliability of  $f(G_x)$ . The outcomes indicate that a 15-size network has  $f(G_x)$  value of 251,  $R^*(G_x)$  value of 0.950 and the confidence interval is (0.943, 0.953). Similarly, outcomes associated with sizes 20 and 25 are specified below:

–  $f(G_x)$  are 153 and 237,  $R^*(G_x)$  are 0.949 and 0.945, confidence intervals (0.946, 0.953) and (0.944, 0.947), respectively.

c) Implement the enhance ant colony system (EACS) algorithm on large-sized networks.

Table 3

Data of moderate size problems

No.	Networks						
	Size	$N$	$L$	$p$	$R_0$	Optimal solutions	Network type
1	Moderate	15	105	0.9	0.95	N/A	FC
2	Moderate	20	190	0.95	0.95	N/A	FC
3	Moderate	25	300	0.95	0.9	N/A	FC

Note: \* – FC, Fully connected, \*\* – N/A, Unavailable

Table 4

Optimal topologies for medium-sized networks

Networks		Best topology	$f(G_x)$	$R^*(G_x)$	Confidence Interval*
$N$	$L$				
15	105	(1, 5) (2, 3) (2, 10) (2, 14) (3, 12) (3, 14) (4, 11) (4, 12) (5, 8) (5, 13) (6, 11) (6, 15) (7, 10) (7, 12) (7, 15) (8, 10) (8, 15) (9, 13) (12, 14) (1, 6) (1, 9)	251	0.9515	[0.947, 0.956]
20	190	(1, 4) (1, 5) (1, 9) (2, 11) (2, 16) [2, 19] (3, 8) (3, 9) (4, 6) (4, 8) (5, 14) (6, 13) (7, 10) (7, 14) (9, 16) (10, 15) (11, 12) (11, 17) (11, 20) (12, 18) (13, 16) (15, 17) (18, 19) (18, 20)	153	0.9525	[0.946, 0.959]
25	300	(5, 20) (14, 22) (1, 4) (3, 7) (4, 6) (8, 17) (9, 19) (9, 21) (11, 18) (11, 23) (13, 16) (13, 24) (15, 16) (17, 22) (24, 25) (1, 12) (2, 10) (2, 18) (3, 10) (5, 12) (6, 15) (7, 19) (8, 21) (10, 23) (13, 20) (14, 25)	237	0.9045	[0.893, 0.916]

Note: \* – 95 % Confidence level

This section discusses the performance assessment of the proposed approach determined using three large networks. Problem details are specified in Table 5.

Table 5

Data of large size problems

No.	Networks						
	Size	$N$	$L$	$p$	$R_0$	Optimal solutions	Network type
1	Large	30	435	0.90	0.95	N/A	FC
2	Large	40	780	0.90	0.90	N/A	FC
3	Large	50	1225	0.95	0.90	N/A	FC

Note: \* – FC, Fully connected, \*\* – N/A, Unavailable

Best topologies concerning large networks were identified using the proposed EACS approach; the data are presented in Table 6 as follows.

Table 6

Best topologies for large-sized networks

Networks		Best topology	$f(G_x)$	$R'(G_x)$	Confidence Interval*
$N$	$L$				
30	435	(1, 14) (1, 24) (1, 26) (3, 5) (3, 25) (3, 27) (4, 17) (5, 18) (7, 13) (8, 15) (8, 29) (9, 21) (10, 11) (10, 12) (10, 25) (11, 17) (12, 15) (12, 29) (13, 19) (13, 20) (14, 19) (14, 30) (15, 28) (16, 19) (16, 30) (17, 28) (18, 25) (18, 30) (20, 21) (20, 23) (21, 22) (22, 23) (22, 29) (26, 27) (27, 28) (2, 4) (2, 8) (5, 6) (6, 24) (9, 11) (9, 26) (23, 24) (6, 28) (7, 8) (7, 12)	341	0.9528	[0.9467, 0.9589]
40	780	(1,23)(1,30)(1,38)(2,8)(2,12)(2,14)(2,15)(2,25)(3,6)(3,13)(3,22)(4,21)(4,35)(4,38)(5,6)(5,7)(5,13)(5,18)(7,33)(7,38)(8,20)(8,32)(8,33)(8,37)(9,17)(9,33)(10,25)(10,32)(11,21)(11,36)(11,40)(12,29)(12,39)(13,17)(13,40)(14,26)(14,34)(15,25)(15,31)(16,27)(16,37)(16,40)(17,23)(17,31)(18,20)(18,21)(19,23)(19,25)(19,30)(20,26)(21,22)(21,29)(22,28)(22,32)(24,27)(24,32)(24,36)(25,37)(26,36)(28,30)(30,34)(30,37)(31,35)(32,39)(33,34)(33,39)	559	0.9011	[0.892, 0.9102]
50	1225	(1,4)(1,18)(1,45)(2,10)(2,36)(3,6)(3,48)(3,50)(4,27)(4,34)(5,13)(5,21)(5,30)(6,15)(7,31)(7,34)(7,41)(8,11)(8,39)(9,11)(9,28)(9,38)(9,47)(10,18)(10,20)(12,27)(12,33)(13,15)(13,44)(13,47)(14,26)(14,33)(14,37)(15,22)(15,35)(16,35)(16,46)(17,19)(17,21)(17,26)(18,30)(19,21)(19,25)(19,34)(20,39)(22,31)(22,39)(23,32)(23,42)(24,43)(24,44)(25,29)(25,36)(28,29)(29,37)(30,40)(31,40)(32,38)(32,45)(32,50)(34,49)(41,48)(42,48)(43,46)(43,49)	368	0.9077	[0.8979, 0.9175]

Note: \* – 95 % Confidence level

This table includes outcomes associated with large network sizes (30, 40, and 50). The outcomes for such networks are specified below:

–  $f(G_x)$  are 341, 559, and 368,  $R'(G_x)$  are 0.9528, 0.9011, and 0.9077, and confidence intervals are [0.9467, 0.9589], [0.8908, 0.9102], and [0.8979, 0.9175] respectively.

d) Benchmarking the proposed algorithm EACS with the current state-of-the-art techniques.

This section describes how the proposed algorithm compares to the present state-of-the-art techniques designated  $GABB_1$  and  $GABB_2$  for addressing RCND proposed by [3]. Tables 7–9 have presented the results of those comparisons as follows.

Table 7

Computational results for proposed algorithm EACS and other algorithms ( $GABB_1$ ,  $GABB_2$ ) for small-sized networks

Networks		APD		
$N$	$L$	Proposed algorithm EACS	$GABB_1$	$GABB_2$
6	15	0	0.033	0
7	21	0	0.055	0
8	28	0	0.022	0
9	36	0	0.087	0.003
10	45	0.002	0.041	0
NFC problems	–	0	0.046	0.024
Ave. APD	–	0.0003	0.047	0.005
Ave. CPU (s.)	–	57.780	314.20	68.63

Note: \* – APD and CPU time for 3 problems (NFC), \*\* – APD and CPU time for 15 problems (FC)

Comparison outcomes for the proposed algorithm EACS and the latest techniques ( $GABB_1$ ,  $GABB_2$ ) specified in the literature to address RCND. The outcomes indicate that the proposed algorithm is superior to the existing methods for solution of 18 small networks (comprising 15 FC and 3 NFC problems). The proposed algorithm scored an average APD of 57.780 % using a 57.780 second CPU time. In contrast, the  $GABB_1$  and  $GABB_2$  techniques scored 0.047 and 0.005 average APD values using 314.20 seconds and 68.63 seconds CPU time, respectively. Other contrasts between the outcomes determined using the proposed approach and those from other approaches for addressing medium-sized networks are specified in Table 8.

Table 8

Computational results for proposed algorithm EACS and other algorithms ( $GABB_1$ ,  $GABB_2$ ) for moderate-sized networks

Networks		Best Cost		
$N$	$L$	Proposed algorithm EACS	$GABB_1$	$GABB_2$
15	105	251	283	254
20	190	153	172	157
25	300	237	266	244
Ave. CPU (min.)	–	7.435	42.66	9.00

The proposed algorithm EACS also improves optimal RCND costs identified using  $GABB_1$  and  $GABB_2$  for medium-sized networks (15, 20, and 25). The proposed algorithm provides the best cost values of 251, 153, and 237 for 15, 20, and 25 size networks, consuming 7.435 minutes of average CPU time. Additionally, the  $GABB_1$  technique arrived at best cost values of 283, 172, and 266 using 42.66 minutes of average CPU time. The corresponding values for  $GABB_2$  are 254, 157, and 244, with 9.00 minutes CPU time.

Finally, the outcomes determined for the proposed EACS and existing  $GABB_1$  and  $GABB_2$  techniques for large networks (30, 40, and 50) are specified in Table 9.

The results in Table 9 show that the EACS algorithm has got values of best cost were 341, 559, and 368 for the networks 30, 40, and 50 respectively within average CPU was 423.655. minutes Whilst the results of those networks that obtained by using the algorithms  $GABB_1$

and  $GABB_2$  were 380, 414, and 328 for sizes 30, 40, and 50 respectively within average CPU was 770.75 minutes by using  $GABB_1$  and average CPU was 470.59 minutes by using  $GABB_2$ .

Table 9

Computational results for proposed algorithm EACS and other algorithms ( $GABB_1$ ,  $GABB_2$ ) for large-sized networks

Networks		Best Cost		
$N$	$L$	Proposed algorithm EACS	$GABB_1$	$GABB_2$
30	435	341	380	342
40	780	559	414	363
50	1225	368	328	291
Ave. CPU (min.)	–	423.655	770.75	470.59

## 6. Discussion of experimental results of Reliable Communication Network Design (RCND) Problems

The basic algorithm used in this work belongs to the metaheuristic algorithms of the swarm intelligence which is called the ACS algorithm. As mentioned in the objectives section of this study, the first objective was to investigate the RCND problems, and the techniques used to address them. The outcome of this goal has led to the identification of new directions by identifying current knowledge gaps and issues in the RCND problems, whether it's domain solutions in small, medium, and large networks which still open research, and need to be optimized, or in the techniques used to solve them. This investigation indicates that the ACS algorithm has not been implemented to solve RCND problems. Achievement of this objective was achieved through in-depth analysis and critique of the results of the related studies in the literature.

Regarding the second objective of this study, there are several advantages in the ACS algorithm that have made it required in industries among them their accuracy, reliability, and efficiency, and can perform the same task repeatedly [20]. Moreover, the proposed algorithm EACS is that its components are congruent with the characteristics of the problem to be solved. Although all those advantages in the proposed algorithm, it can readily fall into a locally optimal ambush. All of these features made its performance superior to those algorithms in the literature in sizes small and medium of RCND data. As for large networks sizes 30, 40, and 50, the performance of the EACS algorithm has superior to  $GABB_1$  and  $GABB_2$  for size 30 networks, where

it determines 341 as the best cost. Comparatively,  $GABB_1$  and  $GABB_2$  have the best cost values of 380 and 342 respectively. On other hand, the proposed algorithm did not exceed the No Free Lunch (NFL) theorem proposed by [21] established that algorithm performance is not superior universally for all problems. Hence, let's devise an algorithm that performs optimally for most problems but is not comprehensive for every problem which is evident from the results obtained for the size of the networks 40, and 50, where the EACS algorithm obtained 559 and 368 for sizes 40, and 50 respectively, while the algorithms  $GABB_1$  and  $GABB_2$  have achieved the best cost for these sizes were 414, and 328 respectively.

Local search helps support the utilization mechanism of the EACS algorithm and can therefore represent an extension of this research work. For example, instead of performing a local search on all parts of the solution, it is possible to select a random part of the solution and perform a local search on them.

On other hand, this work can be developed by examining the main issue of these metaheuristic algorithms: how to achieve a balance between exploration and exploitation mechanisms.

## 7. Conclusions

1. The proposed approach used in this research work belongs to the ACO family of algorithms developed to solve the path problems, which is the main difference from the recent approach that was compared with it.

2. The proposed algorithm ACS has achieved the optimal solutions for 4 instances from 5 instances small sizes networks that are fully connected (FC). And the optimal solutions for 3 instances out of 3 instances small sizes networks that are not fully connected (NFC) problems ( $n=14$ , 16, and 20). In another way, the proposed algorithm EACS has achieved the rate of 87.5 % from all optimal solutions. Regarding the current solutions of the moderate size networks ( $n=15$ , 20, and 25) in the literature, the EACS algorithm has been able to improve upon their solutions. For the results of 3 large network instances ( $n=30$ , 40 and 50), the proposed algorithm EACS improved the current solution with a network size of  $n=30$ , and the remaining network size results were satisfactory compared with the recent related works. On the other hand, the CPU average for all sizes of network cases with the proposed algorithm EACS was better than the rest of the algorithms for the associated works.

## References

1. Watcharasitthiwat, K., Wardkein, P. (2009). Reliability optimization of topology communication network design using an improved ant colony optimization. Computers & Electrical Engineering, 35 (5), 730–747. doi: <https://doi.org/10.1016/j.compeleceng.2009.02.006>
2. Ozkan, O., Ermis, M. (2013). Reliable communication network design with metaheuristics. The International IIE Conference. Available at: [https://www.researchgate.net/profile/Omer-Ozkan/publication/263663161\\_Reliable\\_communication\\_network\\_design\\_with\\_metaheuristics/links/5b064f4b4585157f870937c2/Reliable-communication-network-design-with-metaheuristics.pdf](https://www.researchgate.net/profile/Omer-Ozkan/publication/263663161_Reliable_communication_network_design_with_metaheuristics/links/5b064f4b4585157f870937c2/Reliable-communication-network-design-with-metaheuristics.pdf)
3. Ozkan, O., Ermis, M., Bekmezci, I. (2019). Reliable communication network design: The hybridisation of metaheuristics with the branch and bound method. Journal of the Operational Research Society, 71 (5), 784–799. doi: <https://doi.org/10.1080/01605682.2019.1582587>
4. Nesmachnow, S., Cancela, H., Alba, E. (2007). Evolutionary algorithms applied to reliable communication network design. Engineering Optimization, 39 (7), 831–855. doi: <https://doi.org/10.1080/03052150701503553>



5. Yeh, W.-C., Lin, Y.-C., Chung, Y. Y., Chih, M. (2010). A Particle Swarm Optimization Approach Based on Monte Carlo Simulation for Solving the Complex Network Reliability Problem. *IEEE Transactions on Reliability*, 59 (1), 212–221. doi: <https://doi.org/10.1109/tr.2009.2035796>
6. Fatimah MohamadAyop, S., Shahizan Othman, M., Mi Yusuf, L. (2020). Ant Colony Optimization Using Different Heuristic Strategies for Capacitated Vehicle Routing Problem. *IOP Conference Series: Materials Science and Engineering*, 864 (1), 012082. doi: <https://doi.org/10.1088/1757-899x/864/1/012082>
7. Dorigo, M., Birattari, M., Stutzle, T. (2006). Ant Colony Optimization. *IEEE Computational Intelligence Magazine*, 1 (4), 28–39. doi: <https://doi.org/10.1109/ci-m.2006.248054>
8. Mutar, M. L., Burhanuddin, M. A., Hameed, A. S., Yusof, N., Mutashar, H. J. (2020). An efficient improvement of ant colony system algorithm for handling capacity vehicle routing problem. *International Journal of Industrial Engineering Computations*, 11, 549–564. doi: <https://doi.org/10.5267/j.ijiec.2020.4.006>
9. Omar, A. H., Naim, A. A. (2021). New crossover via hybrid ant colony system with genetic algorithm and making study of different crossover for TSP. *Journal of Theoretical and Applied Information Technology*, 99 (20), 4824–4836. Available at: <http://www.jatit.org/volumes/Vol99No20/14Vol99No20.pdf>
10. Agárdi, A., Kovács, L., Bányai, T. (2021). The Fitness Landscape Analysis of the Ant Colony System Algorithm in Solving a Vehicle Routing Problem. *Academic Journal of Manufacturing Engineering*, 19 (2), 85–89. Available at: [https://ajme.ro/PDF\\_AJME\\_2021\\_2/L11.pdf](https://ajme.ro/PDF_AJME_2021_2/L11.pdf)
11. Dengiz, B., Altıparmak, F., Belgin, O. (2010). Design of reliable communication networks: A hybrid ant colony optimization algorithm. *IIE Transactions*, 42 (4), 273–287. doi: <https://doi.org/10.1080/07408170903039836>
12. Dengiz, B., Alabas-Uslu, C. (2015). A self-tuning heuristic for the design of communication networks. *Journal of the Operational Research Society*, 66 (7), 1101–1114. doi: <https://doi.org/10.1057/jors.2014.74>
13. Abd-El-Barr, M. (2006). Design and analysis of reliable and fault-tolerant computer systems. *World Scientific*, 464. doi: <https://doi.org/10.1142/p457>
14. Jan, R.-H. (1993). Design of reliable networks. *IEEE International Conference on Communications*, 20, 25–34. doi: <https://doi.org/10.1109/icc.1992.268264>
15. Jan, R.-H., Hwang, F.-J., Chen, S.-T. (1993). Topological optimization of a communication network subject to a reliability constraint. *IEEE Transactions on Reliability*, 42 (1), 63–70. doi: <https://doi.org/10.1109/24.210272>
16. Yeh, W. C. (1994). A New Monte Carlo Method for the Network Reliability. *Proceedings of the First International Conference on Information Technology and Applications (ICITA 2002)*. Available at: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.11.8926&rep=rep1&type=pdf>
17. Dorigo, M., Gambardella, L. M. (1997). Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1 (1), 53–66. doi: <https://doi.org/10.1109/4235.585892>
18. Socha, K., Knowles, J., Sampels, M. (2002). A MAX-MIN Ant System for the University Course Timetabling Problem. *Lecture Notes in Computer Science*, 1–13. doi: [https://doi.org/10.1007/3-540-45724-0\\_1](https://doi.org/10.1007/3-540-45724-0_1)
19. Stützle, T., Hoos, H. H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16 (8), 889–914. doi: [https://doi.org/10.1016/s0167-739x\(00\)00043-1](https://doi.org/10.1016/s0167-739x(00)00043-1)
20. Vochozka, M., Horák, J., Krulický, T. (2019). Advantages and Disadvantages of Automated Control Systems (ACS). *Digital Age: Chances, Challenges and Future*, 416–421. doi: [https://doi.org/10.1007/978-3-030-27015-5\\_50](https://doi.org/10.1007/978-3-030-27015-5_50)
21. Wolpert, D. H., Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1 (1), 67–82. doi: <https://doi.org/10.1109/4235.585893>