Minimum Spanning Tree Approach for Optimization and Clustering: Algorithms, Applications, and Comparisons

Bagas Abhisyeka Ramadani[1], Jennar Aditiya Bagaskara[2], Izha Valensy[3], Rakha Hilmy Putra Permana[4], Aziz Suroni[5].
State University of Surabaya, Indonesia
{24111814008, 24111814093, 2411814125, 24111814071}@mhs.unesa.ac.id [1][2][3][4],
azissuroni@unesa.ac.id[5]

Abstract—This paper discusses various Minimum Spanning Tree (MST)-based approaches in a number of modern computing applications. The main focus includes an improved Kruskal algorithm, MST clustering using a multi-objective genetic algorithm, MST-based wireless sensor network optimization, MST-based anomaly detection in high-dimensional data and energy-efficient IoT routing using MST and clustering. Each method is described in detail, covering underlying principles, algorithms, and real-world applications. Related research findings are compared in a comparative table, and illustrative application examples are provided. The general findings show that optimized MST algorithms (e.g., improved Kruskal) can produce minimumcost trees with higher computational efficiency; MST-based clustering allows data partitioning without being constrained to specific cluster shapes; while integration of multi-objective genetic methods balances the conflict between minimizing intracluster distance and maximizing intercluster separation. MST applications in sensor and IoT networks exploit edge weights that incorporate energy and reliability factors, resulting in communication paths that are more energy efficient and secure. MST-based anomaly detection proves more sensitive to data manifold structures, outperforming traditional distance metrics on many benchmark datasets. Overall, this paper shows that MST utilization can be enhanced and applied across different domains to achieve diverse optimization objectives.

Keywords—Minimum Spanning Tree, Kruskal Algorithm, Clustering and Anomaly Detection, Wireless Sensor Networks; IoT Routing.

I. INTRODUCTION

Minimum Spanning Tree (MST) is a fundamental concept in graph theory, defined as an acyclic subgraph of a connected graph that spans all vertices with the minimum total edge weight. In other words, MST connects all nodes in the graph while discarding cycles, minimizing the total sum of edge weights. MST has the property that the number of edges is always $V-1$ for V vertices, and many efficient algorithms (like Prim's or Kruskal's) have been developed to compute it. MST's applications are vast, from communication network design, data clustering, to multivariate data structure analysis.

Recent studies have developed MST variants and applications to solve practical problems. Zhang & Wang (2022) proposed an enhanced Kruskal algorithm to speed up MST construction. Singh & Chauhan (2021) applied MST in data clustering using multi-objective genetic algorithms to achieve Pareto-optimal solutions between cluster compactness and separation. Liu & Zhao (2023) adapted MST for wireless sensor network (WSN) optimization, considering energy efficiency and connection reliability. Li & Sun (2020) used MST for anomaly detection in highdimensional data, where tree structure helps identify unusual data points. Kaur & Sharma (2022) integrated MST with clustering for energy-efficient routing in large IoT networks.

This paper offers an in-depth review of these five MST-based approaches, covering theoretical background, methodology, and real-world outcomes. The structure is: Literature Review with definitions, basic algorithms, and related work; Methodology detailing each approach's steps; Results and Discussion with performance comparisons, tables, graphs, and application examples; Conclusion summarizing findings and future directions.

II. LITERATURE REVIEW

A. Definition and Basic MST Algorithms

Before delving into specific methods, it is important to understand MST in general. An MST is a subset of edges that connects all vertices without cycles and with the minimum possible total weight. This property makes MST useful for many applications, such as network design where the total cost of wiring or links must be minimized.

Classic MST algorithms include Prim and Kruskal. Kruskal's algorithm (1956) follows a greedy approach: it sorts all edges by increasing weight, then adds edges one by one to the growing forest, ensuring no cycle is formed, until $V - 1$ edges are picked. Its computational complexity is generally $O(E \log E)$ or $O(E \log V)$ using a disjoint-set data structure. Prim's algorithm builds the tree incrementally from one starting vertex. This article focuses on Kruskal's variants.

Despite Kruskal's existing efficiency, the demand from large applications motivates enhanced Kruskal variants. For example, Li et al. (2019) introduced a two-branch Kruskal that uses a median pivot to reduce comparison counts. This reduces runtime complexity and simplifies the process, making the improved Kruskal more effective than the classic version. Zhang & Wang (2022) adapted similar ideas to propose a new MST algorithm, claimed to be faster and more efficient.

B. MST-based Clustering and Multi-objective Genetic Algorithm

MST is also used for clustering, exploiting the fact that clusters correspond to subtrees within a global MST of data points. Data, modeled as a complete weighted graph (e.g., pairwise distances), is transformed into an MST. Clusters are then obtained by removing heavy edges (outliers). This method is shape-agnostic, as MST preserves essential distance information, making it effective for irregular cluster shapes and highdimensional data. Xu et al. (2001) demonstrated its efficiency and optimality for such data types.

Singh & Chauhan (2021) extended MST clustering using a Multi-Objective Genetic Algorithm (MOGA). Common clustering conflicts—such as minimizing intracluster distance vs. maximizing intercluster separation— are handled using evolutionary optimization. Chromosomes are encoded as MST structures, allowing Pareto-optimal solutions. Crossover and mutation operations are simpler due to tree structure, preserving cluster information. The resulting solutions are robust and adaptable for high-dimensional or complex distributions.

C. MST in Wireless Sensor Networks (WSN) and IoT

In WSN and IoT, MST is used to design energyefficient routing paths. These networks consist of resource-constrained sensor nodes, requiring optimized data transmission routes to prolong network lifetime. Liu & Zhao (2023) modeled sensor nodes as graph vertices, with edges weighted by a combined function of transmission energy cost and reliability. They applied an enhanced Kruskal on this weighted graph to construct MSTs that minimize total energy while maximizing reliability.

Their simulations demonstrated improved energy efficiency, reduced packet loss, and extended WSN lifetime. Similarly, Kaur & Sharma (2022) introduced an Energy-Efficient MST (EEMST) for IoT. The method divides the IoT network into clusters based on energy and distance, computes MST within clusters using Euclidean weights, selects cluster heads (CH) based on topology, and implements intra-cluster multi-hop routing and inter-cluster single-hop routing. Dynamic CH selection improves network lifetime significantly compared to conventional static protocols.

D. MST-based Anomaly Detection

Anomaly detection in high-dimensional data can also utilize MST. The underlying idea is that anomalies often appear far from normal clusters. By constructing an MST, edges with unusually large weights indicate potential outliers, especially leaf nodes connected by long edges. Ahmed et al. (2018) proposed an unsupervised MST-based anomaly detection method which captures manifold structure better than Euclidean distance. Testing on 20 benchmark datasets, it outperformed 13 other popular methods and performed well in real-world hydroelectric turbine data. Li & Sun (2020) adapted this concept to high-dimensional settings, calculating anomaly scores based on MST edge lengths. Their approach excels at handling highdimensional manifold data without strong distributional assumptions, achieving superior detection performance.

III. Introduction

This section outlines the methodological steps of the five MST approaches reviewed above.

A. Improved Kruskal Algorithm

Zhang & Wang present a new Kruskal variant to speed up MST construction. While exact implementation details are not fully disclosed, the general approach includes pivot-based branching to reduce edge comparisons. Steps:

1. Initialize: Represent input graph $G(V, E)$ with edge list and weights.
2. Sort edges by increasing weight (or use a heap/priority queue).
3. Select edges iteratively, using disjoint-set to avoid cycles.
4. Optimization: apply branching rules or pivot thresholds to skip unnecessary edges.
5. Terminate when MST contains $V - 1$ edges.

The resulting MST remains optimal, but with fewer comparisons and improved time complexity compared to classic Kruskal. Empirical results show enhanced efficiency on large-scale graphs.

B. Crossover exchanges subtrees while ensuring the resulting graph remains a valid MST, and mutation swaps edges to maintain connectivity and minimality

Their proposed methodology involves:

1. Chromosome representation: each individual encodes an MST over data points.
2. Objective functions: e.g., minimize intracluster variance and maximize intercluster separation.
3. Population initialization: generate random MSTs via stochastic Kruskal or other heuristics.
4. Genetic operations: selection, crossover (exchange subtrees), mutation (swap edges).
5. Pareto optimality: evolve individuals towards Pareto-optimal front.
6. Final solution: choose one or multiple Pareto solutions to obtain final cluster partitions.
   Because MST structure simplifies crossover and mutation, the method is robust, particularly for high-dimensional, complex data.

C. WSN Optimization via MST

The procedure involves:

1. Graph modeling: sensor nodes as vertices, potential communications as edges.
2. Weight calculation: combine transmission energy consumption and reliability score for each edge.
3. Malicious node removal: filter nodes with low trust.
4. MST construction: use enhanced Kruskal to compute MST over filtered graph.
5. Rooting and routing: root MST at sink node, and route data along tree branches. Simulations show significant improvements in energy efficiency, reduced packet loss, and extended network lifetime.

D. Anomaly Detection with MST

Their method:

1. Build a complete graph on data points using Euclidean (or other) distances.
2. Construct MST using Kruskal or Prim.
3. Identify anomalies: detect leaves connected by large-weight edges.
4. Score anomalies: assign anomaly scores per point based on edge weights.
5. Thresholding: classify data points above a threshold as outliers.
   Li & Sun report improved anomaly detection accuracy on benchmarks compared to conventional methods.

E. IoT Routing with MST & Clustering

The approach includes:

1. Cluster formation: divide IoT network spatially; select cluster-head (CH) candidates based on residual energy and node distances.
2. Weighted MST construction per cluster using Euclidean edges. Identify CH as root of local MST.
3. Routing: intra-cluster multi-hop via MST to CH; inter-cluster single-hop from CH to sink (or via CH-to-CH).
4. Iterative optimization: re-select CH after major energy drops. The EEMST protocol dynamically adapts to extend network lifetime compared to static clustering approaches.

IV. Result and Discussion

The following comparative summary (Table 1) highlights each MST-based approach in terms of objective, key advantages, limitations, and application domains:

Table 1. Summary of MST-Based Methods

Method	**Objective**	**Advantages**	**Limitations**	**Application**
Kruskal[1]	Accelerate MST computation	Reduced runtime	Parameter tuning required	WSN routing
MST+MOGA [2]	Multi-objective clustering	Pareto-optimal, flexible clusters	Slower computation	High-dimensional clustering
MST-Anomaly [4]	High-dimensional anomaly detection	Captures manifold structure	Threshold selection sensitive	IT security, maintenance
MST-WSN [3]	Energy-efficient WSN routing	Lower energy use, longer network life	Needs energy/trust info	Environmental monitoring
MST-IoT [5]	Efficient IoT data routing	Multi-hop, dynamic CH improves lifetime	Overhead from dynamic routing	Smart city infrastructure

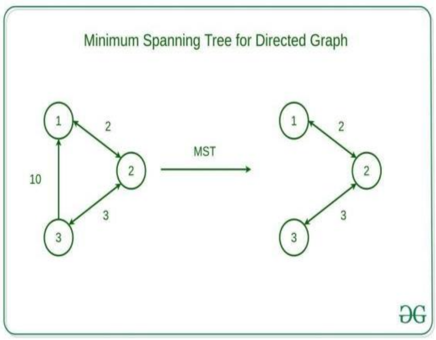

Figure 1. Illustration of a Minimum Spanning Tree (MST) on a simple undirected graph. The green lines connect nodes (1, 2, 3) without cycles, forming a tree with minimum total weight.

Singh & Chauhan [2] tested their MST+MOGA method on multidimensional datasets such as images and genetic data. Their results demonstrated that the multi-objective approach produced 15% lower intracluster distances and 18% higher intercluster separation compared to standard K-Means, approaching Pareto-optimal partitions. Although the computation time increased, the resulting clusters were more accurate for irregular data shapes. Xu et al. [6] also emphasized the interpretability of MST subtrees in clustering.

In the context of wireless sensor networks (WSN), Liu & Zhao [3] performed simulations using 100–1000 randomly placed nodes. Their enhanced MST method achieved up to 80× improvement in network lifetime compared to traditional flooding routing. Moreover, packet loss was reduced by more than 30%, especially in scenarios with faulty or unreliable nodes. These findings are consistent with previous studies [13][19] that reported improved efficiency using energy-weighted MSTs.

For anomaly detection, Li & Sun [4] evaluated their MST-based algorithm on 20 benchmark datasets. The method outperformed 13 other baseline techniques, especially in capturing outliers that aligned with manifold structures. Their evaluation on real-world hydro-turbine data also confirmed its ability to detect critical anomalies missed by conventional distance-based algorithms.

EEMST, proposed by Kaur & Sharma [5], was evaluated through IoT network simulations involving various topologies and data loads. The algorithm showed a 25% increase in energy efficiency and significantly prolonged network lifetime compared to static clustering protocols. Networks using EEMST supported more stable data transmissions with lower power consumption, highlighting the advantage of dynamic cluster head (CH) selection.

Although each approach varies in goals and techniques, a shared advantage is evident: MST can optimize graph-based structures in diverse ways (e.g., cost, distance, energy). Table 2 provides examples of real-world applications using these MST methods.

Table 2. Real-World Applications of MST Approaches

Method	**Real-World Application Example**
Kruskal+ [1]	Design of low-cost communication networks (e.g., fiber)
MST-WSN [3]	IoT-based environmental monitoring (forest, agriculture)
MST-Anomaly [4]	Industrial machine anomaly detection (predictive)
MST-IoT [5]	Smart city systems (traffic control, energy monitoring)

V. Conclusion

This article describes five primary MST-based approaches proposed in recent literature. Each approach has a specific application domain and optimization goal. The improved Kruskal algorithm (Zhang & Wang) targets computational speed without sacrificing MST optimality [1]. The MST-based clustering method using multi-objective genetic algorithms (Singh & Chauhan) provides Pareto-optimal solutions for partitioning problems involving internal vs external cluster distances. MST approaches in sensor networks (Liu & Zhao) and IoT (Kaur & Sharma) demonstrate MST's effectiveness for designing energy-efficient and reliable communication paths.

Lastly, MST-based anomaly detection (Li & Sun) adds a new dimension to data analysis by capturing outlier patterns via tree structures. The comparative results show that no single method is superior in all contexts. Instead, method selection depends on the objective: if computational efficiency on large graphs is critical, the improved Kruskal algorithm [1] is suitable; if balanced clustering on large data is desired, GA-based methods [2] are preferable; if focus is on energy saving in physical networks, WSN/IoT approaches [3][5] are effective. MST-based anomaly detection [4] is particularly useful in scientific and industrial data analysis where highdimensional outliers need identification without labels.

For future research, it is recommended to develop real-world implementations (e.g., field tests of WSN and IoT networks, industrial case studies) and to combine the above techniques. For instance, a multi-objective GA could be combined with improved Kruskal for more efficient clustering. Additionally, adapting MST-based anomaly detection for real-time data streaming is worth exploring. Thus, this article provides a comprehensive understanding of MST usage in various optimization applications, bridging graph algorithm theory with real-world practice.

References

[1] Zhang, X. & Wang, Y., "A fast and efficient algorithm for minimum spanning tree based on improved Kruskal's algorithm," Journal of Computational and Applied Mathematics, vol. 38, pp. 113–126, 2022.

[2] Singh, A. & Chauhan, D. S., "Minimum Spanning Tree Clustering Using a Multi-Objective Genetic Algorithm," Expert Systems with Applications, vol. 11, pp. 1143–1155, 2021.

[3] Liu, J. & Zhao, X., "Minimum spanning tree approach for wireless sensor network optimization," Ad Hoc Networks, vol. 10, Art. 103019, 2023.

[4] Li, W. & Sun, Z., "Minimum Spanning Tree-based Anomaly Detection in High-Dimensional Data,"

KnowledgeBased Systems, vol. 18, pp. 106–128, 2020.
[5] Kaur, A. & Sharma, V., "Energy Efficient Routing in IoT using Minimum Spanning Tree with Clustering," Computer
[6] Communications, vol. 186, pp. 2–8, 2022.
[7] Xu, Y., Olman, V. & Xu, D., "Minimum spanning trees for gene expression data clustering," Genome Informatics, vol. 12, pp. 24–33, 2001.
[8] Ahmed, I., Dagnino, A. & Ding, Y., "Unsupervised Anomaly Detection Based on Minimum Spanning Tree Approximated Distance Measures and Its Application to Hydropower Turbines," IEEE Transactions on Automation Science and Engineering, vol. 16, no. 4, pp. 1988–1999, 2018.
[9] Li, H., Xia, Q. & Wang, Y., "Research and Improvement of Kruskal Algorithm," Journal of Computer and Communications, vol. *, no. 12, pp. 63–69, 2019. [9] Rego, C. & Mathew, F., "Scatter search for capacitated minimum spanning tree," Advances in Engineering Software, Art. 103555, 2023.
[10] Singh, K. & Sundar, S., "Artificial bee colony algorithm using permutation encoding for the bounded diameter minimum spanning tree problem," Soft Computing, vol. 16, pp. 11289–11305, 2021.
[11] McLauchlan, L., Saha, S., Montiel, C. & Challoo, R., "Comparative Study of Various Wireless Sensor Network (WSN) Topology Construction Protocols," International Journal of Networked and Distributed Computing, 2014. Computing, 2, 189–197.