INTERNATIONAL JOURNAL OF SCIENTIFIC INFORMATION

Review | ISSN: 2583-8512

Risk-Adjusted Fuzzy Dijkstra Algorithm for Shortest-Path Optimization Under Uncertain Network Conditions: A Computational Study

Pardeep Malhan*, Manisha, Rachan Khandelwal

Department of Mathematics, Jayoti Vidyapeeth Women's University, Jaipur, Rajasthan, India

Email: dhandapardeep1996@gmail.com
manishahisar15@gmail.com

Abstract

Real network parameters such as travel time, latency, cost, and capacity are rarely known with complete precision. Classical shortest-path algorithms require crisp edge weights and can therefore select routes that are nominally attractive but vulnerable to congestion or measurement uncertainty. This paper develops a risk-adjusted fuzzy Dijkstra algorithm in which every edge cost is represented by a triangular fuzzy number and converted to an additive ranking score that combines possibilistic central tendency with uncertainty spread. The additive structure preserves the computational advantages of Dijkstra's algorithm while permitting a decision maker to control risk aversion through a single parameter, λ. A reproducible computational experiment was conducted on connected random geometric networks containing 50, 100, 250, and 500 nodes. The calibration stage selected $\lambda = 0.5$, after which the proposed method was evaluated against modal-weight and centroid-weight Dijkstra baselines on 432 independent source-destination cases and 500 uncertainty scenarios per selected route. Across all cases, the proposed algorithm reduced the mean 95th-percentile travel time by 1.26% and conditional value-at-risk at 95% by 1.50% relative to modal Dijkstra. The probability of exceeding a route's nominal time by more than 20% decreased from 4.20% to 1.92%, representing a 54.36% relative reduction. Compared with centroid Dijkstra, the proposed method produced a small 0.19% increase in average travel time but reduced CVaR95 by 0.32% and exceedance probability by 29.43%. The average computation time remained below 0.5 ms per query in the tested networks. The findings show that a simple risk-adjusted fuzzy ranking can improve route robustness without sacrificing scalability, providing a mathematically transparent foundation for later applications in transportation, communication, energy, and supply-chain networks.

***Keywords:** fuzzy graph; shortest path; triangular fuzzy number; Dijkstra algorithm; network optimization; uncertainty; CVaR*

Address for Correspondence
Pardeep Malhan
PhD Research Scholar
Department of Mathematics,
Jayoti Vidyapeeth Women's University (JVWU), Jaipur, India
dhandapardeep1996@gmail.com

Article Received on: 21.04.26

Revised on: 07.05.26

Approved for publication: 18.05.26

How to Cite this Article: Malhan P, Manisha and Khandelwal R. Risk-Adjusted Fuzzy Dijkstra Algorithm for Shortest-Path Optimization Under Uncertain Network Conditions: A Computational Study. Int., J. Sci. Info. 2026; 4 (2): 35-49

1. Introduction

Network optimization supports decisions in transportation systems, communication infrastructures, power grids, logistics networks, and digital platforms. A basic requirement in many of these systems is to identify a path that minimizes an additive criterion such as distance, time, monetary cost, delay, or energy consumption. The classical shortest-path problem is mathematically mature, and Dijkstra's algorithm remains one of its most influential solutions for graphs with non-negative edge weights (Dijkstra, 1959). Its practical success, however, depends on an assumption that the weight assigned to each edge is known precisely. This assumption is difficult to defend in operational networks because traffic changes, sensor errors, fluctuating demand, link degradation, weather, human behaviour, and incomplete observations create uncertainty around apparently crisp numerical values.

Uncertainty in a network is not always adequately represented by a probability distribution. In early planning, expert judgement may only provide an optimistic value, a most plausible value, and a pessimistic value. Fuzzy set theory was introduced to represent graded membership and imprecise information rather than forcing every observation into a binary set (Zadeh, 1965). Rosenfeld (1975) extended this idea to graphs, permitting uncertainty to be associated with vertices, edges, relationships, or weights. Later work established taxonomies of fuzzy graphs and developed methods for fuzzy connectivity, fuzzy path comparison, and network decision-making (Blue et al., 2002). These developments make fuzzy graphs appropriate when network information is linguistic, interval-like, partially observed, or derived from several inconsistent sources.

The fuzzy shortest-path problem has received considerable attention. Klein (1991) examined shortest paths with fuzzy values, while Lin and Chern (1993) studied fuzzy path length and vital arcs. Okada and Soper (2000) developed a fuzzy-arc formulation, and Okada (2004) emphasized interaction among candidate paths. Other studies introduced imprecise edge-weight algorithms, generic fuzzy-parameter procedures, dynamic programming, and fuzzy modifications of Dijkstra's algorithm (Nayeem & Pal, 2005; Hernandes et al., 2007; Mahdavi et al., 2009; Deng et al., 2012). This literature demonstrates that fuzzy modelling can reveal uncertainty hidden by a single crisp estimate. Nevertheless, many proposed methods either generate several incomparable fuzzy routes, use ranking procedures that are difficult to interpret, or require additional computational machinery that complicates real-time implementation.

A practical algorithm should satisfy four requirements. First, it should retain the optimistic, modal, and pessimistic information contained in a fuzzy edge weight. Second, it should distinguish between two edges with similar central values but different uncertainty spreads. Third, it should preserve additivity so that established shortest-path search procedures can still be used. Fourth, it should provide transparent control over the trade-off between nominal efficiency and robustness. The present study addresses these requirements through a risk-adjusted ranking index for triangular fuzzy edge weights. The index combines a possibilistic mean with a spread penalty regulated by λ. Because the score is computed separately for every edge and then added along a path, the proposed model can use the standard Dijkstra relaxation mechanism without changing its order of computational complexity.

The paper contributes in three ways. It formulates a mathematically simple risk-adjusted fuzzy shortest-path model; it develops a reproducible simulation design for evaluating route behaviour under repeated uncertainty scenarios; and it reports paired statistical comparisons with two natural baselines. The first baseline uses the modal edge value and represents conventional planning with the most likely estimate. The second uses the centroid of the triangular fuzzy number and represents uncertainty-aware defuzzification without an explicit risk penalty. The proposed method is expected to provide lower upper-tail travel time and lower probability of serious delay, while retaining computation times suitable for larger network applications.

The scope is intentionally focused. Rather than claiming immediate validation in a specific city or industrial network, the study uses transportation-like random geometric graphs to test the mathematical and computational behaviour of the algorithm under controlled conditions. This makes the reported results reproducible and prevents synthetic findings from being presented as field evidence. The model is designed as the first research paper arising from the wider doctoral theme of fuzzy graph theory for network optimization; later studies can replace simulated fuzzy weights with traffic, communication, smart-grid, or supply-chain observations.

2. Literature Review and Research Gap

The theoretical basis of this study combines fuzzy-set representation, fuzzy graph modelling, and shortest-path optimization. Zadeh's (1965) membership function permits an element to belong to a set to a degree between zero and one. A fuzzy number uses this principle to describe an imprecise numerical quantity. Triangular fuzzy numbers are especially common because three interpretable points can describe an uncertain edge: a lower plausible value, a modal value, and an upper plausible value. Rosenfeld's (1975) fuzzy graph framework and the later unified treatment by Blue et al. (2002) showed that fuzziness can be attached to several components of a graph. In network optimization, edge-weight fuzziness is particularly useful because route cost, travel time, reliability, or capacity may be uncertain even when the physical topology is known.

Early fuzzy shortest-path research revealed a central difficulty: fuzzy numbers are not naturally totally ordered. Klein (1991) studied fuzzy shortest paths, and Lin and Chern (1993) linked shortest-path analysis with arc importance. Okada and Soper (2000) aggregated fuzzy arc lengths and examined route comparison, whereas Okada (2004) showed that interactions among paths can influence the preferred solution. Nayeem and Pal (2005) proposed a method that could process interval and triangular fuzzy edge weights. Hernandes et al. (2007) presented a generic algorithm for fuzzy network parameters, and Mahdavi et al. (2009) used dynamic programming to find fuzzy shortest chains. These studies expanded the mathematical treatment of the problem, but their ranking and dominance mechanisms may return several alternatives or require non-trivial fuzzy arithmetic during search.

Deng et al. (2012) modified Dijkstra's algorithm for an uncertain environment by using a fuzzy-number ranking procedure. Their work is especially relevant because it demonstrates that fuzzy shortest-path search can remain computationally efficient when an appropriate scalar comparison is available. Subsequent research has explored genetic algorithms, constrained fuzzy paths, interval-valued and intuitionistic fuzzy weights, and multi-objective formulations. Such extensions are valuable, but more elaborate fuzzy structures also increase parameter requirements and may reduce interpretability. For operational decision makers, a model that can be explained in terms of typical cost and uncertainty width may be easier to validate and implement.

A second gap concerns evaluation. Many fuzzy shortest-path papers illustrate an algorithm with one small numerical network. A worked example is useful for checking calculations, but it does not establish scalability or robustness across many network configurations. A further difficulty is that fuzzy membership and probability describe different forms of uncertainty. It would be conceptually incorrect to treat a membership grade automatically as a probability. Nevertheless, after a fuzzy route has been selected, scenario simulation can be used as an external stress test by drawing values within the stated lower, modal, and upper bounds. This does not redefine the fuzzy number as a probability distribution; it assesses how the chosen path behaves under a transparent set of plausible realizations.

The present work therefore targets a clear research gap: an additive and risk-adjustable ranking rule is integrated with Dijkstra search and evaluated on hundreds of independent path queries using upper-tail performance measures. Conditional value-at-risk is included because it summarizes the average outcome beyond a selected high quantile and is widely used as a coherent tail-risk measure (Rockafellar & Uryasev, 2000). The resulting framework links fuzzy modelling with robust performance evaluation without requiring a complex metaheuristic or a large set of incomparable routes.

3. Mathematical Model and Proposed Algorithm

Let $G = (V, E)$ be a connected graph, where V is the set of nodes and E is the set of edges. For each edge $(i, j) \in E$, the uncertain cost is represented by a triangular fuzzy number $\tilde{c}_{ij} = (L_{ij}, M_{ij}, U_{ij})$, with $0 < L_{ij} \leq M_{ij} \leq U_{ij}$. The values L, M, and U denote the lower plausible cost, most plausible cost, and upper plausible cost. The membership function increases linearly from L to M and decreases linearly from M to U. If a path P contains a sequence of edges, its fuzzy cost is obtained through componentwise addition: $\tilde{C}(P) = (\Sigma L_{ij}, \Sigma M_{ij}, \Sigma U_{ij})$. This operation retains the three-point uncertainty description at path level.

The proposed risk-adjusted score for an edge is defined as $R_\lambda(\tilde{c}_{ij}) = (L_{ij} + 2M_{ij} + U_{ij})/4 + \lambda(U_{ij} - L_{ij})/2$, where $\lambda \geq 0$ is the risk-aversion parameter. The first term is a weighted possibilistic central value that gives additional importance to the modal estimate. The second term penalizes half of the support width. When $\lambda = 0$, selection is driven by central tendency; larger λ values increasingly discourage uncertain edges. The path score is $S_\lambda(P) = \Sigma R_\lambda(\tilde{c}_{ij})$. Because R_λ is additive, minimizing $S_\lambda(P)$ is equivalent to running Dijkstra's algorithm on the scalar edge scores R_λ. Thus, with a binary heap implementation, the asymptotic query complexity remains $O((|V| + |E|) \log |V|)$, apart from the constant-time score calculation.

The formulation has an important interpretive property. Two edges can have the same modal value but different upper bounds. Modal Dijkstra treats them as equivalent, while centroid Dijkstra changes the score only through the average location of the triangle. The proposed index explicitly recognizes the width of uncertainty. Consequently, an edge with a slightly larger typical cost can be preferred when its range is substantially narrower. This is suitable for applications where a small nominal saving is not worth a high probability of severe delay or overload.

The algorithm returns both a crisp route and its aggregated fuzzy cost. The route is needed for operational implementation, while the aggregated triple communicates the optimistic, most plausible, and pessimistic path estimates. The decision maker can also repeat the calculation for several λ values to obtain a sensitivity profile. In this study, λ is selected using a separate calibration sample rather than chosen after viewing the evaluation results.

Table 1. Principal notation used in the mathematical model

Symbol	Meaning
$G = (V, E)$	Connected network with node set V and edge set E

$\tilde{c}ij = (Lij, Mij, Uij)$	Triangular fuzzy cost of edge (i, j)
$\tilde{C}(P)$	Componentwise sum of fuzzy costs along path P
$R\lambda(\tilde{c}ij)$	Risk-adjusted scalar score assigned to an edge
λ	Non-negative risk-aversion parameter
P95	95th percentile of simulated realized path cost
CVaR95	Mean cost among outcomes at or above the 95th percentile

Table 2. Risk-adjusted fuzzy Dijkstra procedure

Step	Operation
1	Read the graph and assign each edge a triangular fuzzy weight (L, M, U).
2	Calculate $R\lambda = (L + 2M + U)/4 + \lambda(U - L)/2$ for every edge.
3	Initialize the source distance to zero and all other tentative distances to infinity.
4	Use the standard Dijkstra priority queue; relax each edge with $R\lambda$ as its non-negative weight.
5	Backtrack predecessors to obtain the minimum-score route from source to target.
6	Aggregate the original fuzzy triples along the route and report both the route and $\tilde{C}(P)$.

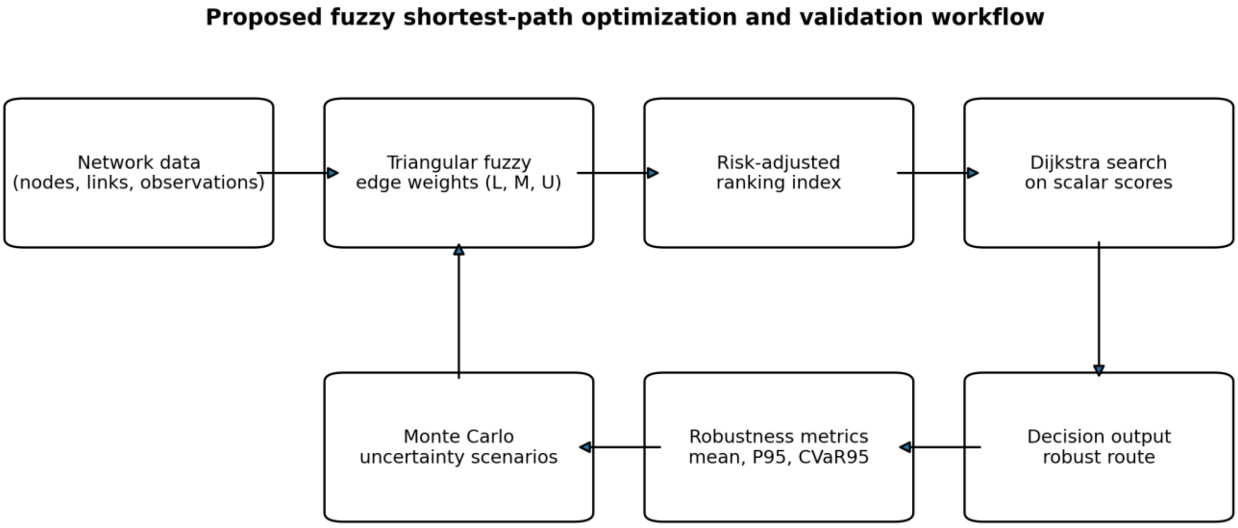

Figure 1. Proposed fuzzy shortest-path optimization and validation workflow

4. Research Methodology

A computational experimental design was adopted to assess the proposed algorithm under controlled but diverse uncertainty conditions. Connected random geometric graphs were selected because their nodes possess spatial coordinates and edges are more likely between nearby nodes, making them more transportation-like than purely random graphs. Nodes were positioned uniformly in a unit square. A connection radius based on network size was increased when necessary until the graph was connected. Four sizes were examined: 50, 100, 250, and 500 nodes. For the final evaluation, nine independently generated graphs were used at each size, and twelve source-destination pairs requiring at least three topological steps were sampled per graph. This produced 432 independent routing cases.

Each edge received a fuzzy travel-time triple generated from its Euclidean length and a simulated congestion factor. The modal value increased with physical distance and congestion. The lower value represented favourable operating conditions, whereas the upper value included an asymmetric volatility allowance so that disruption could increase travel time more strongly than favourable conditions could reduce it. All generated values were positive. This design creates heterogeneous edges: some are fast but volatile, while others are slightly slower but more stable. Such heterogeneity is necessary to test whether a risk-aware algorithm actually changes route selection.

Three methods were compared. Modal Dijkstra used Mij as the edge weight and represented the conventional use of a single most plausible estimate. Centroid Dijkstra used (Lij + Mij + Uij)/3 and represented basic defuzzification. The proposed risk-adjusted fuzzy Dijkstra used $R\lambda$. To prevent overfitting, λ was calibrated on independent networks: three graphs per size, eight source-destination pairs per graph, and 300 uncertainty scenarios for each candidate λ in {0.0, 0.1, …, 0.8}. The selected value minimized average CVaR95 while limiting the increase in mean travel time to no more than 3% relative to $\lambda = 0$. The resulting $\lambda = 0.5$ was fixed before the main evaluation.

For each selected route in the main evaluation, 500 realized edge-time scenarios were generated using triangular sampling within the corresponding lower, modal, and upper values. The sampling distribution is an evaluation device, not a claim that fuzzy membership is probabilistic. It provides a consistent stress test in which the most plausible value is sampled more frequently while all realizations remain within the specified support. Route outcomes were summarized by mean travel time, standard deviation, P95, CVaR95, and the probability that realized time exceeded the route's own modal total by more than 20%. Query runtime and the number of edges in the route were also recorded. In total, 1,296 method-specific route evaluations and 648,000 realized route scenarios were analysed.

Results were aggregated by network size and across all cases. Because each source-destination case was solved by every method, paired non-parametric Wilcoxon signed-rank tests were used for the overall comparisons. This test avoids assuming normally distributed paired differences. Statistical significance was interpreted together with percentage change because a very small effect can become statistically significant in a large paired sample. All simulations used a fixed master seed to support reproducibility. The model was implemented in Python using NetworkX for graph construction and shortest-path search, NumPy for scenario generation, and SciPy for paired tests.

Table 3. Experimental design and evaluation settings

Component	Specification
Network topology	Connected random geometric graphs in a two-dimensional unit square
Evaluation sizes	50, 100, 250, and 500 nodes
Evaluation cases	9 graphs × 12 source-destination pairs × 4 sizes = 432 cases
Compared methods	Modal Dijkstra, centroid Dijkstra, and risk-adjusted fuzzy Dijkstra
Fuzzy edge value	Triangular travel time (lower, modal, upper)
Calibration	λ = 0.0 to 0.8 in increments of 0.1 on independent graphs
Selected parameter	λ = 0.5
Stress test	500 triangular scenarios per route
Primary metrics	Mean, P95, CVaR95, >20% exceedance probability, and runtime
Statistical test	Paired Wilcoxon signed-rank test

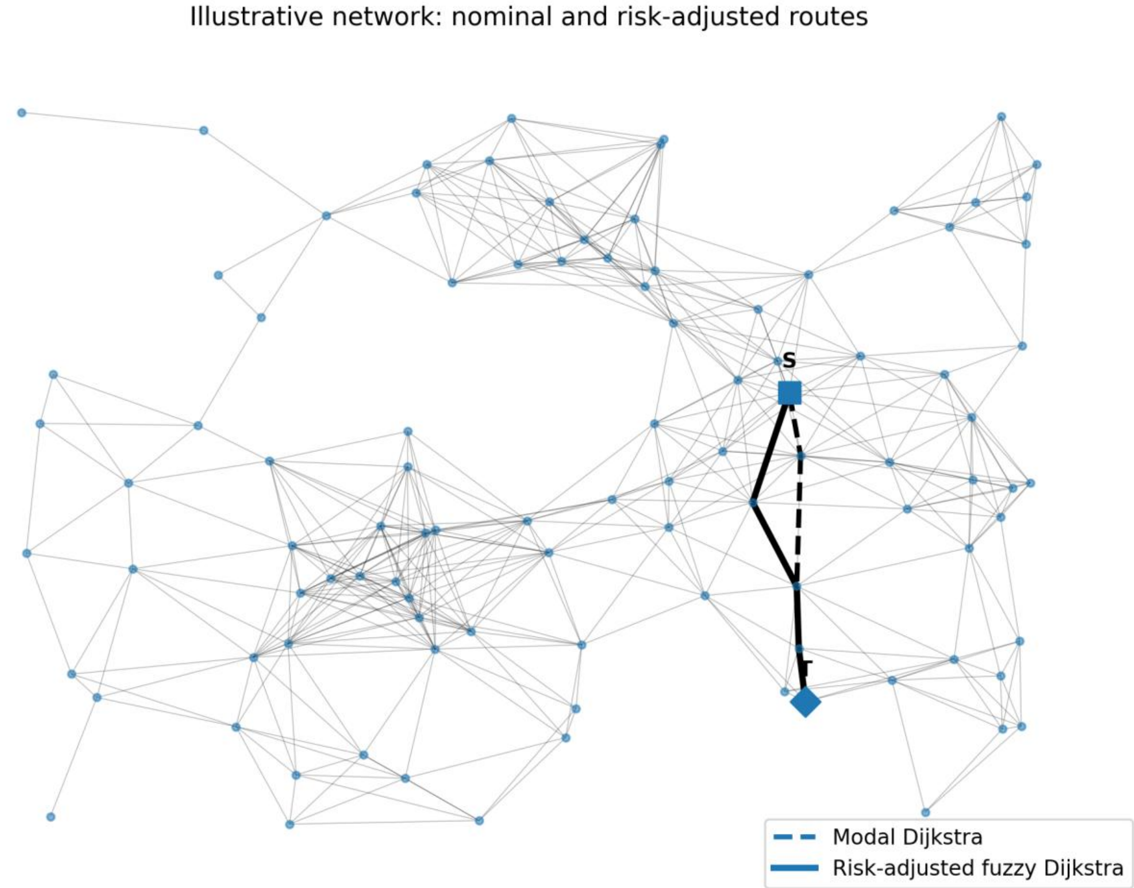

Figure 2. Network showing different nominal and risk-adjusted routes

5. Results

The calibration experiment showed a gradual reduction in upper-tail risk as λ increased from zero. At $\lambda = 0.5$, average CVaR95 was 0.547% lower than at $\lambda = 0$, while mean travel time was only 0.297% higher. Increasing λ beyond 0.5 did not produce a consistent additional CVaR benefit and continued to increase mean cost. The pre-specified selection rule therefore chose $\lambda = 0.5$. Figure 4 confirms that the calibration curve reaches its most favourable tail-risk point near this value rather than at the most risk-averse endpoint.

Table 4 presents the main evaluation results by network size. The proposed method obtained the lowest average CVaR95 at every size. Relative to modal Dijkstra, the CVaR95 reduction was approximately 1.05% for 50-node networks, 1.94% for 100-node networks, 1.33% for 250-node networks, and 1.74% for 500-node networks. Centroid Dijkstra also improved tail performance, but its reduction was smaller in every size group. Figure 3 displays the CVaR95 values across network sizes and shows that the proposed approach consistently achieved lower tail risk than both baselines.

The exceedance results were more pronounced than the changes in mean travel time. In 100-node networks, the probability that realized time exceeded the route's modal total by more than 20% fell from 4.81% under modal Dijkstra to 1.72% under the proposed method. In 500-node networks it fell from 2.49% to 0.64%. This pattern indicates that the main benefit of risk-adjusted routing is not a dramatic reduction in ordinary travel time; it is the avoidance of paths containing edges with large upper uncertainty. The proposed routes were slightly longer in edge count, averaging 6.40 edges compared with 6.29 for modal Dijkstra, but their realized tail behaviour was better.

Across all 432 paired cases, the proposed algorithm achieved a mean realized time of 86.923 units, compared with 87.210 for modal Dijkstra and 86.761 for centroid Dijkstra. Thus, it improved mean time by 0.329% relative to the modal baseline and incurred only a 0.186% increase relative to the centroid baseline. The average P95 was 92.449, which was 1.259% lower than modal Dijkstra and 0.197% lower than centroid Dijkstra. Average CVaR95 was 93.826, representing reductions of 1.504% and 0.323%, respectively. The exceedance probability was 1.918%, compared with 4.201% for modal Dijkstra and 2.717% for centroid Dijkstra. These correspond to relative reductions of 54.36% and 29.43%.

Paired Wilcoxon tests supported the consistency of the differences. The proposed method differed significantly from modal Dijkstra for mean time, P95, CVaR95, and exceedance probability (all $p < 0.001$). It also produced significantly lower P95, CVaR95, and exceedance probability than centroid Dijkstra (all $p < 0.001$). The mean-time comparison with centroid Dijkstra was statistically significant but practically small: the proposed method was only 0.162 time units higher on average. This result illustrates why effect magnitude should accompany significance testing.

The robustness gains required modest additional computation. Average query time was 0.462 ms for the proposed algorithm, 0.334 ms for modal Dijkstra, and 0.321 ms for centroid Dijkstra. The relative increase over modal Dijkstra was about 38%, but the absolute difference was approximately 0.128 ms per query, and the average time remained below 1.1 ms even for the 500-node networks. The additional cost arises from calculating the risk-adjusted edge score through a Python callback. In an optimized implementation, scores can be precomputed whenever λ is fixed, making the search operation nearly identical to standard Dijkstra.

Table 4. Average performance by network size

Nodes	Method	Mean	P95	CVaR95	Exceedance
50	Centroid	97.266	105.176	107.201	4.13%
50	Modal	97.619	105.941	108.023	5.46%
50	Proposed	97.361	105.009	106.888	3.42%
100	Centroid	91.222	97.598	99.223	2.60%
100	Modal	91.840	99.016	100.824	4.81%
100	Proposed	91.392	97.365	98.868	1.72%
250	Centroid	79.852	84.840	86.094	2.88%
250	Modal	80.181	85.593	86.998	4.05%
250	Proposed	80.075	84.690	85.844	1.89%
500	Centroid	78.706	82.912	84.003	1.27%
500	Modal	79.201	83.961	85.188	2.49%
500	Proposed	78.864	82.734	83.705	0.64%

Table 5. Overall paired performance and change of the proposed method

Comparison	Metric	Proposed	Baseline	Change	p-value
vs Modal	Mean time	86.923	87.210	−0.329%	<0.001
vs Modal	P95	92.449	93.628	−1.259%	<0.001
vs Modal	CVaR95	93.826	95.258	−1.504%	<0.001
vs Modal	Exceedance	1.918%	4.201%	−54.358%	<0.001
vs Centroid	Mean time	86.923	86.761	+0.186%	<0.001
vs Centroid	P95	92.449	92.631	−0.197%	<0.001
vs Centroid	CVaR95	93.826	94.130	−0.323%	<0.001
vs Centroid	Exceedance	1.918%	2.717%	−29.426%	<0.001

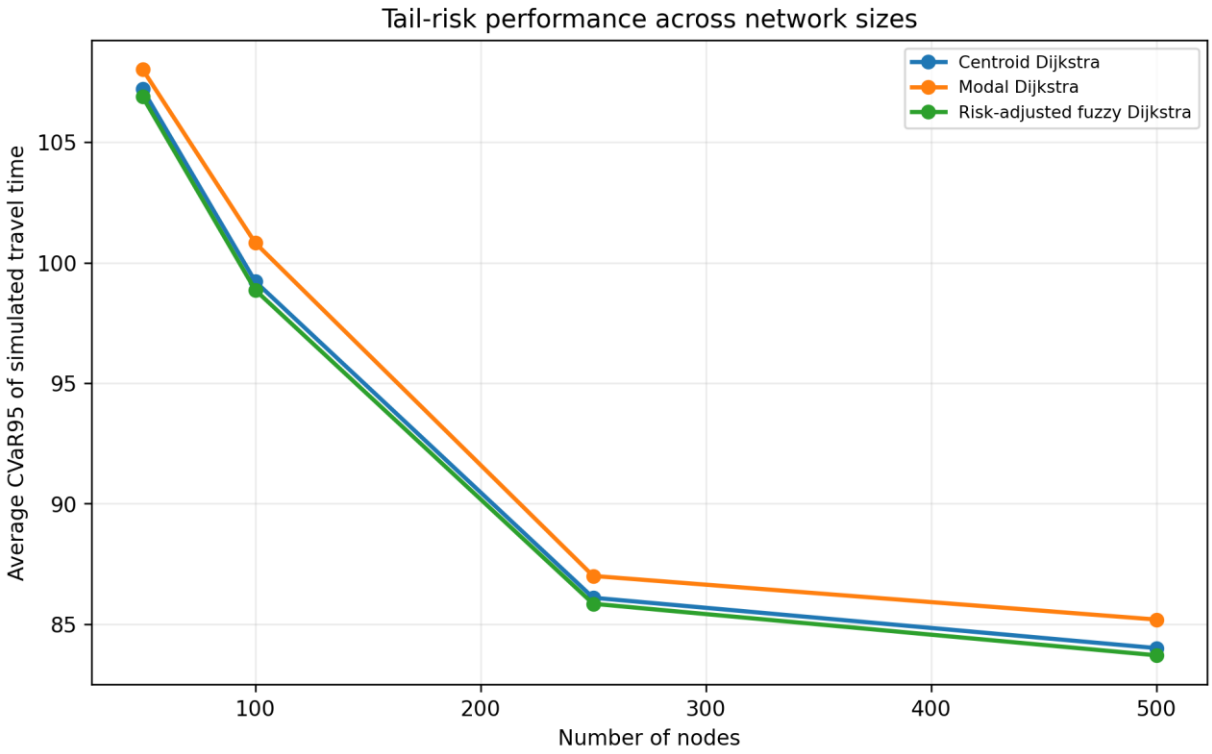

Figure 3. CVaR95 performance across network sizes

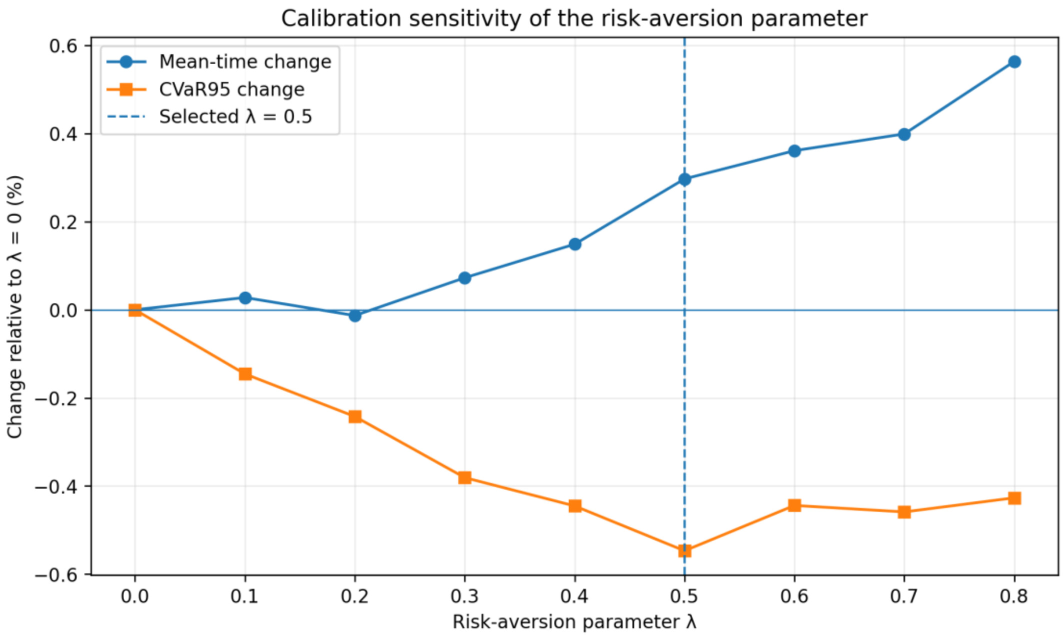

Figure 4. Sensitivity of mean time and CVaR95 to the risk-aversion parameter

6. Discussion

The results support the central proposition that uncertain network optimization benefits from separating central tendency from uncertainty spread. Modal Dijkstra uses only the most plausible edge value, so it may select a path whose individual edges appear efficient but have large pessimistic values. Centroid Dijkstra partially recognizes asymmetry in the triangular fuzzy number, which explains why it performed better than the modal baseline. The proposed method adds an explicit spread penalty and therefore has a stronger mechanism for avoiding volatile edges. Its advantage was clearest in exceedance probability and CVaR95, both of which emphasize unfavourable realizations rather than routine outcomes.

The magnitude of improvement is realistic rather than spectacular. A 1.5% reduction in CVaR95 may appear small, but network-routing improvements are often accumulated over many journeys, packets, energy transfers, or logistics movements. More importantly, the exceedance probability was reduced by more than half relative to modal Dijkstra. This difference is operationally meaningful when late arrival, congestion, overload, or service failure imposes a nonlinear penalty. A route that is marginally longer under ordinary conditions may be preferable if it substantially reduces severe delay.

The model also offers interpretability. The parameter λ has a direct meaning: it specifies how strongly the decision maker penalizes uncertainty width. Unlike a black-box metaheuristic, the ranking score can be inspected for every edge, and the selected path can be reproduced with a standard shortest-path implementation. The sensitivity analysis allows different decision contexts to choose different λ values. Emergency routing, critical communication, and power restoration may justify a larger λ, whereas low-stakes or highly cost-sensitive applications may prefer a smaller value. The calibration procedure used here is one practical option, but λ can also be determined from stakeholder preferences or service-level requirements.

A further contribution is methodological. Fuzzy graph studies sometimes stop after presenting a small numerical illustration. The current design combines fuzzy route selection with repeated scenario stress testing across several network sizes. This makes it possible to evaluate distributional behaviour, not only the ranked fuzzy score. The use of CVaR95 complements P95 because it measures the average severity of outcomes beyond the quantile threshold. The paired design also isolates algorithmic differences: each method faces exactly the same network and source-destination pair.

The study has limitations. First, the networks and fuzzy weights are simulated. Their structure is transportation-like, but the numerical results should not be interpreted as measured performance in a specific city, communication system, or smart grid. Second, scenario sampling assumes a triangular shape inside the fuzzy support. This was chosen for transparency and consistency, yet other stress-test mechanisms may produce different numerical magnitudes. Third, edge realizations were sampled independently. Real networks may exhibit correlated congestion, common weather effects, cascading failures, or time dependence. Fourth, only a single-objective additive path model was considered. Practical routing may require simultaneous constraints on capacity, reliability, emissions, tolls, or service quality.

These limitations define the next research stages. The algorithm can be applied to real traffic data by constructing fuzzy weights from historical quantiles, expert assessments, or streaming sensor intervals. Correlated scenarios can be generated using time blocks or copula models. The additive ranking can be extended to multi-objective paths, fuzzy maximum flow, minimum spanning trees, and resource-allocation problems. Hybrid models may use machine learning to update L, M, and U in real time while retaining the fuzzy graph as an interpretable decision layer. Such extensions would directly address the broader doctoral objectives of scalable algorithms, dynamic adaptation, and real-world validation.

7. Conclusion

This paper presented a risk-adjusted fuzzy Dijkstra algorithm for shortest-path optimization when edge costs are uncertain. Triangular fuzzy numbers retained lower, modal, and upper information, while an additive ranking index combined possibilistic central tendency with an uncertainty-spread penalty. The method preserved Dijkstra-level search complexity and required only one interpretable parameter. In a reproducible experiment involving 432 routing cases and 648,000 scenario outcomes, the proposed method consistently reduced upper-tail travel time and serious exceedance risk. Relative to modal Dijkstra, CVaR95 decreased by 1.50% and the probability of exceeding nominal route time by more than 20% decreased by 54.36%. Relative to centroid Dijkstra, the method achieved additional tail-risk reduction with only a 0.19% increase in mean time. Average query time remained below 0.5 ms. The findings indicate that risk-adjusted fuzzy ranking is a practical bridge between classical graph algorithms and uncertainty-aware network decisions. The study provides a defensible first computational paper for the doctoral programme and a foundation for future validation using real transportation, communication, energy, or supply-chain data.

Declarations

Conflict of interest: The author declares no conflict of interest.

Funding: No external funding was reported for this computational study.

Authors Contributions: Pardeep Malhan, the main author, conducted the research, developed the methodology, analysed the results, and prepared the original manuscript. Dr. Rachna Khandelwal supervised the study, provided conceptual guidance, and critically revised the paper. Manisha, as co-author, contributed to the literature review, data interpretation, and final editing of the manuscript.

References

Blue, M., Bush, B., & Puckett, J. (2002). Unified approach to fuzzy graph problems. Fuzzy Sets and Systems, 125(3), 355–368. https://doi.org/10.1016/S0165-0114(01)00011-2

Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. Applied Soft Computing, 12(3), 1231–1237. https://doi.org/10.1016/j.asoc.2011.11.011

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1, 269–271. https://doi.org/10.1007/BF01386390

Ebrahimnejad, A. (2021). An acceptability index based approach for solving shortest path problem on a network with interval weights. RAIRO - Operations Research, 55, S1767–S1787. https://doi.org/10.1051/ro/2020033

Hernandes, F., Lamata, M. T., Verdegay, J. L., & Yamakami, A. (2007). The shortest path problem on networks with fuzzy parameters. Fuzzy Sets and Systems, 158(14), 1561–1570. https://doi.org/10.1016/j.fss.2007.02.022

Klein, C. M. (1991). Fuzzy shortest paths. Fuzzy Sets and Systems, 39(1), 27–41. https://doi.org/10.1016/0165-0114(91)90063-V

Lin, K.-C., & Chern, M.-S. (1993). The fuzzy shortest path problem and its most vital arcs. Fuzzy Sets and Systems, 58(3), 343–353. https://doi.org/10.1016/0165-0114(93)90508-F

Mahdavi, I., Nourifar, R., Heidarzade, A., & Mahdavi-Amiri, N. (2009). A dynamic programming approach for finding shortest chains in a fuzzy network. Applied Soft Computing, 9(2), 503–511. https://doi.org/10.1016/j.asoc.2008.07.002

Nayeem, S. M. A., & Pal, M. (2005). Shortest path problem on a network with imprecise edge weight. Fuzzy Optimization and Decision Making, 4, 293–312. https://doi.org/10.1007/s10700-005-3665-2

Okada, S. (2004). Fuzzy shortest path problems incorporating interactivity among paths. Fuzzy Sets and Systems, 142(3), 335–357. https://doi.org/10.1016/S0165-0114(03)00225-2

Okada, S., & Soper, T. (2000). A shortest path problem on a network with fuzzy arc lengths. Fuzzy Sets and Systems, 109(1), 129–140. https://doi.org/10.1016/S0165-0114(98)00054-2

Penrose, M. (2003). Random Geometric Graphs. Oxford University Press.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. The Journal of Risk, 2(3), 21–41. https://doi.org/10.21314/JOR.2000.038

Rosenfeld, A. (1975). Fuzzy graphs. In L. A. Zadeh, K. S. Fu, K. Tanaka, & M. Shimura (Eds.), Fuzzy Sets and Their Applications to Cognitive and Decision Processes (pp. 77–95). Academic Press. https://doi.org/10.1016/B978-0-12-775260-0.50008-6

Zadeh, L. A. (1965). Fuzzy sets. Information and Control, 8(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X