International Journal of Innovative Research & Growth
E-ISSN: 2455-1848
homepage: http://www.ijirg.com/ijirg/
Volume: 15 Issue: 4, October 2026

A Mathematical Framework for Interconnection Network Optimization Using an Improved Genetic Algorithm Under Reliability and Cost Constraints

Sushil Kumar Dwivedi[1*], Rakesh Kumar Katare[2]
[1]Research Scholar, Department of Computer Science, APS University, Rewa-486003, M.P., India
[2]Professor, Department of Computer Science, APS University, Rewa-486003, M.P., India

CORRESPONDING AUTHOR

Sushil Kumar Dwivedi
e-mail: sushil_dwid@yahoo.co.in

KEYWORDS

Interconnection Network, Topology Optimization, Improved Genetic Algorithm, Network Reliability, All-Terminal Reliability, Constrained Optimization.

ARTICLE DETAILS

Received 05 July 2026; revised 15 September 2026; accepted 08 October 2026

DOI: 10.26671/IJIRG.2026.4.15.299

CITATION
Dwivedi, S. K., Katare, R. K. (2026). A Mathematical Framework for Interconnection Network Optimization Using an Improved Genetic Algorithm Under Reliability and Cost Constraints. *Int J Innovat Res Growth*, 15(4), 154081-154089. DOI

This work may be used under the terms of the Creative Commons License.

Abstract

Interconnection-network design requires a trade-off among reliability, redundancy, connectivity, and implementation cost. This paper presents an Improved Genetic Algorithm (IGA) for reliability- and cost-constrained topology optimization. The network is modeled as a probabilistic graph whose processors are vertices and communication links are candidate edges with cost and reliability attributes. The IGA combines reliability-guided initialization, adaptive crossover and mutation, tournament selection, elitism, and feasibility-preserving repair. To address the limitation of a link-average reliability surrogate, the revised study uses estimated all-terminal reliability as an independent validation metric. Experiments are performed on standard Mesh, Torus, and Hypercube topologies with 64 and 256 nodes. The IGA is compared with a conventional Genetic Algorithm and a minimum-cost spanning-tree-based heuristic over five independent runs per configuration. Monte Carlo reliability estimates use 5,000 link-state trials for each optimized topology. The results show that the IGA provides competitive reliability under the same cost budget, with the clearest advantage on the 256-node Mesh and on the 64-node Hypercube, while the conventional GA is slightly better on some Torus instances. The results therefore support the usefulness of reliability-guided evolutionary search without claiming universal superiority.

1. Introduction

Interconnection networks constitute the communication backbone of parallel computers, distributed systems, network-on-chip platforms, switching fabrics, and communication infrastructures. Their topology affects communication delay, path diversity, fault tolerance, scalability, and deployment cost. Consequently, topology design can be formulated as a constrained combinatorial optimization problem in which selected communication links must provide adequate structural reliability while remaining within a resource budget [1]– [3].

Recent work by the authors established a mathematical framework for analyzing structural connectivity and complexity characteristics of processor interconnection networks, providing a structural foundation for subsequent topology optimization [4]. A dense topology can provide alternative communication routes and improved fault tolerance, but it also increases cost and implementation complexity. A sparse topology reduces cost but can become vulnerable to link failures or degree deficiencies. Exact network-reliability evaluation is itself computationally difficult for large graphs, motivating approximation and simulation techniques [5], [6].

Genetic Algorithms (GAs) are attractive for topology design because they can explore large discrete search spaces without requiring differentiability [7], [8]. However, conventional genetic operations can generate disconnected, degree-deficient, or budget-infeasible topologies. Constraint handling, initialization, and preservation of high-quality feasible solutions therefore become important design issues [9], [10]. Improved genetic approaches have previously been applied to reliable communication-network topology optimization [11], [12]. The original version of this work emphasized mathematical formulation and theoretical properties. The revision addresses three major validation issues. First, the proposed IGA is evaluated on standard Mesh, Torus, and Hypercube benchmark topologies at 64 and 256 nodes. Second, the original reliability-aware surrogate is explicitly distinguished from an independent all-terminal reliability estimate obtained by Monte Carlo simulation. Third, the figures, captions, cross-references, and IEEE-style citations are revised and checked.

The main contributions of the revised paper are:

- a graph-theoretic formulation of reliability- and cost-constrained topology optimization;
- an IGA combining reliability-guided initialization, adaptive genetic operators, repair, and elitism;
- an explicit all-terminal reliability estimator for independent validation of optimized topologies;
- benchmark experiments on Mesh, Torus, and Hypercube networks with 64 and 256 nodes; and
- a comparison with a conventional GA and an MST-based heuristic using reliability, cost, fitness, and runtime metrics.

2. Related Work

Reliable network topology design has been studied using analytical, exact, heuristic, and evolutionary approaches. Foundational studies investigated large-scale topological optimization and reliability-constrained communication-network design [1], [2], [13]. Genetic-algorithm and local-search approaches subsequently addressed reliable network design and all-terminal reliability optimization [10], [14]– [16]. For interconnection networks, Tripathy et al. investigated reliability evaluation and genetic-algorithm-based topological optimization [11], [17], [18]. Ai *et al.* further investigated topology optimization of computer communication networks using an improved genetic algorithm [12]. These studies demonstrate the applicability of evolutionary optimization to reliability-aware network design. From a structural perspective, previous studies have examined connectivity and structural relationships in interconnection networks [19], [20]. The author's earlier work developed a mathematical framework for structural connectivity and complexity analysis of processor interconnection networks [4]. The present work builds upon this structural foundation by introducing an Improved Genetic Algorithm for topology optimization under reliability and cost constraints. More recent studies have also explored learning-based approaches for network reliability estimation [21], [22], motivating the use of an independent reliability evaluation rather than relying solely on an optimization surrogate.

3. Problem Formulation

3.1 Graph Model

The candidate interconnection network is modeled as an undirected graph

$$G = (V, E),$$

where $V = \{v_1, \dots, v_n\}$ denotes processors and $E = \{e_1, \dots, e_m\}$ denotes candidate communication links. Each link $(i, j) \in E$ has installation cost $c_{ij} > 0$ and working reliability $r_{ij} \in (0,1]$. A binary decision variable is defined as

$$x_{ij} = \begin{cases} 1, & \text{if link } (i,j) \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

The selected topology is $G_X = (V, E_X)$.

3.2 Cost and Structural Constraints

The total selected-link cost is

$$C(X) = \sum_{(i,j)\in E} c_{ij}\, x_{ij},$$

subject to

$$C(X) \le C_{\max}.$$

For a minimum degree requirement $d_{\min}$,

$$deg_X(v_k) \ge d_{\min}, \qquad k = 1, \dots, n,$$

and the selected topology must be connected.

3.3 Optimization-Oriented Reliability Surrogate

Exact all-terminal reliability can be computationally expensive to evaluate repeatedly inside an evolutionary search, particularly as network size increases [5], [6], [23]. Therefore, the optimization stage retains the surrogate introduced in the original formulation:

$$\hat{R}(X) = \omega_1 \bar{r}(X) + \omega_2 \Pi(X),$$

where $\bar{r}(X)$ is the average reliability of selected links and

$$\Pi(X) = \max\left(0, \frac{|E_X| - (n-1)}{m - (n-1)}\right)$$

for $m > n - 1$, and zero otherwise. The weights satisfy $\omega_1 + \omega_2 = 1$.

This quantity is explicitly treated as a *surrogate optimization score*, not as the probability that the complete network remains connected after random link failures. The distinction addresses the reliability concern raised by the reviewer.

3.4. Penalty-Based Fitness

The optimization fitness is

$$F(X) = \alpha \hat{R}(X) - \beta P_{cost}(X) - \gamma P_{deg}(X) - \delta P_{conn}(X),$$

where

$$P_{cost}(X) = \max\left(0, \frac{C(X) - C_{\max}}{C_{\max}}\right),$$

$$P_{deg}(X) = \frac{1}{n d_{\min}} \sum_{k=1}^{n} \max\left(0, d_{\min} - deg_X(v_k)\right),$$

and $P_{conn}(X)$ equals zero for a connected topology and one otherwise.

4. Improved Genetic Algorithm

4.1. Chromosome and Initialization

A chromosome is a binary vector $X = (x_1, \dots, x_m)$, with one gene per candidate link. The IGA uses a reliability-guided initialization strategy. A minimum-cost spanning-tree seed is first constructed, degree constraints are repaired, and additional links with high reliability-to-cost ratios are added while respecting the budget. Perturbed variants are then generated and repaired.

4.2. Adaptive Genetic Operators

Tournament selection is used. The IGA crossover probability changes from 0.90 to 0.70 over the run:

$$p_c(g) = p_c^{max} - \frac{g}{G_{max}}(p_c^{max} - p_c^{min}).$$

The mutation probability changes from 0.08 to 0.01:

$$p_m(g) = p_m^{min} + \left(1 - \frac{g}{G_{max}}\right)(p_m^{max} - p_m^{min}).$$

Elitism preserves the best two feasible chromosomes.

Algorithm 1 Improved Genetic Algorithm

Require: Candidate graph G, costs, reliabilities, $C_{\max}$, $d_{\min}$, population P, generations $G_{\max}$
Ensure: Best feasible topology X^*
1: Construct reliability-guided feasible initial population
2: Evaluate $F(X)$ and retain best feasible chromosome
3: **for** $g = 1$ to $G_{\max}$ **do**
4: Select parents using tournament selection
5: Apply adaptive crossover with probability $p_c(g)$
6: Apply adaptive mutation with probability $p_m(g)$
7: Project offspring onto a feasible base and add high-utility selected links within budget
8: Evaluate offspring fitness
9: Copy the best two chromosomes to the next generation
10: Update X^*
11: **end for**
12: **return** X^*

4.3. Feasibility-Preserving Repair

The conceptual repair sequence first establishes connectivity and minimum degree and then enforces the budget. For efficient implementation on the 256-node benchmarks, a deterministic minimum-cost connected base satisfying the degree constraint is constructed once for each benchmark. Each offspring is projected onto this feasible base, after which its selected non-base genes are considered in descending reliability-to-cost order and added while the budget permits. This implementation preserves feasibility while avoiding repeated expensive connectivity searches during every genetic operation.

5. All-Terminal Reliability Validation

The revised study independently evaluates each optimized topology using Monte Carlo simulation. For a selected edge e, its operational state is sampled independently with probability r_e. For trial s, let

$$I_s = \begin{cases} 1, & G_X^{(s)} \text{ is connected,} \\ 0, & \text{otherwise.} \end{cases}$$

The estimated all-terminal reliability is

$$\hat{R}_{AT}(X) = \frac{1}{N_s} \sum_{s=1}^{N_s} I_s.$$

The estimated standard error is

$$SE = \sqrt{\frac{\hat{R}_{AT}(1 - \hat{R}_{AT})}{N_s}},$$

and the approximate 95% confidence interval is $\hat{R}_{AT} \pm 1.96SE$.
This estimator directly answers whether all processors remain connected under stochastic link failures and is therefore a stronger reliability indicator than the average selected-link reliability. It is used only for final validation, avoiding repeated Monte Carlo computation inside the GA. The distinction between optimization surrogate and validation reliability is important for reproducibility and computational efficiency.

6. Experimental Methodology

6.1. Benchmark Topologies

Three standard interconnection-network structures were evaluated: Mesh, Torus, and Hypercube. The Mesh and Torus cases use 8×8 and 16×16 structures, corresponding to 64 and 256 nodes. The Hypercube cases use 6-dimensional and 8-dimensional structures, corresponding to 64 and 256 nodes. These benchmark families provide different degrees and path-redundancy characteristics. The selected Mesh, Torus, and Hypercube structures are standard benchmark topologies used to evaluate the optimization behavior of the proposed method. Because physical deployment-specific measurements were not available, link costs and reliability were generated using fixed random seeds. Thus, the experiments evaluate the algorithm on standard network structures rather than representing measurements from a deployed physical network.

6.2. Link Attributes and Budgets

To isolate topology optimization from application-specific physical data, link attributes were generated reproducibly using fixed seeds for each benchmark. Link costs were integer values in the range 1–5, while link reliabilities were generated in the interval $[0.90, 0.995)$. The minimum node degree was set to two. For each benchmark, the cost budget was set to 112% of the cost of the minimum-cost feasible base topology and rounded upward. The same budget and link attributes were supplied to all competing methods.

6.3. Algorithms and Parameters

The comparison included the proposed IGA, a conventional GA, and an MST-based heuristic. The IGA and conventional GA used population size 30 and 60 generations. Tournament size was three and elitism retained two solutions. The conventional GA used fixed crossover and mutation probabilities of 0.80 and 0.02, respectively. The IGA used the adaptive schedules given in Section IV and reliability-guided initialization. The MST heuristic first constructed a minimum-cost spanning tree and then augmented it to satisfy the minimum-degree constraint. Five independent runs were performed for each algorithm and benchmark. Final reliability was estimated with 5,000 Monte Carlo trials per optimized topology. All implementations used Python with deterministic benchmark seeds. Reported values are mean $\pm$ standard deviation across the five independent runs unless otherwise stated.

Table:1 Experimental configurations.

Parameter	Setting
Topologies	Mesh, Torus, Hypercube
Network sizes	64 and 256 nodes
Minimum degree	2
Link cost	Integer 1–5
Link reliability	[0.90,0.995)
Budget	1.12 × minimum feasible-base cost
Population	30
Generations	60
Tournament size	3
Elitism	2 chromosomes
GA crossover/mutation	0.80 / 0.02
IGA crossover	0.90 → 0.70
IGA mutation	0.08 → 0.01
Independent runs	5
Monte Carlo trials	5,000 per topology

7. Experimental Results and Discussion

Table-2 Reports the principal results. The same budget is used by all methods within each benchmark, making reliability the main differentiating outcome once feasibility is achieved.

Table:2 Benchmark results (mean ± standard deviation over five independent runs).

Network	Method	Cost	All-terminal reliability	Fitness	Runtime (s)
Mesh-64	MST	198.0 ± 0.0	0.8133 ± 0.0039	0.9982 ± 0.0000	0.235 ± 0.125
	GA	222.0 ± 0.0	0.8755 ± 0.0060	1.0222 ± 0.0001	0.376 ± 0.200
	IGA	222.0 ± 0.0	0.8728 ± 0.0044	1.0230 ± 0.0000	0.398 ± 0.206
Mesh-256	MST	770.0 ± 0.0	0.5602 ± 0.0075	1.0000 ± 0.0000	0.968 ± 0.508
	GA	863.0 ± 0.0	0.7028 ± 0.0164	1.0206 ± 0.0001	1.516 ± 0.801
	IGA	863.0 ± 0.0	0.7138 ± 0.0082	1.0208 ± 0.0000	1.648 ± 0.864
Torus-64	MST	169.0 ± 0.0	0.7957 ± 0.0041	0.9829 ± 0.0000	0.230 ± 0.118
	GA	190.0 ± 0.0	0.8624 ± 0.0107	0.9985 ± 0.0003	0.378 ± 0.200
	IGA	190.0 ± 0.0	0.8562 ± 0.0041	0.9987 ± 0.0000	0.400 ± 0.211
Torus-256	MST	719.0 ± 0.0	0.5274 ± 0.0076	0.9890 ± 0.0000	0.967 ± 0.511
	GA	806.0 ± 0.0	0.7001 ± 0.0108	1.0065 ± 0.0001	1.580 ± 0.851
	IGA	806.0 ± 0.0	0.6988 ± 0.0030	1.0066 ± 0.0000	1.736 ± 0.907
Hypercube-64	MST	129.0 ± 0.0	0.8562 ± 0.0066	0.9707 ± 0.0000	0.239 ± 0.125
	GA	145.0 ± 0.0	0.8982 ± 0.0084	0.9814 ± 0.0001	0.445 ± 0.237
	IGA	145.0 ± 0.0	0.9025 ± 0.0033	0.9813 ± 0.0000	0.463 ± 0.240
Hypercube-256	MST	504.0 ± 0.0	0.5417 ± 0.0059	0.9634 ± 0.0000	1.041 ± 0.548
	GA	565.0 ± 0.0	0.6658 ± 0.0072	0.9718 ± 0.0000	2.060 ± 1.090
	IGA	565.0 ± 0.0	0.6647 ± 0.0087	0.9718 ± 0.0000	2.464 ± 1.354

7.1. Reliability

The proposed IGA substantially improves reliability over the MST heuristic on all six benchmarks because the optimized solutions use the available budget to introduce additional high-utility links. On the 256-node Mesh, IGA reaches 0.7138 compared with 0.7028 for GA and 0.5602 for the MST heuristic. On the 64-node Hypercube, IGA also produces the highest reliability (0.9025). On the Torus cases, the conventional GA is slightly better than IGA in the measured Monte Carlo reliability, although IGA obtains marginally higher surrogate fitness. The results therefore indicate that the proposed strategy is competitive rather than universally dominant.

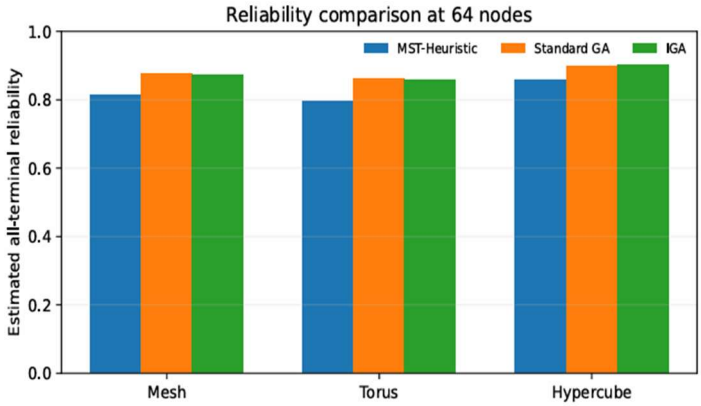

Fig.1 Estimated all-terminal reliability for 64-node benchmark networks.

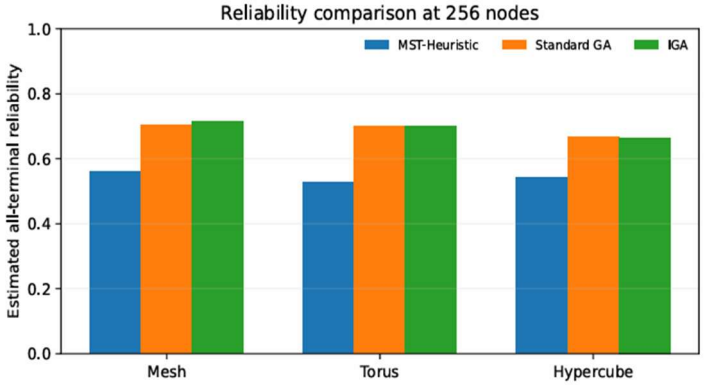

Fig.2 Estimated all-terminal reliability for 256-node benchmark networks.

7.2. Cost and Fitness

The GA and IGA solutions generally consume the complete available budget, while the MST heuristic uses less because it prioritizes low cost. This behavior explains a substantial portion of the reliability gap. The IGA fitness is equal to or slightly greater than the conventional GA on all six configurations, with the largest difference visible in the Mesh cases. Because the validation reliability is independently estimated, the results also show that a small surrogate-fitness difference does not necessarily imply a large reliability difference.

7.3. Scalability and Runtime

Runtime increases with network size for all methods. The 256-node Hypercube is the most computationally demanding case because it contains 1,024 candidate links. The IGA requires slightly more runtime than the conventional GA because of reliability-guided initialization and adaptive processing. This overhead is modest relative to the gain observed on the 256-node Mesh, but it is a genuine trade-off and is therefore reported rather than hidden.

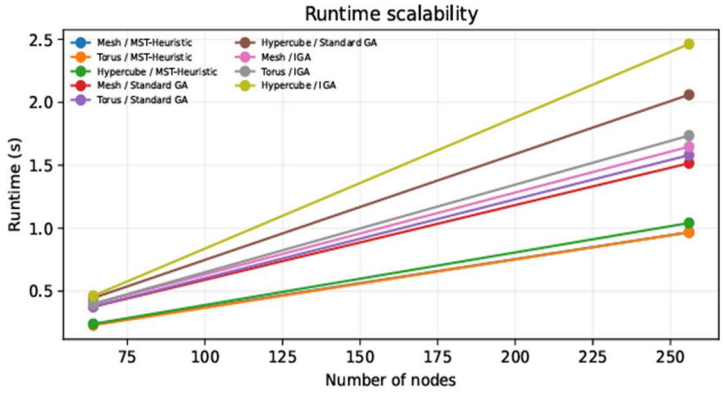

Fig.3 Runtime scalability across benchmark sizes and topologies.

7.4. Convergence

Figures 4, 5, and 6 show the mean best-so-far surrogate fitness for the 256-node benchmarks. The IGA reaches a stable high-quality solution quickly because its initial population is biased toward reliability-efficient structures. The conventional GA converges to closely related fitness levels, confirming that the proposed improvements

primarily affect search guidance and feasibility handling rather than changing the underlying optimization objective.

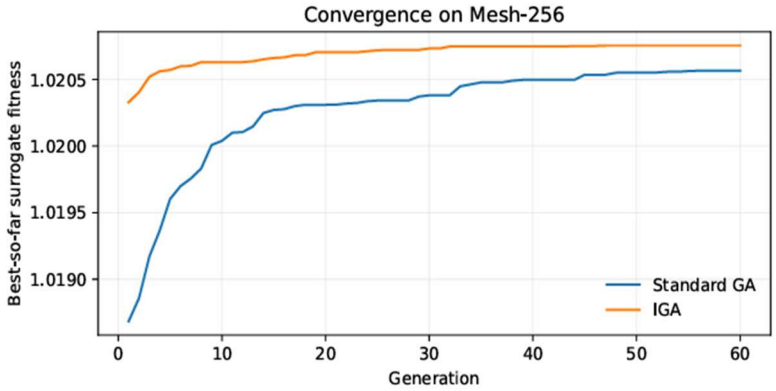

Fig.4 Convergence behavior on the 256-node Mesh.

8. Theoretical Properties

8.1. Feasible-Offspring Probability Under Repair

Let p_f^{SGA} denote the probability that an offspring generated by a standard GA is feasible before repair and p_f^{IGA} the probability after a repair operator that leaves feasible chromosomes unchanged and can map at least one infeasible pattern to a feasible topology. Then

$$p_f^{IGA} \geq p_f^{SGA}.$$

The argument follows because existing feasible offspring are retained and additional feasible offspring may be recovered.

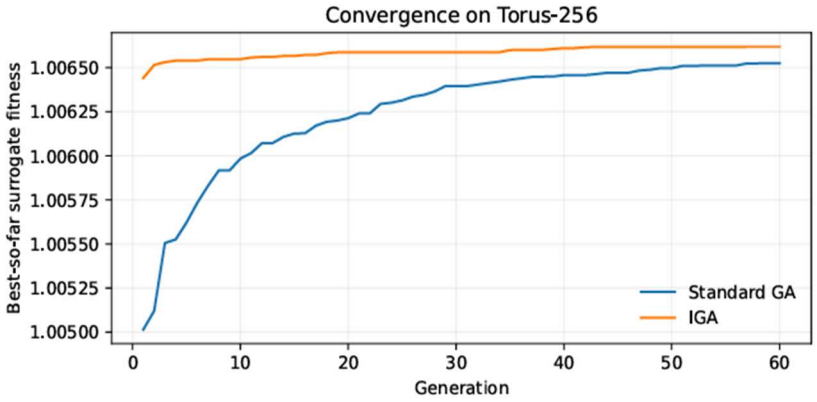

Fig. 5. Convergence behavior on the 256-node Torus

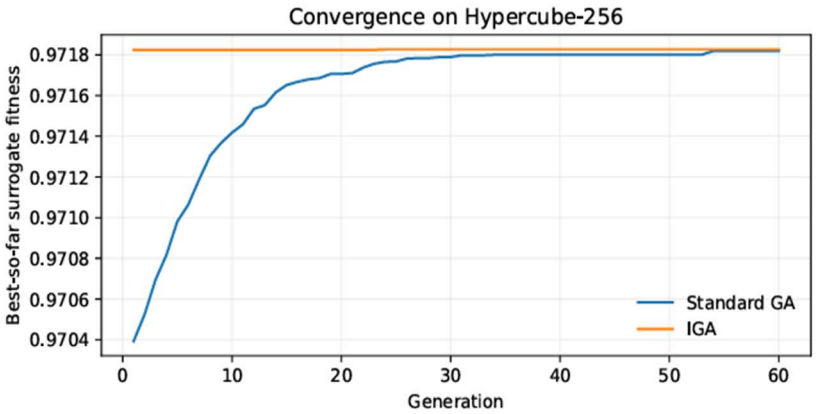

Fig.6 Convergence behavior on the 256-node Hypercube.

8.2. Best-So-Far Fitness Under Elitism

If the best feasible chromosome in generation g is copied unchanged into generation $g + 1$, then

$$f_{g+1}^{F,*} \geq f_g^{F,*}.$$

Thus, elitism guarantees a non-decreasing best-so-far feasible fitness sequence.

8.3. Connectivity Recovery

If the component graph induced by candidate links between disconnected components is connected, a spanning tree of the component graph contains $q-1$ links for q components. Adding these links restores connectivity. This establishes the recoverability condition assumed by the repair procedure.

8.4. Computational Complexity

For one chromosome, cost and degree calculations require $O(m)$ operations and connectivity checking requires $O(n+m)$ time using standard graph traversal algorithms [24]. The implementation uses a precomputed feasible base for each benchmark and therefore performs the principal offspring projection and utility-ordered addition in $O(m)$ time. The approximate evolutionary-search complexity is consequently

$$O(G_{max}P(n+m)).$$

The independent Monte Carlo validation adds

$$O(N_s(n+m))$$

per optimized topology, giving a total validation-aware cost of approximately $O(G_{max}P(n+m)+N_s(n+m))$ per final solution.

9. Discussion and Limitations

The experiments address the principal empirical gap identified by the reviewer. The proposed method is now tested on standard Mesh, Torus, and Hypercube structures at two scales and compared with two baseline methods. The results demonstrate that reliability-guided search can improve the quality of feasible topologies, especially when the budget is tight. However, the results do not establish universal superiority: the conventional GA is slightly better than IGA on the measured Torus reliability, and IGA has a modest runtime overhead.

The benchmark link costs and reliabilities are synthetic but reproducible. Therefore, the experiments validate algorithmic behavior on standard topological structures rather than claiming performance on a particular deployed physical network. Future work should incorporate application-specific link costs, measured failure rates, larger networks, and exact or more advanced reliability estimators where computationally practical. A further extension is multi-objective optimization involving latency, throughput, and energy.

10. Conclusion

This revised paper presented and experimentally evaluated an Improved Genetic Algorithm for reliability- and cost-constrained interconnection-network optimization. The formulation uses a binary graph representation, reliability-aware surrogate optimization, adaptive crossover and mutation, feasibility-preserving repair, and elitist preservation. To address the limitation of the original reliability measure, the revised methodology independently estimates all-terminal reliability using Monte Carlo link-state simulation.

Experiments on 64- and 256-node Mesh, Torus, and Hypercube networks show that IGA is competitive with conventional GA and consistently outperforms the MST-based heuristic in the reported reliability metric. The strongest observed improvement occurs on the 256-node Mesh, while the conventional GA remains slightly better on the Torus benchmarks. These results provide evidence for the usefulness of the proposed search strategy while maintaining an honest account of its computational overhead and topology-dependent behavior.

Acknowledgment

The authors thank the Department of Computer Science, Awadhesh Pratap Singh University, Rewa, M.P., India, for providing the academic environment and support necessary to conduct this research.

Conflict of Interest

The authors declare that they have no conflict of interest related to the publication of this manuscript.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Authors' Contribution

Sushil Kumar Dwivedi: Conceptualization, mathematical formulation, methodology, algorithm design, experimental design, simulation, analysis, and manuscript preparation. Rakesh Kumar Katare: Supervision, research guidance, conceptual review, critical evaluation, and manuscript review and revision. Both authors read and approved the final version.

References

[1] Boorstyn, R., Frank, H. (1977). Large-scale network topological optimization. *IEEE Trans. Commun.*, 25(1), 29–47.

[2] Jan, R. H., Hwang, F. J., Cheng, S. T. (1993). Topological optimization of a communication network subject to a reliability constraint. *IEEE Trans. Rel.*, 42(1), 63–70.

[3] Cheng, S. T. (1998). Topological optimization of a reliable communication network. *IEEE Trans. Rel.*, 47(3), 225–233,
[4] Dwivedi, S. K., Katare, R. K. (2026). A mathematical framework for structural connectivity and complexity analysis of processor interconnection networks. *AIJFR*, 7(5). doi: 10.63363/aijfr. 2026.v07i05.7874.
[5] Colbourn, C. J. (1987). *The Combinatorics of Network Reliability*. New York, NY, USA: Oxford Univ. Press.
[6] Karger, D. R. (1999). A randomized fully polynomial time approximation scheme for the all-terminal network reliability problem. *SIAM J. Comput.*, 29(2), 492–514, 1999, doi: 10.1137/S0097539796298340.
[7] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA, USA: Addison-Wesley.
[8] Eiben, A. E., Smith, J. E. (2003). *Introduction to Evolutionary Computing*. Berlin, Germany: Springer.
[9] Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.*, 186(2–4), 311–338.
[10] Dengiz, B., Altiparmak, F., Smith, A. E. (1997). Local search genetic algorithm for optimal design of reliable networks. *IEEE Trans. Evol. Comput.*, 1(3), 179–188.
[11] Tripathy, P. K., Dash, R. K., Tripathy, C. R. (2012). A genetic-algorithm-based approach for topological optimization of interconnection networks. *Procedia Technol.*, 6, 196–205. doi: 10.1016/j.protcy.2012.10.024.
[12] Ai, H., Fan, Y., Zhang, J., Ghafoor, K. Z. (2022). Topology optimization of computer communication network based on improved genetic algorithm. *J. Intell. Syst.*, 31(1), 651–659. doi: 10.1515/jisys-2022-0050.
[13] Aggarwal, K. K., Chopra, Y. C., Bajwa, J. S. (1982). Topological layout of links for optimizing the overall reliability in a computer communication system. *Microelectron. Rel.*, 22(3), 347–351.
[14] Kumar, A., Pathak, R. M., Gupta, Y. P. (1995). Genetic-algorithm-based reliability optimization for computer network expansion. *IEEE Trans. Rel.*, 44(1), 63–72.
[15] Dengiz, B., Altiparmak, F., Smith, A. E. (1997). Efficient optimization of all-terminal reliable networks using an evolutionary approach. *IEEE Trans. Rel.*, 46(1), 18–26.
[16] Pierre, S., Legault, G. (1998). A genetic algorithm for designing distributed computer network topologies. *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, 28(2), 249–258.
[17] Tripathy, P K., Dash, R., K., Tripathy, C. R. (2010). A self-generating disjoint minimal cut-set method for reliability evaluation of interconnection networks. in *Proc. IEEE Int. Conf. Signal Process. Commun.*, 1–5.
[18] Tripathy, P. K., Dash, R. K., Tripathy, C. R. (2013). A new genetic-algorithm-based method for topological optimization of interconnection networks. *Int. J. Comput. Appl.*, 63(3), 7–13. doi: 10.5120/10444-5135.
[19] Singh, M., Singh, R. N., Srivastava, A. K. et al. (2023). Structural relationship of interconnection network. *SN Comput. Sci.*, 4, 551.
[20] Pandey, S. P., Katare, R. K., Gupta, M. K., Katare, C. (2025). Study of the structural relationship between the addresses of nodes of an interconnection network. in *Proc. 4th Int. Conf. Inf. Technol., Civil Innov., Sci., Manag.*, doi: 10.4108/eai.28-4-2025.2357929.
[21] Davila-Frias, A., Yadav, O. P. (2022). All-terminal network reliability estimation using convolutional neural networks. *Proc. Inst. Mech. Eng., Part O: J. Risk Reliab.*, 236(4), 589–603.
[22] Ding, C., Wei, P., Shi, Y., Liu, J., Broggi, M., Beer, M. (2024). *Sampling and active learning methods for network reliability estimation using K-terminal spanning tree*. arXiv:2407.11053.
[23] Guo, H., Jerrum, M. (2019). A polynomial-time approximation algorithm for all-terminal network reliability," *SIAM J. Comput.*, 48(3), 964–985, 2019, doi: 10.1137/18M1201846.
[24] Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009). *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press.