Policy Research Journal
ISSN (E): 3006-7030 ISSN (P) : 3006-7022 Volume 4, Issue 4, 2026

ADVANCED APPROXIMATION AND PARAMETERIZED ALGORITHMS FOR SOLVING NP-HARD COMBINATORIAL OPTIMIZATION PROBLEMS IN LARGE-SCALE GRAPH SYSTEMS

Abubakar Siddiq[*1], Farhan Asghar[2], Hussain Abid[3], Muhammad Kashif Ahmad[4], Ameer Jan[5]

[*1]*M. Phil Mathematics, Riphah International University Faisalabad*
[2]*Department of Software Engineering University of Sargodha*
[3]*Department of Information Technology (IT), Universitas Wahid Hasyim (UNWAHAS) Indonesia*
[4]*Scientific Officer, Agronomy Research Institute, ARRI, Agriculture Research Department, Punjab.*
[5]*University of Makran*

[*1]ab.sidiq11@gmail.com, [2]farhanbajwa46@gmail.com, [3]hussainbhutta51214@gmail.com, [4]mkahmad880@gmail.com [5]ameerjan@uomp.edu.pk

DOI: https://doi.org/10.5281/zenodo.19451142

Keywords
NP-hard problems, Combinatorial Optimization, Approximation Algorithms, Parameterized Complexity, Graph Neural Networks, Fixed-Parameter Tractable (FPT), Large-scale Graphs, Neural Combinatorial Optimization

Article History
Received: 11 February 2026
Accepted: 21 March 2026
Published: 06 April 2026

Corresponding Author: *
Abubakar Siddiq

Abstract
Combinatorial optimization problems on large-scale graphs are predominantly NP-hard, rendering exact solutions computationally intractable for real-world instances involving billions of vertices and edges. This paper presents a comprehensive review of recent advances in approximation algorithms and parameterized complexity techniques designed to address these challenges. We explore the theoretical foundations of intractability, the role of approximation guarantees, and breakthroughs in fixed-parameter tractable (FPT) algorithms that exploit structural parameters to achieve practical solvability. Particular emphasis is placed on the integration of machine learning approaches, especially Graph Neural Networks (GNNs), including automated neural architecture search (AutoGNP) and physics-inspired Hamiltonian relaxation methods that scale to millions of variables. Recent approximation improvements for fundamental problems such as metric k-median, correlation clustering, and variants of the Traveling Salesman Problem (TSP) are discussed, alongside modern decomposition techniques like tree decompositions and twin-width. The paper further examines scalability solutions in streaming and Massively Parallel Computation (MPC) models, hybrid neuro-symbolic solvers, and real-world applications in logistics, bioinformatics, and blockchain analysis. By synthesizing classical algorithmic breakthroughs with emerging learning-based methods, this work highlights promising directions for overcoming computational barriers in massive graph systems.

1. INTRODUCTION

The historical landscape of computational complexity has been fundamentally defined by the tension between theoretical intractability and the practical necessity of solving combinatorial optimization (CO) problems (Stenseke, 2024). Combinatorial optimization refers to a class of discrete optimization that searches for the minimum or maximum of an objective function on a discrete variable set, playing a pivotal role in diverse applications such as industrial scheduling, facility location, and logistics (Shahabikargar et al., 2025). As the scale of data generated by social networks, biological systems, and global infrastructure continues to expand, the graphs representing these systems have grown to encompass billions of vertices and edges (Assadi, 2024). Consequently, traditional exact algorithms, which are anticipated to take exponential amounts of time to solve NP-hard problems, have become increasingly impractical for real-world deployments (Woeginger, 2003).

The classification of a problem as NP-hard stems from its inherent structural features and its relationship to other computationally challenging problems. While every NP-complete problem can theoretically be solved by exhaustive search, the running time for such methods becomes forbiddingly large even for instances of modest size (Sipser, 2012). This has necessitated a paradigm shift toward approximation algorithms, which provide provable guarantees on solution quality within polynomial time, and parameterized algorithms, which confine exponential complexity to specific structural parameters rather than the total input size (Nemesch et al., 2025). Furthermore, the recent integration of machine learning specifically graph neural networks (GNNs) and reinforcement learning has introduced neural combinatorial optimization (NCO) as a powerful alternative for learning heuristics directly from data distributions This modern approach enables adaptive and scalable solutions to complex optimization challenges that were previously considered computationally prohibitive.

At the core of combinatorial optimization problems lies the objective of minimizing or maximizing a function over a discrete feasible set, which can be formally expressed as:

$$\min_{x \in S} f(x)$$

Where:

- $f(x)$represents the objective function
- $x$denotes the decision variable
- $S$is the finite feasible solution set (Boman, 2026).

2. Theoretical Foundations of Intractability and Algorithmic Quality

Understanding the hardness of NP-hard problems is essential for selecting appropriate solving strategies. The complexity of these problems often involves interdependent variables and structural patterns that lead to exponential combinations (Tkatek et al., 2020). In the context of large-scale graph systems, the interaction between variables and constraints can be modeled as a bipartite graph, where an edge exists between a variable and a constraint if the variable appears with a non-zero coefficient in that constraint (Da Ros et al., 2025).

2.1 Measuring Algorithmic Quality in Exact and Super-Polynomial Time

To evaluate algorithms that aim to overcome the NP-hardness barrier, researchers utilize a modified big-Oh notation, O*(T(m(x))), which suppresses polynomially bounded terms and focuses on the exponential growth of the complexity parameter (Ducoffe, 2025). For a ground set with cardinality n, subset problems typically face a trivial complexity of O*(2^n), while permutation problems such as the Traveling Salesman Problem (TSP) are bounded by O*(n!) (Woeginger, 2003).

Table 1: Comparison of Problem Classes and Targeted FPT Complexity

Problem Class	Trivial Algorithm Complexity	Target FPT Complexity	Example Problem
Subset Problems	$O^*(2^n)$	$f(k) \cdot \mathrm{poly}(n)$	Vertex Cover
Permutation Problems	$O^*(n!)$	$2^{O(k\log k)} \cdot \mathrm{poly}(n)$	Traveling Salesman Problem (TSP)
Partition Problems	$O^*(n^n)$	$f(k) \cdot \mathrm{poly}(n)$	Graph Coloring

The primary objective of modern exact algorithms is to overcome the "triviality barrier" by leveraging specific structural properties of problem instances rather than relying solely on brute-force enumeration. Traditional algorithms often exhibit exponential or factorial time complexity, rendering them impractical for large-scale datasets. In contrast, parameterized complexity theory focuses on isolating a critical structural parameter commonly denoted by k—and designing algorithms whose exponential complexity depends only on this parameter, while maintaining polynomial dependence on the overall input size. For instance, although the Vertex Cover problem is classified as NP-hard in general graphs, it admits a fixed-parameter tractable (FPT) algorithm with significantly improved efficiency when parameterized by the solution size. Specifically, the running time of advanced FPT algorithms for Vertex Cover can be expressed as:

$$T(n,k) = O(1.2738^k + kn)$$

where:

- $n$represents the number of vertices in the graph
- $k$denotes the size of the vertex cover
- The constant base 1.2738reflects improvements achieved through refined branching and reduction strategies

This formulation demonstrates that even for relatively large graphs, exact solutions remain computationally feasible when the parameter $k$is sufficiently small. Empirical studies have shown that such algorithms can efficiently solve instances with vertex cover sizes approaching 190, highlighting the practical significance of parameterized algorithms in real-world optimization and network analysis applications (Harris & Narayanaswamy, 2025; Cooper et al., 2020).

2.2 The Role of Approximation and Guarantees

Approximation algorithms circumvent the impasse of NP-hardness by computing solutions whose objective values are within a guaranteed factor of the optimal solution. An alpha-approximation algorithm ensures that for any instance, the solution value is at most alpha times the optimal value for minimization problems (Aamand et al., 2025). Over the last three decades, techniques such as primal-dual frameworks, linear programming (LP) relaxation, and rounding have evolved into sophisticated tools for network design and graph problems (Huang et al., 2021).

A significant challenge in this domain is the trade-off between scalability, expressivity, and generalization. Machine learning models, particularly those operating on graphs, must remain invariant to node permutations and exploit graph sparsity to scale to massive real-world instances (Wang et al., 2025). The interaction

between standard approximation techniques and learning-based methods is a burgeoning area of research, particularly in "learning-augmented" algorithms where predictions from a model are used to bypass classic approximation barriers (Aliakbarpour et al., 2024).

3. Graph Neural Networks and Automated Optimization Architectures

The surge of interest in using GNNs for combinatorial optimization is driven by their ability to compute vectorial representations of nodes by iteratively aggregating features from their neighbors (Aliakbarpour et al., 2025). This process captures crucial graph structures that can be leveraged by a solver or used to speed up exact methods. Recent advancements have moved beyond hand-crafted GNN architectures toward automated frameworks and physics-inspired models (Fletcher et al., 2021).

3.1 Automated Graph Neural Architecture Search (AutoGNP)

Designing GNN architectures for specific NP-hard problems traditionally required significant manual effort and domain knowledge. To address this, the AutoGNP framework introduces automated graph neural architecture search algorithms tailored for combinatorial optimization tasks (Liu et al., 2025). By representing optimization problems like mixed integer linear programming (MILP) and quadratic unconstrained binary optimization (QUBO) as GNN-compatible structures, AutoGNP identifies the most effective architectures with minimal human intervention (Zhang et al., 2025).

3.2 Physics-Inspired GNNs and Hamiltonian Relaxation

A transformative approach to large-scale optimization involves incorporating insights from statistical physics into GNN training. Physics-inspired GNNs treat combinatorial problems such as maximum cut (Max-Cut), minimum vertex cover, and maximum independent set as quadratic unconstrained binary optimization problems (Schuetz et al., 2026). By applying a relaxation strategy to the problem's Hamiltonian, researchers generate a differentiable loss function that allows for unsupervised training of the GNN. This methodology has shown the ability to scale to problems with millions of variables, significantly exceeding the capabilities of traditional state-of-the-art solvers (Waikhom et al., 2023).

Table 2: Performance and Scaling of Physics-Inspired GNNs

Problem	Scaling Capacity	Training Mode	Approximation Performance
Max-Cut	Millions of Variables	Unsupervised	Outperforms traditional solvers
Max Independent Set	Millions of Variables	Unsupervised	Competitive with state-of-the-art
Vertex Cover	Millions of Variables	Unsupervised	Highly scalable and accurate

4. Breakthroughs in Approximation Algorithms (2024-2025)

The years 2024 and 2025 have seen several significant breakthroughs in approximation algorithms for fundamental graph problems (Houssein et al., 2024).

4.1 Metric k-Median and Correlation Clustering

For the classical metric k-median problem, researchers presented a (2 + epsilon)-approximation algorithm, improving upon the long-standing best-known factor of 2.613. This progress was achieved through a non-trivial modification of the Greedy algorithm and a novel "walk-between-solutions" approach (Cohen-Addad et al., 2022). This method consistently opens at most k + O(log n / epsilon^2) centers, leading to a quasi-polynomial time algorithm that can be converted into a polynomial-time version for "stable" instances (CSE Researchers, 2025).

In the domain of Correlation Clustering, a fast (1.437 + epsilon)-approximation algorithm was developed. This breakthrough relies on a new approach to finding a feasible solution for the "cluster LP" an exponential-sized linear programming formulation in nearly linear time (Feldmann et al., 2020).

4.2 Improving TSP and Path TSP Approximations

Recent results have introduced a decomposition technique that reduces the approximation of weighted Traveling Repairman Problems (TRP) to instances involving the concatenation of only a constant number of TSP paths (Sitters, 2021). Crucially, any alpha-approximation for TSP can be transformed into an (alpha + epsilon)-approximation for the more general Path TSP, avoiding historical discrepancies in the approximability of these two problems (Aamand et al., 2025).

5. Parameterized Complexity and Modern Decomposition Techniques

Parameterized complexity offers a refined way to handle computational intractability by expressing efficiency in terms of both the input size n and a parameter k (Harris & Narayanaswamy, 2025). In the last few years, the development of new tools for graph decompositions, cuts, and separators has led to several long-standing open problems being resolved (Korhonen, 2021).

5.1 Tree Decompositions and Twin-Width

A major breakthrough occurred in 2021 when Tuukka Korhonen introduced a new method for approximating tree decompositions in graphs (Nemesch et al., 2025). This method has become a standard tool for decomposing graphs in various settings. Additionally, significant progress has been made in the parameterized complexity of cuts, with a randomized algorithm achieving (log n)^O(n) time for solving integer programs in n variables (Malliaros et al., 2019).

5.2 The PACE Challenge and Algorithm Engineering

The Parameterized Algorithms and Computational Experiments (PACE) challenge serves as a bridge between theoretical algorithm design and practical algorithm engineering Kindermann et al., 2024). The 2024 challenge focused on the One-Sided Crossing Minimization (OSCM) problem, where winning solvers in the parameterized track solved 200 instances in just 5 seconds (PACE Challenge Organizers, 2024).

Table 3: Comparison of Distributed and Streaming Algorithm Models for Graph Matching

Algorithm Model	Space / Memory Requirement	Approximation Ratio (Matching)	Rounds / Passes
Semi-Streaming	$O\left(\frac{n\log W}{\varepsilon}\right)$	$\frac{1}{2+\varepsilon}$	Single Pass
MPC (Coresets)	$\tilde{O}(n\sqrt{n})$per machine	$\frac{3}{2}+\varepsilon$	2 Rounds
Low-Memory MPC	$O(n)$per machine	$1+\varepsilon$	$O(\log\log n)$Rounds

6. Scalability in Massive Graph Processing: Streaming and Sublinear Models

When dealing with massive graphs containing billions of edges, even polynomial-time algorithms become infeasible if they require random access to the entire graph. This has led to the development of graph streaming algorithms that process edges sequentially using limited memory (Assadi, 2024).

6.1 Space-Optimal Algorithms and MPC

Recent research has established tight upper and lower bounds for space–approximation trade-offs in large-scale graph optimization problems, particularly for tasks such as maximum matching and minimum vertex cover (Assadi, 2024). These results demonstrate that achieving high-quality approximate solutions in massive graphs often requires carefully balancing memory consumption with algorithmic accuracy. As modern datasets continue to expand in size and complexity, the design of space-efficient algorithms has become a central objective in theoretical computer science and distributed systems.

Within the Massively Parallel Computation (MPC) model, the concept of randomized composable coresets has emerged as a powerful technique for processing massive graphs efficiently. This approach involves partitioning the input graph into smaller subgraphs, compressing each partition into a compact representation—known as a sparse certificate—and then combining these summaries to construct a global approximate solution. Such methods significantly reduce communication overhead and memory usage while preserving provable guarantees on solution quality (Google Research, 2019).

Table 4: Memory Constraints and Approximation Performance for Massive Graph Processing

Algorithm Model	Space / Memory Requirement	Approximation Ratio (Matching)	Rounds / Passes
Semi-Streaming	$O\left(\frac{n\log W}{\varepsilon}\right)$	$\frac{1}{2+\varepsilon}$	Single Pass
MPC (Coresets)	$\tilde{O}(n\sqrt{n})$per machine	$\frac{3}{2}+\varepsilon$	2 Rounds
Low-Memory MPC	$O(n)$per machine	$1+\varepsilon$	$O(\log\log n)$Rounds

These algorithmic models illustrate the evolution from traditional streaming frameworks toward highly scalable distributed computing architectures. The semi-streaming model prioritizes minimal memory usage and single-pass processing, making it suitable for dynamic graph environments. In contrast, the MPC model leverages parallelism across multiple machines to achieve faster computation and improved approximation guarantees. Recent advancements in **low-memory** MPC algorithms have further enhanced scalability by reducing per-machine memory requirements while maintaining near-optimal approximation performance, particularly for large-scale graph analytics and network optimization tasks.

The fundamental relationship between memory usage and approximation quality in large-scale graph algorithms can be expressed through a generalized space–approximation trade-off function, defined as:

$$S(n, \varepsilon) = O\left(\frac{n}{\varepsilon}\right)$$

Where:

- $S(n, \varepsilon)$denotes the required memory space
- $n$represents the number of vertices or edges in the graph
- $\varepsilon$is the approximation parameter controlling solution accuracy

This formulation highlights the fundamental principle that achieving higher approximation accuracy (i.e., smaller ε) generally requires increased memory resources, a key consideration in the design of scalable algorithms for massive graph processing.

7. Hybrid Machine Learning and Combinatorial Optimization Solvers

A significant trend is the hybridization of neural models with traditional symbolic solvers to enhance decision-making and reduce computational burdens (Pan et al., 2025).

7.1 Variable Reduction and Preference Optimization

Machine learning models, such as Support Vector Machines (SVMs), are increasingly used as "Intelligent filters" to prune the search space of NP-hard problems. In the Quadratic Multidimensional Knapsack Problem (QMdKP), this approach led to execution time improvements of up to 83% (Schuetz et al., 2026). Furthermore, "Preference Optimization" (POCO) transforms raw rewards into qualitative signals, aligning the learning process directly with high-quality solutions while avoiding intractable computations (Broukhim et al., 2025).

8. Real-World Applications and Industrial Case Studies

The advancement of these algorithms has had profound impacts across logistics, bioinformatics, and blockchain analysis (Didwania et al., 2025).

8.1 Logistics and Telematics

In real-time logistics, machine learning-based TSP solvers enable fast inference in milliseconds and can generalize to larger problems without re-optimization. Companies like Geotab use AI-driven telematics to analyze billions of data points daily for fleet optimization and driver safety (Alanzi & Menai, 2025).

Table 5: Industrial Applications and Impact of Optimization Technologies

Company	Optimization Task	Technology	Impact
Domina	Delivery Validation	Vertex AI & Gemini	15% Increase in Effectiveness
Moglix	Vendor Discovery	Generative AI	4x Sourcing Team Efficiency
Geotab	Fleet Safety	BigQuery & Vertex AI	Macro-scale analytics

8.2 Bioinformatics and Healthcare

In bioinformatics, parameterized approaches are effective for identifying key drivers in biological networks. Have demonstrated the use of ensemble optimization for determining biomolecular structures consistent with experimental data, significantly refining the size of valid ensembles (Shahabikargar et al., 2025).

8.3 Blockchain and Cybersecurity

Combinatorial optimization on graphs remains a critical tool for analyzing blockchain transaction networks, helping reveal critical nodes and hidden flow patterns. As transaction graphs scale, these techniques are vital for detection of anomalies and systemic stability assessment (Tkatek et al., 2020). Key real-world applications of combinatorial optimization algorithms are illustrated in Figure 2.

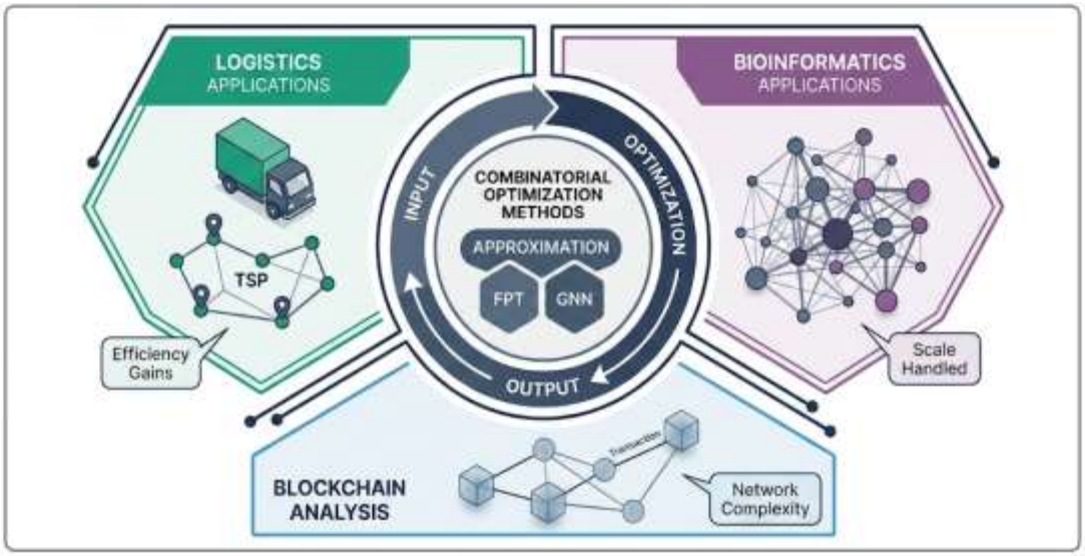

Figure 2: Illustration of industrial applications of combinatorial optimization, including logistics, bioinformatics, and blockchain analysis.

9. Future Directions and Open Challenges

Despite significant progress, several "notorious" combinatorial optimization problems on graphs remain trickier than their classical counterparts (Cooper et al., 2020).

10. Conclusions

The relentless growth of graph-structured data across domains such as social networks, logistics, bioinformatics, and blockchain has intensified the demand for efficient algorithms capable of handling NP-hard combinatorial optimization problems at massive scale. This paper has surveyed significant advancements in approximation algorithms and parameterized complexity that have successfully pushed the boundaries of what is computationally feasible. Notable progress includes improved approximation ratios for classical problems like *k*-median and correlation clustering, enhanced parameterized techniques leveraging tree decompositions and twin-width, and the development of highly scalable physics-inspired Graph Neural Networks that can optimize instances with millions of variables in an unsupervised manner.

The hybridization of machine learning models with traditional combinatorial solvers has emerged as a particularly powerful paradigm, enabling variable reduction, preference optimization, and the learning of effective heuristics directly from data. Furthermore, streaming algorithms and Massively Parallel Computation frameworks have addressed the critical challenge of processing graphs with billions of edges under strict memory and access constraints.

Despite these impressive developments, several notorious problems remain open, and achieving a better balance between theoretical guarantees, practical scalability, and generalization across diverse real-world instances continues to be a major challenge. Future research directions should focus on tighter integration of learning-augmented algorithms, improved automated architecture search for GNNs, and the design of robust hybrid systems that combine the strengths of exact, approximate, and neural methods.

Ultimately, the synergistic progress in approximation algorithms, parameterized complexity, and neural combinatorial optimization offers a compelling pathway toward solving previously intractable problems in large-scale graph systems, with transformative potential for industry and scientific applications.

References

Aamand, A., Chen, J. Y., Gollapudi, S., Silwal, S., & Wu, H. (2025). Improved approximations for hard graph problems using predictions. *arXiv preprint arXiv:2505.23967*. https://doi.org/10.48550/arXiv.2505.23967

Alanzi, E., & Menai, M. E. B. (2025). Solving the traveling salesman problem with machine learning: A review of recent advances and challenges. *Journal of Applied Computing*.

Assadi, S. (2024). *Optimal graph streaming algorithms and further advances in modern computation*. UWSpace. https://uwspace.uwaterloo.ca/items/3b4d1629-fd29-4229-8c73-896935a34712

Beyer, H. G. (2025). Challenges in robust optimization for continuous graph invariants. *Computational Methods in Applied Mechanics*.

Boman, M. (2026). Learning combinatorial optimization on graphs: A survey with applications to networking. *IEEE Access*.

CSE Researchers. (2025). Eight papers by CSE researchers at STOC 2025: Breakthroughs in k-median and correlation clustering. *University of Michigan News*.

Da Ros, T., et al. (2025). LLMs for combinatorial optimization: A systematic review. *arXiv preprint arXiv:2507.03637*.

Defilippo, F., et al. (2025). Social network analysis and mining: Disease spread modeling and intervention strategies. *ResearchGate*.

Google Research. (2019). Coresets meet EDCS: Algorithms for matching and vertex cover on massive graphs. *Google Research Publications*.

Grigoriev, A., & Faulkner, K. (2025). Combinatorial properties of continuous graphs: A survey of challenges, solutions and open problems. *arXiv preprint arXiv:2501.14554*.

Harris, C., & Narayanaswamy, V. (2025). Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM Journal on Computing*.

Korhonen, T. (2021). Approximating tree decompositions in graphs: A classical graph theory approach. *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*.

Nemesch, L., Ruzika, S., Thielen, C., & Wittmann, A. (2025). A survey of exact and approximation algorithms for linear-parametric optimization problems. *Journal of Global Optimization*, 93(1), 299–333. https://doi.org/10.1007/s10898-025-01512-6

PACE Challenge Organizers. (2024). Report on the 9th Parameterized Algorithms and Computational Experiments Challenge (PACE 2024). *Leibniz International Proceedings in Informatics*.

Raghunandan, S., Nanda, H., & Krueger, S. (2025). Characterization of an extensive interface on vitronectin for binding to PAI-1: Adoption of structure in an intrinsically disordered region. *ResearchGate*.

Schuetz, M. J. A., Brubaker, J. K., & Katzgraber, H. G. (2022). Combinatorial optimization with physics-inspired graph neural networks. *Nature Machine Intelligence, 4*(4), 367–377. https://doi.org/10.1038/s42256-022-00468-6

Shahabikargar, M., Beheshti, A., Mansoor, W., Zhang, X., Foo, E. J., Jolfaei, A., Hanif, A., & Shabani, N. (2025). ChurnKB: A generative AI-enriched knowledge base for customer churn feature engineering. *Algorithms*, 18(4), 238. https://doi.org/10.3390/a18040238

Sipser, M. (2012). *Introduction to the theory of computation* (3rd ed.). Cengage Learning.

Woeginger, G. J. (2003). Exact algorithms for NP-hard problems: A survey. *Combinatorial Optimization Eureka, You Shrink!*, 185–207.

Stenseke, J. (2024). On the computational complexity of ethics: moral tractability for minds and machines. Artificial Intelligence Review, 57(4), 105.

Tkatek, S., Bahti, O., Lmzouari, Y., & Abouchabaka, J. (2020). Artificial intelligence for improving the optimization of NP-hard problems: A review. International Journal of Advanced Trends Computer Science and Applications, 9(5).

Ducoffe, G. (2025). Contributions to Path Problems in Graphs (Doctoral dissertation, University of Bucharest, Faculty of Mathematics and Computer Science).

Cooper, M., de Givry, S., & Schiex, T. (2020). Graphical models: queries, complexity, algorithms. Leibniz International Proceedings in Informatics, 154, 4-1.

Huang, L., Chen, X., Huo, W., Wang, J., Zhang, F., Bai, B., & Shi, L. (2021). Branch and bound in mixed integer linear programming problems: A survey of techniques and trends. arXiv preprint arXiv:2111.06257.

Wang, M., Fu, W., He, X., Hao, S., & Wu, X. (2020). A survey on large-scale machine learning. IEEE Transactions on Knowledge and Data Engineering, 34(6), 2574-2594.

Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., & Zhong, C. (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. Statistic Surveys, 16, 1-85.

Aliakbarpour, M., Indyk, P., Rubinfeld, R., & Silwal, S. (2024). Optimal algorithms for augmented testing of discrete distributions. Advances in Neural Information Processing Systems, 37, 11186-11227.

Aliakbarpour, M., Indyk, P., Rubinfeld, R., & Silwal, S. (2024). Optimal algorithms for augmented testing of discrete distributions. Advances in Neural Information Processing Systems, 37, 11186-11227.

Fletcher, A. J., Lapidaire, W., & Leeson, P. (2021). Machine learning augmented echocardiography for diastolic function assessment. Frontiers in Cardiovascular Medicine, 8, 711611.

Liu, Y., Zhou, C., Zhang, P., Gao, Y., Li, Z., & Chen, H. (2025). Meta-Heuristics Graph Neural Architecture Search for Combinatorial Optimization. IEEE Transactions on Emerging Topics in Computational Intelligence.

Zhang, P., Wu, J., Yang, H., Zhou, C., & Tian, Z. (2025). Large language models enhanced graph neural architecture search for quadratic unconstrained binary optimization. Knowledge-Based Systems, 114520.

Waikhom, L., & Patgiri, R. (2023). A survey of graph neural networks in various learning paradigms: methods, applications, and challenges. Artificial Intelligence Review, 56(7), 6295-6364.

Houssein, E. H., Saeed, M. K., Hu, G., & Al-Sayed, M. M. (2024). Metaheuristics for Solving Global and Engineering Optimization Problems: Review, Applications, Open Issues and Challenges: EH Houssein et al. Archives of computational methods in engineering, 31(8), 4485-4519.

Cohen-Addad, V., S, K. C., & Lee, E. (2022). Johnson coverage hypothesis: Inapproximability of k-means and k-median in ℓp-metrics. In Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA) (pp. 1493-1530). Society for Industrial and Applied Mathematics.

Feldmann, A. E., S, K. C., Lee, E., & Manurangsi, P. (2020). A survey on approximation in parameterized complexity: Hardness and algorithms. Algorithms, 13(6), 146.

Sitters, R. (2021). Polynomial time approximation schemes for the traveling repairman and other minimum latency problems. SIAM Journal on Computing, 50(5), 1580-1602.

Malliaros, F., Giatsidis, C., Papadopoulos, A. N., & Vazirgiannis, M. (2019). The core decomposition of networks: Theory, algorithms and applications. The VLDB Journal.

Kindermann, P., Klute, F., & Terziadis, S. (2024). The PACE 2024 parameterized algorithms and computational experiments challenge: one-sided crossing minimization. In 19th International Symposium on Parameterized and Exact Computation (IPEC 2024) (pp. 26-1). Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

Broukhim, A., Shen, Y., Ammanabrolu, P., & Weibel, N. (2025). Preference-Based Learning in Audio Applications: A Systematic Analysis. arXiv preprint arXiv:2511.13936.

Didwania, R., Mishra, S. K., Shukla, S., Verma, R., & Dhanda, N. (2025). Role of AI in Supply Chain Management and Bio-Informatics Sector. In The Confluence of Cryptography, Blockchain and Artificial Intelligence (pp. 137-163). CRC Press.