Asia Mathematika
Volume: 10 Issue: 1 (2026) Pages: 64 – 72
Available online at **www.asiamath.org**

Enhancing multi-terminal network reliability with cost constraints using differential evolution and particle swarm optimization algorithms

Haider Saleh Howeidi[1*], **Amir Sahib Mohammad Ali**[2] **and Zahir Abdul Haddi Hassan**[3]
[1]Department of Mathematics, College of Education for Pure Sciences, University of Babylon, Iraq.
[2]Ministry of Education, Department of Mathematics, Babil Education Directorate, Babylon, Iraq.
[3]Department of Mathematics, College of Education for Pure Sciences, University of Babylon, Iraq.

Received: 06 Mar 2026 • **Accepted:** 07 May 2026 • **Published Online:** 30 May 2026

Abstract: In this study, we emphasize on reliability improvement of multi-terminal networks, that is, the complex of network nodes with inputs and outputs in flow of communication or energy with applications of the DE and PSO algorithms. The objective is to enhance the performance of the network within the bounds of a specified cost. The mutation, crossover, and selection based DE and the particle moving towards optimal solutions based PSO algorithms are modified to the unique needs of the multi-terminal network. These enhancements allow the algorithms to adapt component reliability effectively and therefore they address the dilemma of designing and operating such networks for reliability at an acceptable cost.

Key words: Multi-terminal networks, network reliability, differential evolution, particle swarm optimization, cost constraints

1. Introduction

Multi-terminal networks are important infrastructures that support a wide range of applications, such as telecommunications, power delivery, and information management, and are composed of complex arrangements of linked nodes and elements. The reliability of such networks, as their capacity to fulfill required functions under defined conditions, should be guaranteed, which requires the development of super optimization tools that can combine the best solutions with the least cost. The present study is focused on analyzing the performance of optimization procedures, namely Differential Evolution and Particle Swarm Optimization, on reliability assessment of multi-terminal networks. The aim of this work is to adapt the above algorithms to complex network requirements in order to provide a methodology for network design and operation balancing reliability and cost, so as to inform the design of better and more sustainable systems.

2. History and evolution of algorithms

The application of algorithms to optimize network reliability has its roots in the 1970s, with the development of the Genetic Algorithm in 1975 [1, 2]. Developed by John Holland, this algorithm was inspired by biological evolution for enhancing solutions to complex problems through natural selection simulation. Main algorithms of the subject:

- Genetic Algorithm (GA): introduced by John Holland in 1975, utilized to improve network reliability.

©Asia Mathematika, DOI: 10.5281/zenodo.20997173
*Correspondence: haider1saleh81@gmail.com

- Simulated Annealing: derived in 1983 by Scott Kirkpatrick among others, simulated heat dissipation to obtain optimal solutions for designing networks [3].
- Ant Colony Optimization: proposed in 1992 by Marco Dorigo, emulated ant behavior to find optimal paths [4].
- Grey Wolf Optimization: obtained in 2020, proposed by Seyedali Mirjalili, applied to improve network dependability [5].
- Harmony Search Algorithm: designed in 2021 by Zong Woo Geem, inspired from musical creation [6].
- Bat Algorithm: emerged in 2022, by Xin-She Yang, based on echolocation simulation [7].

Algorithms stated in the paper (DE and PSO):

- Differential Evolution: first proposed in 1995 by Rainer Storn and Kenneth Price [8].
- Particle Swarm Optimization: conceived in 1995 by James Kennedy and Russell Eberhart [9].

3. Key definitions

Definition 3.1 (Network reliability)**.** Network reliability refers to the probability that a network system will perform its intended functions under specified operating conditions over a given period. It is typically calculated based on the reliability of individual components and the network's configuration [11].

Definition 3.2 (Multi-terminal network)**.** A multi-terminal network is a system comprising multiple nodes (connection points) and components, with specific nodes designated as inputs and others as outputs. It is used to model complex flows, such as communication or power transmission [12].

Definition 3.3 (Optimization)**.** Optimization is the process of finding the best solution to a problem within a set of constraints, such as maximizing network reliability while adhering to cost limits [13].

Definition 3.4 (Differential Evolution algorithm)**.** Differential Evolution (DE) is an evolutionary optimization algorithm that improves a population of candidate solutions through mutation, crossover, and selection operations, suitable for addressing complex non-linear problems [14].

Definition 3.5 (Particle Swarm Optimization algorithm)**.** Algorithm inspired by the behavior of natural swarms, where particles move in the search space based on their individual best positions and the global best position to achieve an optimal solution [15].

Definition 3.6 (Cost)**.** Cost refers to the financial or resource value required to develop or enhance network components, serving as a primary constraint in optimization processes [16].

The cost of component i is calculated using the function [17]:

$$C_i(R_i) = a_i \exp(b_i R_i)$$

where a_i and b_i are component-specific parameters, and R_i is the component's reliability. The total system cost is:

$$C_{\text{total}} = \sum C_i$$

Definition 3.7 (Component)**.** A component is an individual element within the network, such as a link or node, characterized by specific reliability and cost values that influence overall network performance [11].

Definition 3.8 (Search space)**.** The search space is the entire set of possible solutions explored by optimization algorithms to identify the optimal solution for a given problem [18].

Definition 3.9 (Fitness)**.** Fitness is a measure used to evaluate the quality of a candidate solution in optimization algorithms, such as network reliability within cost constraints [19].

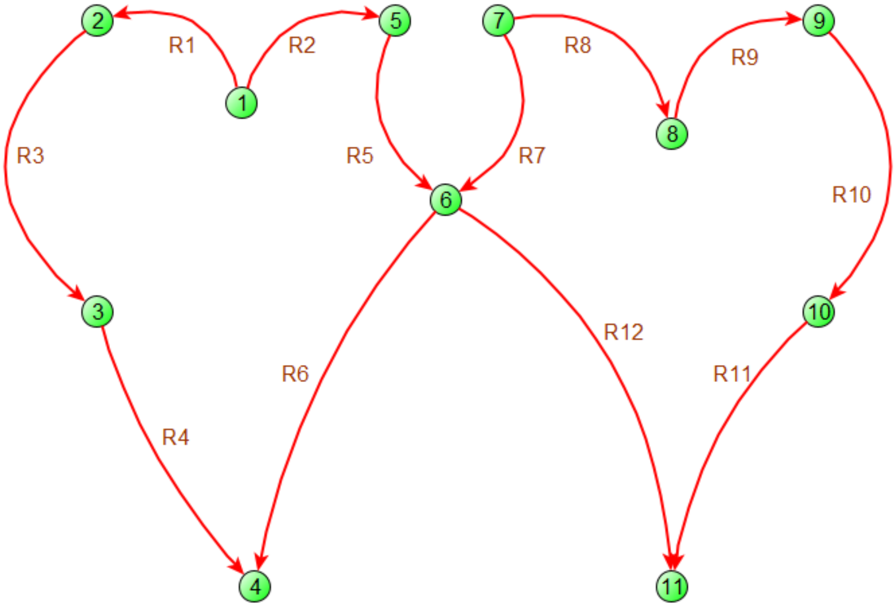
Definition 3.10 (Iteration)**.** An iteration is a single cycle in the optimization algorithm process, where candidate solutions are updated and evaluated to improve performance [2].

4. Study model

A 12-component multistate network has input and output terminal sets, with nodes 1 and 7 serving as inputs and nodes 4 and 11 serving as outputs (see Fig. 1), and the performance of the algorithms is tested for such a network to measure the processing time. In the problem, system reliability is maximized subject to predefined cost constraints, using Whale Algorithm and Genetic Algorithm. The network reliability at output node 4 is calculated using the following equation [11]:

$$\begin{aligned} R_{s4} &= 1-(1-R_6R_7)(1-R_1R_3R_4)(1-R_2R_5R_6) \\ &= R_6R_7+R_1R_3R_4+R_2R_5R_6-R_2R_5R_6R_7 \\ &\quad -R_1R_3R_4R_6R_7-R_1R_2R_3R_4R_5R_6 \\ &\quad +R_1R_2R_3R_4R_5R_6R_7 \end{aligned} \tag{1}$$

Figure 1. Multi-terminal network topology (nodes 1,7 inputs; nodes 4,11 outputs).



5. Methodology

5.1. Differential Evolution (DE) algorithm

DE works through the following steps [20, 21]:

1. Initialization: A population of candidate solutions (component reliabilities) is randomly generated.
2. Mutation: A mutant vector is created by adding the weighted difference between two population vectors to a third vector.
3. Crossover: On the basis of a crossover probability, the mutant vector and the original solution are mixed to create a trial vector.
4. Selection: If the trial vector results in a better objective value (higher reliability with cost constraints), then it replaces the original solution.
5. Iteration: Repeat the above steps for 50 evolution runs.

This algorithm is suitable for non-linear problems.

5.2. Particle Swarm Optimization (PSO) algorithm

In PSO, each particle represents a candidate solution, and the movement of every particle is dynamically adjusted to move towards its individual best location and the global best position [22, 23]:

1. Initialization: A swarm of particles (component reliabilities) is initialized at random positions and velocities.
2. Update velocity: At each iteration, the velocity of each particle is updated by applying inertia, cognitive and social components.
3. Update position: The position of each particle is updated according to its velocity, corresponding to a candidate new solution.
4. Evaluation: Each particle (reliability within cost limits) is evaluated; personal and global best positions are updated.
5. Iteration: The process repeats for 50 iterations.

5.3. Experimental setup

Both algorithms with the same initial conditions were executed on a single network:

- Initial R_{s4}: reliability 0.6289, cost 425.00
- Initial R_{s11}: reliability 0.5376, cost 570.00
- Initial total system cost: 785.00
- Iterations: 50

Performance indices included optimized reliability and cost of R_{s4} and R_{s11}, total system cost, and execution time.

6. Results

6.1. Differential Evolution (DE)

After 50 iterations DE reached the following results:

- Optimized R_{s4} reliability: 0.9663 (54% higher)
- Optimized R_{s11} reliability: 0.8387 (56% higher)
- Optimized R_{s4} cost: 497.82 (within 500 limit)
- Optimized R_{s11} cost: 597.03 (within 600 limit)
- Optimized system cost: 876.27 (11.5% higher)
- Execution time: 1.51 seconds

Component-level results (Table 1) indicate diverse reliability enhancement.

Table 1. DE component-level results

Component	**Initial R**	**Optimized R**	**Initial Cost**	**Optimized Cost**
1	0.6000	0.9890	50.00	82.42
2	0.6500	0.4700	60.00	43.38
3	0.6800	0.9781	45.00	64.73
4	0.6200	0.9761	55.00	86.59
5	0.6700	0.6840	70.00	71.46
6	0.6100	0.4271	65.00	45.51
7	0.6900	0.8947	80.00	103.73
8	0.6600	0.4402	75.00	50.03
9	0.6000	0.6511	60.00	65.11
10	0.5300	0.5437	50.00	51.29
11	0.5400	0.4702	85.00	74.01
12	0.5800	0.8895	90.00	138.02

6.2. Particle Swarm Optimization (PSO)

After 50 iterations PSO resulted in:

- Optimized R_{s4} reliability: 0.9928 (58% increase)
- Optimized R_{s11} reliability: 0.9832 (83% increase)
- Optimized cost for R_{s4}: 499.89 (below limit 500)
- Optimized cost for R_{s11}: 599.60 (below limit 600)
- Optimum system cost: 905.99 (15.4% increase)
- Execution time: 0.34 seconds

Component-level results (Table 2) show enhanced reliabilities for components 6,7,9,12 and reduced for 2,5,8,11.

Table 2. PSO component-level results

Component	**Initial R**	**Optimized R**	**Initial Cost**	**Optimized Cost**
1	0.6000	0.9014	50.00	75.11
2	0.6500	0.4000	60.00	36.92
3	0.6800	0.9712	45.00	64.27
4	0.6200	0.6934	55.00	61.51
5	0.6700	0.4000	70.00	41.79
6	0.6100	0.9900	65.00	105.49
7	0.6900	0.9900	80.00	114.78
8	0.6600	0.4007	75.00	45.54
9	0.6000	0.8673	60.00	86.73
10	0.5300	0.6057	50.00	57.14
11	0.5400	0.4007	85.00	63.08
12	0.5800	0.9900	90.00	153.62

7. Discussion

7.1. Comparison of algorithm performance

- **Reliability:** PSO significantly outperforms DE at R_{s4} (0.9928 vs. 0.9663) and at R_{s11} (0.9832 vs. 0.8387). This superiority may be attributed to better exploration/exploitation balance in PSO.

- **Cost:** Both satisfy cost constraints ($R_{s4} \leq 500$, $R_{s11} \leq 600$). However, PSO yields a higher system cost (905.99 vs. 876.27), implying a reliability-cost trade-off. DE is more cost-efficient at the expense of reliability.

- **Execution time:** PSO is significantly faster (0.34 s vs. 1.51 s), demonstrating higher computational efficiency due to simpler velocity/position updates compared to DE's mutation and crossover.

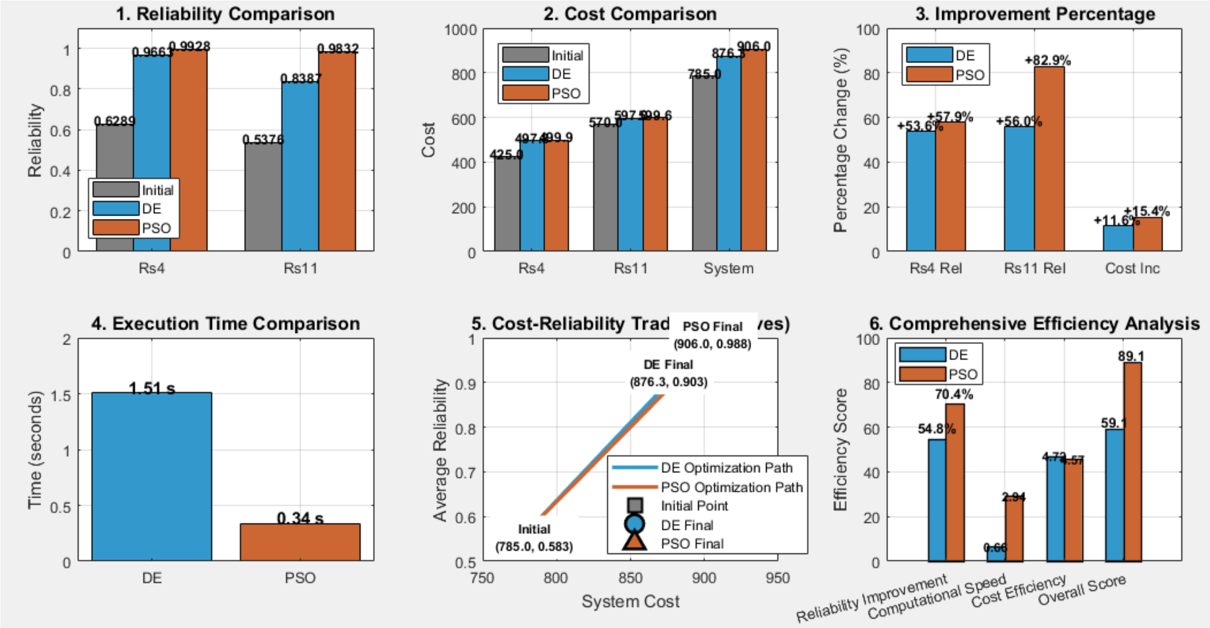
- **Component-level optimization:** PSO biases critical components (e.g., 6,7,12) to very high reliability, while DE distributes gains more evenly.

DE is based on intensive investigation of the search space by mutation and crossover, which avoids premature convergence but increases computational complexity. PSO uses individual and social learning to converge rapidly to near-optimal solutions, making it more suitable for network reliability optimization.

7.2. Impact

The strong dependability and running efficiency of PSO make sit an ideal candidate to be used in real-time or limited-resource network optimization applications. Nevertheless,the lower system cost of DE may be preferable in cost sensitive applications. This balancing act highlights the importance of optimizing the goals of the optimization (reliability versus cost) when determining an algorithm.

Figure 2. Reliability comparison between DE and PSO for R_{s4} and R_{s11}.



7.3. Restrictions

The work is confined to one network setup and one number of iterations (50). Some other network typologies or extra constraints such as maintenance expenses, could have different results. The performance may also depend on parameter settings of the algorithm,which are not given in the data.

8. Conclusion

The results showed that the PSO algorithm performed better than the DE algorithm in improving the availability of the multi-terminal networks, which are multi-terminal net works, complex systems of nodes interconnected to provide flows such as communication and energy. Both algorithms have been adapted to the new requirements of these net works, leading to substantial increases in component reliability under cost constraints. The Particle Swarm Optimization algorithm achieved higher computational efficiency be cause of its rapid convergence, while the Differential Evolution algorithm took a more conservative strategy in the cost management. In terms of future work, this study paves the way for investigating hybrid algorithms that exploit the advantages of both methods to further maximize the reliability of the multi-terminal networks. The analysis could be generalized to more complex network structures or to additional constraints, e.g., on maintenance costs or energy consumption. These results are expected to lead to the design of novel networks in telecommunication and energy transmission with improved efficiency and sustainability in real-world applications with continuous updates of algorithms dedicated to these networks.

Acknowledgments

The authors thank the Department of Mathematics, University of Babylon for supporting this research.

References

[1] Kumar A, Pathak RM, Gupta YP. Genetic-algorithm-based reliability optimization for computer network expansion. IEEE Transactions on Reliability 1995; 44(1): 63-72.

[2] Mitchell M. An Introduction to Genetic Algorithms. Cambridge, MA, USA: MIT Press, 1998.

[3] Vijayaraghavan A. Effective solutions and algorithmic approaches for solving combinatorial optimization problems using metaheuristic techniques. (n.d.)

[4] Dorigo M, Birattari M, Stutzle T. Ant colony optimization. IEEE Computational Intelligence Magazine 2007; 1(4): 28-39.

[5] Makhadmeh SN, et al. Recent advances in Grey Wolf Optimizer, its versions and applications. IEEE Access 2023; 12: 22991-23028.

[6] Nasir M, et al. Harmony search algorithm and fuzzy logic theory: An extensive review from theory to applications. Mathematics 2021; 9(21): 2665.

[7] Alyasseri ZAA, et al. Recent advances of bat-inspired algorithm, its versions and applications. Neural Computing and Applications 2022; 34(19): 16387-16422.

[8] Price K, Storn RM, Lampinen JA. Differential Evolution: A Practical Approach to Global Optimization. Berlin, Germany: Springer, 2006.

[9] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks IV; 1995; Perth, WA, Australia. pp. 1942-1948.

[10] De Pablo JJ, et al. The materials genome initiative, the interplay of experiment, theory and computation. Current Opinion in Solid State and Materials Science 2014; 18(2): 99-117.

[11] Howeidi HS, Hassan ZAH. A new method to compute the reliability importance of components in reliability system with independent identical units. AIP Conference Proceedings 2023; 2834(1).

[12] Gomory RE, Hu TC. Multi-terminal network flows. Journal of the Society for Industrial and Applied Mathematics 1961; 9(4): 551-570.

[13] Deb K, et al. Reliability-based optimization using evolutionary algorithms. IEEE Transactions on Evolutionary Computation 2009; 13(5): 1054-1074.

[14] Mohamed AW. A novel differential evolution algorithm for solving constrained engineering optimization problems. Journal of Intelligent Manufacturing 2018; 29: 659-692.

[15] Zhang Y, Wang S, Ji G. A comprehensive survey on particle swarm optimization algorithm and its applications. Mathematical Problems in Engineering 2015; 2015: 931256.

[16] Alkhanaik EN, et al. Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues. Journal of Systems and Software 2016; 113: 1-26.

[17] Herzog I. Theory and practice of cost functions. 2013.

[18] Zhang W. State-Space Search: Algorithms, Complexity, Extensions, and Applications. New York, NY, USA: Springer, 1999.

[19] Wu SH, Zhan ZH, Zhang J. SAFE: Scale-adaptive fitness evaluation method for expensive optimization problems. IEEE Transactions on Evolutionary Computation 2021; 25(3): 478-491.

[20] Karaboga D, Ökdem S. A simple and global optimization algorithm for engineering problems: differential evolution algorithm. Turkish Journal of Electrical Engineering and Computer Sciences 2004; 12(1): 53-60.

[21] Deng W, et al. An improved differential evolution algorithm and its application in optimization problem. Soft Computing 2021; 25: 5277-5298.

[22] Wang D, Tan D, Liu L. Particle swarm optimization algorithm: an overview. Soft Computing 2018; 22(2): 387-408.

[23] Marini F, Walczak B. Particle swarm optimization (PSO). A tutorial. Chemometrics and Intelligent Laboratory Systems 2015; 149: 153-165.