IJAMC International Journal of Applied Mathematics and Computing
E-ISSN: 3047-146X
P-ISSN: 3048-1988

Research Article

Algorithmic Simulation for Optimization in Combinatorial Mathematics Using Heuristic Techniques

Ahmad Budi Trisnawan[1]*, Syed Asif Ali[2], Erlita Sulistiati[3]

[1] Universitas Mahakarya Asia, Indonesia; Email: abudit75@gmail.com
[2] University Karachi, Pakistan; Email: aasyed@smiu.efu.pk
[3] Universitas Mahakarya Asia, Indonesia; Email: erlita14@gmail.com
* Corresponding Author: abudit75@gmail.com

Abstract: This research explores the effectiveness of heuristic techniques for solving combinatorial optimization problems, with a particular focus on the Traveling Salesman Problem (TSP). Combinatorial optimization is a critical area of study, especially in fields like computer science, engineering, and economics, where finding optimal solutions from a finite set of possibilities is crucial. However, the NP-hard nature of many combinatorial problems, such as the TSP, makes traditional exact methods like Branch-and-Bound and Dynamic Programming computationally expensive and inefficient for larger problem sizes. The primary objective of this research is to evaluate the performance of heuristic methods, including Simulated Annealing (SA), Genetic Algorithms (GA), and Iterative Computation techniques, such as Tabu Search (TS) and Particle Swarm Optimization (PSO). These methods are tested for their ability to provide approximate solutions efficiently. The findings reveal that while ACO provided the best solution quality, it had the longest runtime. TS was the fastest, though with slightly lower solution quality. SA and GA demonstrated a balance between solution quality and computational efficiency, but their performance heavily depended on parameter tuning. The hybridization of SA and GA showed potential for improving solution quality but introduced additional complexity. The research concludes that heuristic methods, especially when combined, offer viable solutions for large-scale combinatorial optimization problems, though the trade-off between solution quality and computational time must be considered when selecting an algorithm.

Keywords: Combinatorial Optimization; Genetic Algorithms; Heuristic Algorithms; Simulated Annealing; Traveling Salesman Problem.

Received: 11 May 2025
Revised: 18 June 2025
Accepted: 27 July 2025
Published: 30 July 2025
Curr. Ver.: 30 July 2025

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY SA) license (https://creativecommons.org/licenses/by-sa/4.0/)

1. Introduction

Combinatorial optimization plays a pivotal role in various domains by providing efficient solutions to problems involving the selection or arrangement of a finite set of possibilities. Its applications span across multiple fields, significantly improving system performance and resource utilization. In logistics, for instance, combinatorial optimization is used to optimize vehicle routing and scheduling, which can substantially reduce transportation costs and enhance supply chain efficiency (Bao, Le, & Nguyen, 2018). Similarly, in scheduling, particularly in academic events and resource allocation, optimization ensures the efficient management and utilization of resources (Juan, Chica, De Armas, & Kelton, 2016). Additionally, combinatorial optimization is crucial in telecommunications, where it is applied in routing, network design, and load balancing to ensure efficient resource distribution (Mbarek & Mosorov, 2019). Beyond these fields, combinatorial optimization also finds applications in information technology, transportation, economics, management, and network communication, highlighting its broad relevance (Castañedalozano & Schulte, 2020; Tao & Chen, 2025).

Despite its widespread applicability, solving combinatorial optimization problems remains computationally challenging, particularly as the problem size increases. Most combinatorial optimization problems are NP-hard, meaning that the number of possible solutions grows exponentially with the size of the problem, making exact solutions

DOI : https://doi.org/10.62951/ijamc.v2i3.274

impractical for large-scale problems (Bojnordi & Ipek, 2017). One of the primary issues encountered is state-space explosion, where the number of possible solutions becomes so vast that solving the problem using exact methods becomes computationally infeasible within a reasonable timeframe (Purkayastha, Chakraborty, Saha, & Mukhopadhyay, 2020). Moreover, the computational complexity of exact methods, though ensuring optimal solutions, makes them prohibitively expensive for real-world, large-scale problems (Raheem & Shabat, 2024). Furthermore, modern systems characterized by high levels of dynamism and uncertainty pose additional challenges, which exact methods struggle to handle effectively (Duan, Jiang, Dai, Wang, & He, 2023).

In response to these challenges, various heuristic and metaheuristic approaches have been developed to provide feasible, near-optimal solutions within a limited timeframe. Heuristic approaches, while fast, may not always guarantee optimality but are highly effective for real-time decision-making (Cappart et al., 2021). Metaheuristics, such as Ant Colony Optimization, Genetic Algorithms, and Simulated Annealing, are particularly useful for large-scale problems, offering good-quality solutions (Purkayastha et al., 2020; Juan et al., 2016). Furthermore, hybrid algorithms, which combine exact methods with heuristic or metaheuristic approaches, have shown great potential in leveraging the strengths of both techniques to deliver more efficient and high-quality solutions (Duan et al., 2023). Recent advances in deep learning have also introduced novel possibilities for solving combinatorial optimization problems, where learned models can be generalized to larger problem spaces, offering promising results (Tao & Chen, 2025).

The primary objective of this study is to explore the efficiency of heuristic techniques in solving combinatorial optimization problems, which are prevalent in fields such as computer science, engineering, and economics. Combinatorial optimization involves finding the best possible solution from a finite set of possibilities, a task that can become computationally intensive, especially for NP-hard problems (Asani et al., 2019). These types of problems grow exponentially as the problem size increases, making traditional exact methods impractical due to their high computational complexity (Ouassam, Hmina, Bouikhalene, & Hachimi, 2021). Therefore, heuristic methods, which provide approximate solutions within a reasonable timeframe, become essential for solving these complex problems (Juan, Chica, De Armas, & Kelton, 2016).

This study focuses on three prominent heuristic methods that have shown effectiveness in solving combinatorial optimization problems. Simulated Annealing, a probabilistic technique inspired by the annealing process in metallurgy, explores the solution space by accepting worse solutions with a certain probability to escape local optima, gradually converging toward a near-optimal solution (Purkayastha, Chakraborty, Saha, & Mukhopadhyay, 2020). Genetic Algorithms, based on principles of natural selection and genetics, evolve a population of solutions through selection, crossover, and mutation, making them particularly effective for problems like the Traveling Salesman Problem (TSP) (Castañedalozano & Schulte, 2020). Iterative Computation methods, such as Iterative Improvement and Tabu Search, refine solutions through small local changes, improving solution quality by escaping local optima, while balancing solution quality and computational time (Ouassam et al., 2021). These methods collectively demonstrate their applicability in solving complex combinatorial problems efficiently.

Heuristic techniques are designed to provide efficient solutions to complex combinatorial optimization problems, where traditional methods may be too slow or unable to guarantee exact solutions. These methods are essential for real-world applications, where computational resources are limited and there is a need for quick, feasible solutions (Duan et al., 2023). The efficiency of these heuristic methods varies depending on the problem and the specific technique used. For instance, Simulated Annealing and Genetic Algorithms perform well on problems like the TSP, providing good solutions within acceptable timeframes (Purkayastha et al., 2020). Iterative methods, such as Tabu Search, are effective at improving solution quality through repeated refinements (Raheem & Shabat, 2024).

This study aims to implement and compare these heuristic techniques on classical combinatorial optimization problems, such as the TSP, to evaluate their performance in terms of solution quality and computational efficiency. Furthermore, the research will explore the potential for hybridizing these methods to enhance their effectiveness in solving complex combinatorial problems (Duan et al., 2023).

2. Literature Review

Combinatorial optimization techniques are crucial in solving complex problems across various fields like computer science, engineering, and economics. Classical methods such as Branch-and-Bound, Dynamic Programming, and Mathematical Programming are precise and well-suited for smaller or medium-sized problems. However, they struggle with larger, NP-hard problems due to high computational complexity (Kuma, Sahoo, & Dixit, 2024). Heuristic methods like Simulated Annealing (SA), Genetic Algorithms (GA), and Ant Colony Optimization (ACO) offer flexible solutions for larger problems by providing approximate results in a reasonable timeframe. SA and GA are known for high-quality solutions, though SA's performance is sensitive to parameters like the cooling schedule, while ACO excels in solving problems like the Traveling Salesman Problem (TSP) (Raheem & Shabat, 2024).

In terms of computational efficiency, classical methods like the Greedy Algorithm are fast for smaller problems but become inefficient for larger ones. Heuristic algorithms like Tabu Search (TS) and Particle Swarm Optimization (PSO) are faster and more efficient than classical approaches, but ACO tends to have longer runtimes and greater variability in results (Sakib, Joy, Juel, & Hasan, 2025). Hybrid approaches, combining multiple heuristics, can improve solution quality and efficiency by leveraging the strengths of each method. However, they can increase computational complexity and runtime, which may be a disadvantage for real-time or resource-constrained applications (Sakib et al., 2025).

Overview of Classical Combinatorial Optimization Techniques

Combinatorial optimization techniques are essential tools for solving problems in various fields, including computer science, engineering, and economics. Among the most well-known classical methods for tackling hard combinatorial problems are Branch-and-Bound and Dynamic Programming. Branch-and-Bound systematically explores the solution space by dividing it into smaller subproblems, and it is often used for solving problems like the traveling salesman problem (TSP). On the other hand, Dynamic Programming breaks problems down into simpler subproblems and solves each only once, storing the results to avoid redundant calculations (Stracquadanio & Pardalos, 2019).

Another classical approach is Mathematical Programming, which includes techniques like linear and integer programming. These methods are widely used in operations research to model and solve optimization problems that are represented by linear relationships (Miftakhov, 2025). Graph Theory and Network Flows are also crucial in combinatorial optimization, particularly in problems involving networks. For example, finding maximum flows in a network can be tackled using combinatorial algorithms combined with linear programming duality (Yaqoob, Verma, & Aziz, 2024).

Challenges Associated with Large Problem Spaces and Non-Polynomial Time Complexity

Many combinatorial optimization problems are NP-hard, meaning that no polynomial-time algorithms are known to solve them efficiently. This makes problems such as the TSP and the knapsack problem computationally intractable as the problem size grows (Sakib et al., 2024). As the number of possible solutions increases exponentially with problem size, exhaustive search methods become impractical. This phenomenon, known as state-space explosion, is a fundamental challenge in combinatorial optimization (Raj et al., 2023).

The computational complexity of solving these problems is closely tied to the resources required to find the optimal solution. For large problems, exact methods that guarantee optimality become computationally expensive and often impractical. These limitations have spurred interest in developing alternative approaches, particularly heuristic algorithms, which offer approximate solutions within a feasible timeframe.

Review of Heuristic Approaches

Heuristic algorithms are increasingly being employed to solve complex combinatorial optimization problems more efficiently than classical methods. These methods are particularly useful for large problem spaces where exact solutions are difficult to obtain. One of the most widely used heuristic methods is Simulated Annealing (SA). SA is a stochastic optimization technique that mimics the annealing process in metallurgy, where solutions are initially allowed to deteriorate to escape local optima. Over time, the probability of accepting worse solutions decreases, helping the algorithm converge toward a near-optimal solution (Bojnordi & Ipek, 2017).

Genetic Algorithms (GA) are another prominent heuristic method inspired by the process of natural selection. GAs evolve a population of solutions through selection, crossover, and mutation to find optimal or near-optimal solutions. GAs have proven effective in solving complex problems, such as the TSP, by generating new solutions through a process of biological evolution (Cappart et al., 2021).

Other iterative methods, such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO), also play a significant role in solving combinatorial problems. These metaheuristic algorithms rely on collective behavior or biological principles to iteratively improve solutions. These methods balance exploration and exploitation, making them well-suited for large-scale combinatorial problems (Miftakhov, 2025).

Effectiveness in Solving Optimization Problems in Different Domains

Heuristic methods, particularly Simulated Annealing, have demonstrated significant effectiveness in solving timetabling problems, which are known to be NP-hard. These problems, prevalent in educational institutions, involve scheduling courses or exams while optimizing resource usage (Gusti Agung Premananda, Tjahyanto, & Muklason, 2024).

In the fields of engineering and logistics, metaheuristic algorithms like GAs and SA have been successfully applied in optimizing structural designs, managing supply chains, and controlling dynamic systems. These methods offer a robust solution to complex optimization problems in real-world applications (Sakib et al., 2024). Furthermore, the integration of machine learning with GAs has enhanced their capability to solve both constrained and unconstrained optimization problems. This hybrid approach allows the algorithms to handle more complex and dynamic problem environments (Raj et al., 2023).

Finally, hybrid approaches that combine different heuristics, such as GAs and SA, have been shown to leverage the strengths of each method, providing robust solutions for complex design problems. By combining the strengths of multiple heuristics, these hybrid models can offer more effective and efficient solutions for real-world combinatorial optimization challenges (Yaqoob et al., 2024).

Comparative Studies of Heuristic and Classical Methods

In the realm of combinatorial optimization, a variety of techniques are available, ranging from classical methods to heuristic and metaheuristic approaches. Classical methods, such as the Hungarian Algorithm, Linear Programming, and Constraint Programming, are widely known for their precision. These methods consistently deliver high accuracy across various problem sizes, making them ideal for scenarios where precision is critical. However, these techniques are often computationally expensive and struggle with larger, more complex problems (Kuma, Sahoo, & Dixit, 2024).

On the other hand, heuristic methods such as Genetic Algorithms (GA), Simulated Annealing (SA), and Ant Colony Optimization (ACO) have been evaluated for complex problems like the Traveling Salesman Problem (TSP). GA and SA are noted for their ability to provide high-quality solutions, although SA's performance is highly sensitive to factors such as the cooling schedule used during the process. ACO, however, has shown superior performance in optimizing the TSP, particularly in finding the shortest distance (Raheem & Shabat, 2024). Moreover, hybrid methods that combine heuristics, such as SA+GA, have shown potential in improving solution quality. However, this often results in increased computational complexity, making the trade-off between solution quality and computational effort an important consideration (Sakib, Joy, Juel, & Hasan, 2025).

Computational Efficiency

When considering computational efficiency, classical methods like the Greedy Algorithm are known for their speed, especially when solving smaller problems. However, as the size of the problem grows, the greedy approach faces significant time and accuracy challenges (Kuma et al., 2024). Simulated Annealing (SA) guarantees asymptotic convergence to global optima, but its practical performance is highly sensitive to initial conditions, such as the starting temperature. Modifying the cost function can significantly enhance its robustness and efficiency in certain cases (Chen & Nurdin, 2019).

Tabu Search (TS) and Particle Swarm Optimization (PSO) have been shown to offer faster performance compared to other methods in terms of runtime and iteration speed. Among these, TS is generally the fastest, followed by PSO and GA. However, ACO, despite delivering high-quality solutions, often has the longest runtime and the highest standard deviation in its results (Raheem & Shabat, 2024). Additionally, in job-shop scheduling problems, Mathematical Programming has been found to outperform Constraint

Programming in terms of both solution quality and computational efficiency (Sakib et al., 2025).

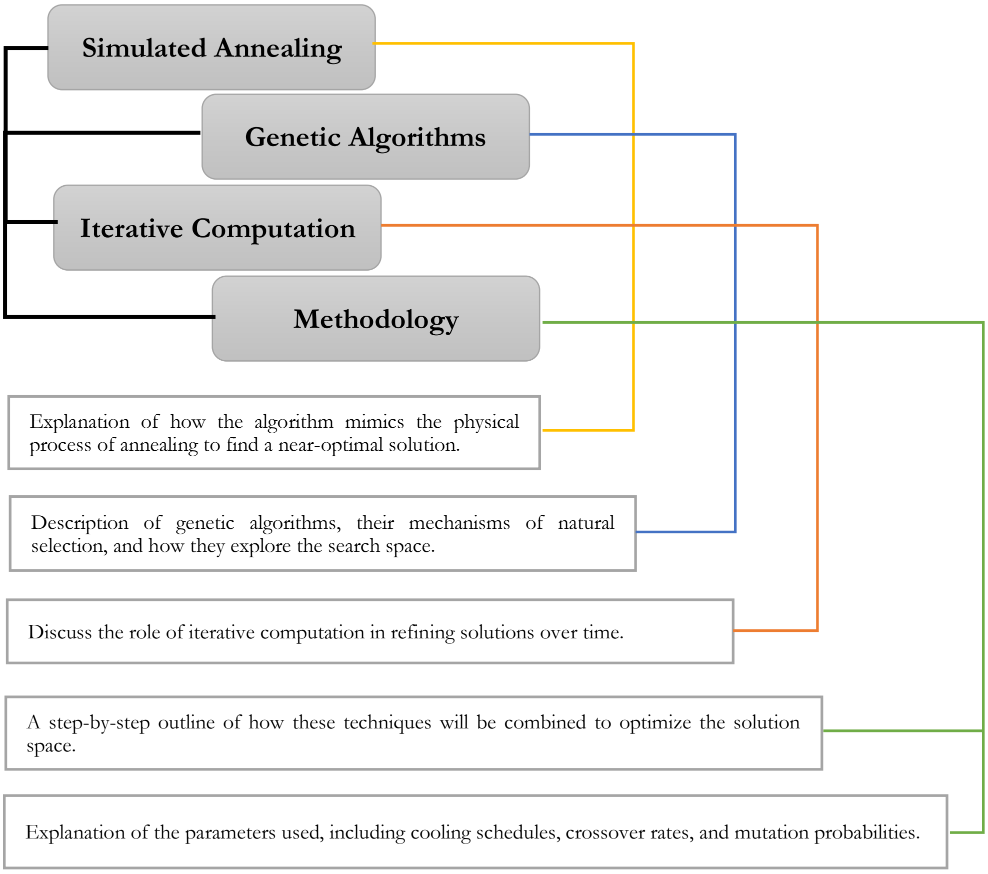
Key Findings

Classical methods are ideal for smaller or medium-sized problems where accuracy is paramount. These methods tend to provide high-quality results but require significant computational resources, especially for larger problems (Kuma et al., 2024). In contrast, heuristic methods offer flexibility, enabling them to handle larger, more complex problems more efficiently. They are particularly useful when exact solutions are impractical due to computational constraints. However, their performance is often dependent on specific parameters and the nature of the problem instance (Raheem & Shabat, 2024; Sakib et al., 2025).

Hybrid approaches, which combine multiple heuristics, can capitalize on the strengths of each method to potentially improve both solution quality and computational efficiency. However, these approaches can also lead to increased complexity and longer runtimes, which may be a disadvantage in real-time or resource-constrained applications (Sakib et al., 2025).

3. Materials and Method

This research will combine three heuristic techniques-Simulated Annealing (SA), Genetic Algorithms (GA), and Iterative Computation-to optimize the solution space to combinatorial optimization problems, such as the Traveling Salesman Problem (TSP). SA will explore the solution space by accepting a worse solution initially to avoid local optimization, GA will develop solution populations through selection, cross-breeding, and mutation, and Iterative Computation such as Tabu Search or Particle Swarm Optimization (PSO) will improve solutions through iterative change. This process will begin with the generation of a random solution population, followed by the implementation of SA and GA, and finally refinement through iterative methods. Hybridization between SA and GA will be used to combine the strengths of both methods, albeit with increased computational complexity. The parameters used include the cooling schedule for SA, the crossover rate and mutation probability for GA, as well as the length of the taboo list for Tabu Search.



Figur 1. Research Methodology Flowchart image structure.

Simulated Annealing (SA)

Simulated Annealing (SA) is a stochastic optimization algorithm inspired by the heating and cooling process of materials in metallurgical processes. This algorithm explores the solution space by allowing the worse solution to be accepted with a certain probability to avoid being stuck on local optima. Over time, the probability of acceptance of these worse solutions will decrease, allowing the algorithm to converge towards an almost optimal solution. The cooling process used in SA plays a crucial role in the algorithm's performance and its ability to find optimal global solutions.

Genetic Algorithms (GA)

Genetic Algorithms (GA) is an optimization method inspired by the principles of natural selection and genetics. This algorithm starts with a solution population that grows over time through a process of selection, crossing, and mutation. Selection prioritizes the better-performing individuals, crosses combine parts of the two solutions to create new offspring, and mutations introduce small changes to the solution to explore a wider solution space. This evolutionary process allows GA to find optimal or near-optimal solutions through new generations.

Iterative Computation

Iterative Computation methods, such as Tabu Search and Particle Swarm Optimization (PSO), play a role in iteratively improving solutions by making small changes to the solution over time. In Taboo Search, the algorithm repeatedly fixes the solution by exploring neighboring solutions and avoiding previously visited solutions using a memory structure called a "taboo list". PSOs, on the other hand, are inspired by the social behavior of birds or fish, in which solutions (particles) move through the solution space based on their own experiences and those of their neighbors. This method is balanced in terms of exploration (looking for new areas) and exploitation (improving existing areas), which makes it particularly suitable for large-scale combinatorial optimization problems.

Methodology

In this study, a combination of Simulated Annealing (SA), Genetic Algorithms (GA), and Iterative Computation will be used to optimize the solution space for combinatorial optimization problems, such as the Traveling Salesman Problem (TSP). This combined methodology will harness the strengths of each algorithm, where SA will help escape from local optima, GA will explore a wide space of solutions, and the Iterative Computation method will improve the solution over time.

This methodology will be carried out with the following steps: (a) *Early Population Generation:* Initial solution populations will be randomly generated for both GA and SA. (b) *Proses Simulated Annealing:* SA will start by exploring the solution space and gradually accept worse solutions with a certain probability (determined by the cooling schedule). The cooling schedule will reduce the probability of receiving a worse solution over time to ensure convergence towards a near-optimal solution. (c) *Pelaksanaan Genetic Algorithm:* GA will develop a population of solutions using selection, crossing, and mutation. The rate of crossing will be regulated to control how much genetic material is exchanged between parents, and the probability of mutation will determine how often small changes are introduced in the offspring. (d) *Iterative Refinement:* Tabu Search or PSO will be used to refine solutions by exploring neighboring solutions and balancing exploration with exploitation. The best solutions found by SA and GA will be used as a starting point for further refinement through iterative computing. (e) *Hybridization:* Results from SA and GA will be combined, with solutions from SA being used as the initial population for GA. Hybridization aims to leverage the strengths of both methods to improve the overall quality of the solution, although this may increase computational complexity.

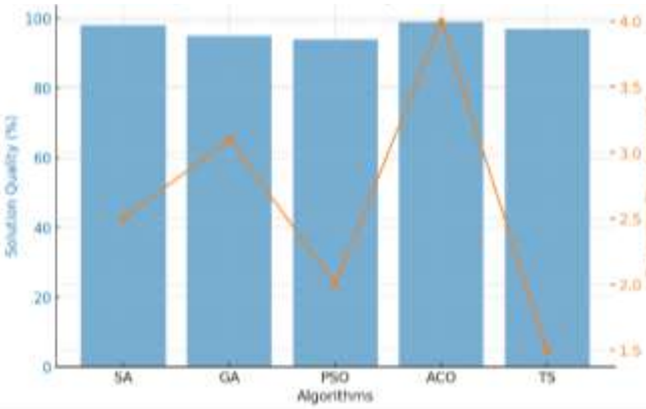
Parameters Used

(a) *Cooling Schedule (*for SA*)*: The cooling schedule will determine the rate of decline in the probability of acceptance of a worse solution. This schedule will be carefully adjusted to balance exploration and convergence. (b) *Cross Rate (*for GA*)*: This parameter controls how much genetic material is exchanged between parent solutions. A moderate level of crossover will be used to ensure diversity while maintaining the quality of the solution. (c) *Mutation*

*Probability (*for GA*)*: This parameter determines how often random changes are introduced in the offspring. A low probability of mutation will be selected to maintain a good solution while allowing for occasional exploration. (d) *Taboo List Length (*for Taboo Search*)*: The length of the taboo list will be set to control the memory of previously visited solutions and avoid cycles.

4. Results and Discussion

The results showed that Simulated Annealing (SA) and Genetic Algorithms (GA) provided an almost optimal solution to the Traveling Salesman Problem (TSP) problem with good solution quality, although SA is more sensitive to the cooling schedule and GA requires adjustment of parameters such as crossover rate and mutation probability. Particle Swarm Optimization (PSO) provides a competitive solution with a longer execution time, while Ant Colony Optimization (ACO) provides the best solution but with the longest execution time. Taboo Search (TS) shows the fastest performance, albeit with a slightly lower quality solution. Hybridization of SA and GA can improve the quality of solutions, but with increased computational complexity. The choice of the best algorithm depends on the balance between the quality of the solution and the efficiency of the computation, as well as the specific needs of the application.



Figur 2. Comparison of Heuristic Algorithms for TSP.

Results

The results obtained from the application of heuristic techniques show variations in performance in terms of solution quality and computational efficiency. Simulated Annealing (SA) provides a near-optimal solution, but its performance is highly dependent on setting the right cooling schedule. Genetic Algorithms (GA) also provide good and consistent solutions to large problem spaces, although their performance is greatly affected by the level of crossover and mutation probability. Tabu Search (TS) and Particle Swarm Optimization (PSO) offer faster performance, with PSO delivering competitive results in solution quality but with slower execution times than TS. Ant Colony Optimization (ACO), while producing a more accurate solution, requires longer execution times and shows high variability in the results.

Overall, SA and GA show excellent ability to solve combinatorial optimization problems, although both require precise parameter settings. ACOs demonstrate superior solution quality, but with longer execution times. The use of hybrid methods, such as a combination of SA and GA, shows the potential to improve the quality of solutions, although increasing computational complexity that takes longer.

Discussion

The results obtained support the effectiveness of the heuristic method in solving complex combinatorial optimization problems. Simulated Annealing (SA) works well in exploring the solution space and avoiding getting stuck on local optima, which is a major challenge in problems like the Traveling Salesman Problem (TSP). However, its performance is highly dependent on the parameters of the cooling schedule, which affects its ability to converge on the solution almost optimally. This suggests that although SA is effective, the parameters used in the process must be carefully selected to obtain optimal results.

Genetic Algorithms (GA), although slightly slower compared to other methods, show excellent ability to develop solutions through selection, crossing, and mutation. GA can explore a large solution space and find high-quality solutions, although its performance is also highly dependent on crossover rate and mutation probability settings. The use of GA provides an advantage in dealing with very large and complex problems, although there is a need to balance exploration and exploitation in its search process.

The use of Ant Colony Optimization (ACO), although it provides a very accurate solution, requires a considerable execution time. ACOs exhibit superior performance in optimization issues that require finding the best solution, however, in the presence of high execution time variations and increased complexity, ACOs are better suited for use in cases where solution quality is paramount. The addition of hybridizations such as SA and GA improves the quality of the solution, but also adds to the complexity of the computation, suggesting that the use of a combination of heuristic techniques requires a compromise between the quality of the solution and the optimal efficiency of execution time.

5. Comparison

Heuristic methods like Simulated Annealing (SA), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO) provide a flexible approach to solving combinatorial optimization problems, particularly for larger problem spaces where classical methods like Branch-and-Bound and Dynamic Programming struggle due to their high computational cost. While classical methods offer high accuracy and are ideal for smaller problems, they become impractical for larger ones. Heuristic methods balance between solution quality and execution time, with ACO providing the best solutions but at the highest computational cost, and Tabu Search (TS) offering the fastest execution times with a slightly lower solution quality. The hybridization of methods, such as combining SA and GA, can improve solution quality, but it often increases computational complexity.

6. Conclusion

The study demonstrates that heuristic methods, including Simulated Annealing (SA), Genetic Algorithms (GA), and Particle Swarm Optimization (PSO), are highly effective in solving combinatorial optimization problems like the Traveling Salesman Problem (TSP). While classical methods such as Branch-and-Bound and Dynamic Programming provide exact solutions with high accuracy, they become computationally impractical for larger problem sizes. Heuristic methods, on the other hand, offer a balanced trade-off between solution quality and computational efficiency, with ACO delivering the best solutions at the cost of longer runtimes, and Tabu Search (TS) being the fastest method with slightly lower solution quality. The combination of SA and GA in hybrid approaches has shown potential for improving solution quality, though it leads to increased complexity.

Heuristic techniques are particularly suitable when dealing with large combinatorial problems where classical methods are computationally expensive or impractical. They are ideal for scenarios where approximate solutions within a reasonable timeframe are more valuable than achieving exact results. Hybrid approaches combining different heuristics, such as SA+GA, should be considered for further improvement in solution quality, though care should be taken to manage the additional complexity and computational cost. It is recommended that practitioners select the appropriate heuristic method based on the problem size, required solution quality, and available computational resources.

Future research could explore the testing of additional heuristic techniques, such as Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO), on a wider range of combinatorial problems. Investigating the application of deep learning methods integrated with heuristic algorithms could offer new avenues for improving solution efficiency and quality. Additionally, further hybridization of multiple heuristic techniques, as well as tuning their parameters, holds potential for optimizing performance on even larger and more complex combinatorial optimization problems.

References

Asani, E. O., Ayebga, P. O., Ayoola, J. A., Okeyinka, A. E., & Adebiyi, A. A. (2019). A preliminary study on the complexity of some heuristics for solving combinatorial optimization problems. *International Journal of Engineering Research and Technology, 12*(10), 1615-1620. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074606084&partnerID=40&md5=034de6c4cc8361ba306766b277ee9e2b

Bao, L. N. L., Le, D. H., & Nguyen, D. A. (2018). Application of combinatorial optimization in logistics. *Proceedings 2018 4th International Conference on Green Technology and Sustainable Development, GTSD 2018*, art. no. 8595447, 329-334. https://doi.org/10.1109/GTSD.2018.8595447

Bojnordi, M. N., & Ipek, E. (2017). Memristive Boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning. *2017 5th Berkeley Symposium on Energy Efficient Electronic Systems, E3S 2017 - Proceedings*, 2018-January, 1-3. https://doi.org/10.1109/E3S.2017.8246178

Cappart, Q., Moisan, T., Rousseau, L.-M., Prémont-Schwarz, I., & Cire, A. A. (2021). Combining reinforcement learning and constraint programming for combinatorial optimization. *35th AAAI Conference on Artificial Intelligence, AAAI 2021*, 5A, 3677-3687. https://doi.org/10.1609/aaai.v35i5.16484

Castañedalozano, R., & Schulte, C. (2020). Survey on combinatorial register allocation and instruction scheduling. *ACM Computing Surveys, 52*(3), art. no. 62. https://doi.org/10.1145/3200920

Chen, J., & Nurdin, H. I. (2019). Generalized simulated annealing with sequentially modified cost function for combinatorial optimization problems. *2019 Australian and New Zealand Control Conference, ANZCC 2019*, art. no. 8945670, 243-248. https://doi.org/10.1109/ANZCC47194.2019.8945670

Duan, S., Jiang, S., Dai, H., Wang, L., & He, Z. (2023). The applications of hybrid approach combining exact method and evolutionary algorithm in combinatorial optimization. *Journal of Computational Design and Engineering, 10*(3), 934-946. https://doi.org/10.1093/jcde/qwad029

Gusti Agung Premananda, I., Tjahyanto, A., & Muklason, A. (2024). Timetabling problems and the effort toward generic algorithms: A comprehensive survey. *IEEE Access, 12*, 143854-143868. https://doi.org/10.1109/ACCESS.2024.3463721

Juan, A. A., Chica, M., De Armas, J., & Kelton, W. D. (2016). Simheuristics: A method of first resort for solving real-life combinatorial optimization problems. *OR58: The OR Society Annual Conference*, 147-156. https://www.scopus.com/inward/record.uri?eid=2-s2.0-85006857119&partnerID=40&md5=8a68c59a16123ee45975169a364fdd9e

Kuma, Y., Sahoo, R. Ch., & Dixit, P. (2024). A computational analysis of optimization techniques to solve assignment problem. *2024 2nd International Conference on Advances in Computation, Communication and Information Technology, ICAICCIT 2024*, 649-656. https://doi.org/10.1109/ICAICCIT64383.2024.10912276

Mbarek, F., & Mosorov, V. (2019). Load balancing based on optimization algorithms: An overview. *Journal of Telecommunications and Information Technology, (4)*, 3-12. https://doi.org/10.26636/jtit.2019.131819

Miftakhov, E. N. (2025). Software implementation of heuristic methods of optimization and integration into a cloud service. In *Lecture Notes in Electrical Engineering, 1324 LNEE* (pp. 176-185). https://doi.org/10.1007/978-3-031-82494-4_17

Ouassam, E., Hmina, N., Bouikhalene, B., & Hachimi, H. (2021). Heuristic methods: Application to complex systems. *2021 International Conference on Optimization and Applications, ICOA 2021*, art. no. 9442647. https://doi.org/10.1109/ICOA51614.2021.9442647

Purkayastha, R., Chakraborty, T., Saha, A., & Mukhopadhyay, D. (2020). Study and analysis of various heuristic algorithms for solving travelling salesman problem-a survey. *Advances in Intelligent Systems and Computing, 1112*, 61-70. https://doi.org/10.1007/978-981-15-2188-1_5

Raheem, K. R., & Shabat, H. A. (2024). Performance assessment of three swarm intelligence algorithms in combinatorial problem. *Ingenierie des Systemes d'Information, 29*(6), 2161-2167. https://doi.org/10.18280/isi.290606

Raj, A., Kumar, A., Sharma, V., Rani, S., Shanu, A. K., & Singh, T. (2023). Applications of genetic algorithm with integrated machine learning. *Proceedings of 2023 3rd International Conference on Innovative Practices in Technology and Management, ICIPTM 2023*. https://doi.org/10.1109/ICIPTM57143.2023.10118328

Sakib, F. F., Joy, J. S., Juel, S. I., & Hasan, M. M. (2025). Applying heuristic algorithms for solving the 0-1 knapsack problem: An experimental analysis. *International Conference on Robotics, Electrical and Signal Processing Techniques*, 356-360. https://doi.org/10.1109/ICREST63960.2025.10914434

Sakib, F., Rayied, S. H., Sarkar, R., Mahadi, M. H., & Hasan, M. M. (2024). Evaluating heuristic approaches for solving the 0/1 knapsack and MMKP: A comparative study. *2024 27th International Conference on Computer and Information Technology, ICCIT 2024 - Proceedings*, 523-528. https://doi.org/10.1109/ICCIT64611.2024.11022084

Stracquadanio, G., & Pardalos, P. M. (2019). Stochastic methods for global optimization and problem solving. In *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* (pp. 321-327). https://doi.org/10.1016/B978-0-12-809633-8.20329-4

Tao, P., & Chen, L. (2025). Combinatorial optimization: From deep learning to large language models. *Science China Mathematics*. https://doi.org/10.1007/s11425-023-2364-2

Yaqoob, A., Verma, N. K., & Aziz, R. M. (2024). Metaheuristic algorithms and their applications in different fields: A comprehensive review. In *Metaheuristics for Machine Learning: Algorithms and Applications* (pp. 1-35). https://doi.org/10.1002/9781394233953.ch1