c

Contents lists available at *Dergipark*

Journal of Scientific Reports-B

journal homepage: *https://dergipark.org.tr/en/pub/jsrb*

E-ISSN: 2717-8625 ***Number 11, December 2024***

RESEARCH ARTICLE

Receive Date: 17.11.2023 *Accepted Date: 04.11.2024*

A Novel Hybrid Approach for Solving the Traveling Salesman Problem: Combining Local Search Techniques for Enhanced Performance

Charmarke Houssein Abdi[a], Hasan Temurtaş[b], Safa Dörterler[c]*, Durmuş Özdemir[d]

[a]*Kütahya Dumlupınar University, Faculty of Engineering, Department of Computer Engineering, 43000, Kütahya, Turkey,*
ORCID:0009-0003-6057-1577

[b]*Kütahya Dumlupınar University, Faculty of Engineering, Department of Computer Engineering, 43000, Kütahya, Turkey,*
ORCID:0000-0001-6738-3024

[c]*Kütahya Dumlupınar University, Faculty of Engineering, Department of Computer Engineering, 43000, Kütahya, Turkey,*
ORCID:0000-0001-8778-081X

[d]*Kütahya Dumlupınar University, Faculty of Engineering, Department of Computer Engineering, 43000, Kütahya, Turkey,*
ORCID:0000-0002-9543-4078

Abstract

The Traveling Salesman Problem (TSP) is a classic combinatorial optimization problem. It involves finding the most efficient route that visits a set of cities exactly once and returns to the starting point. The development of an efficient solution to this problem is of great practical importance, particularly in the context of logistical and transportation applications. Some of the classic local search methods that have been adopted in the quest for better solutions include 2-Opt, 3-Opt, Slide, and Swap. These methods generate neighboring solutions in a systematic manner, eliminating

* Corresponding author. Tel.: +90-274-443-4267; fax: +90-274-265-2066.
E-mail address: safa.dorterler@dpu.edu.tr
http://dx.doi.org/10.1016/j.cviu.2017.00.000

.

suboptimal routes and thus improving the quality of the solutions. Among these, the 2-Opt method involves the elimination of crossed edges in the route. In contrast, the 3-Opt extends this concept to more complex changes, and while it may have the potential to generate superior solutions, it does so at a higher computational cost. The aim of this study is to provide a comprehensive investigation of the performance of the four methods: 2-Opt, 3-Opt, Slide, and Swap. Additionally, this paper proposes a hybrid method, HLSA, which incorporates all four methods in a systematic and balanced manner: 30% 2-Opt, 30% 3-Opt, 20% Slide, and 20% Swap. This approach is designed to yield more optimized results. The results demonstrate that the HLSA is significantly faster and more effective than traditional algorithms, as evidenced by rigorous experimentation and comparison. Furthermore, the solution to TSP has been shown to be both practical and efficient, making it a viable candidate for real-world implementation.

Keywords: Traveling Salesman Problem, local search, Swap, Slide, 2-Opt, 3-Opt

1. Introduction

The TSP has been one of the most famous combinatorial optimization problems that has attracted the interest of the scientific community over the last few decades. The ultimate task of the problem is to find the minimum route that a salesman should travel in order to pass through several previously determined cities and then return once again to the current city. Moreover, each city can be visited only once, further increasing the difficulty of this problem. Because of the rapid growth in the number of possible combinations that a solution can take due to an increase in the number of cities, TSP, being a computationally intensive problem, makes the determination of an optimal solution cumbersome [1].

Exact algorithms and heuristics have tackled the TSP. Exact algorithms can provide only the optimal solution for small-size problems, while for large problem sizes, they become impracticable. Therefore, heuristic methods have become highly important in practice to quickly find good approximate solutions [2].

Some of the most used techniques for solving TSP include local search algorithms. In such algorithms, an initial solution is first obtained, which is then iteratively improved. The underlying idea is to get closer to a better solution by checking the neighbors of the solution in hand. This usually aims at the minimization of the total travel distance by changing the order or location of cities [3].

Combinatorial optimization problems are elementary mathematical building blocks that result in complexity and optimization concerns in many real-world areas of application [4]. Within these, one could mention TSP. The TSP is a problem wherein a salesman seeks to carry out a tour so that he is able to visit a certain number of cities at the shortest distance possible. Several heuristic methods and optimization algorithms have been developed within the literature for finding the solutions of such problems. This paper will discuss some key local search optimization techniques: 2-Opt, 3-Opt, Slide, and Swap.

2-Opt aims to better the TSP solution by exchanging positions in two cities in pursuit of a better route. The underlying basic principle of this method is to minimize the total distance traveled by changing the intersection point of two roads [1]. 3-Opt is an enhanced version providing a number of opportunities to look for the optimal path due to three different breakpoints within the tour [5]. This method requires much more problem calculations, yet it might lead to obtaining better results [3]. The Swap operator generates a new tour by exchanging positions of two cities [6]. The Slide operator attempts to produce a new tour by sliding one of the cities into a new position while keeping intact the order of the cities in the current solution [6].

Local search operators like Slide, Swap, 2-Opt, and 3-Opt have effectively dealt with the TSP. Nevertheless, these approaches may have limitations in certain situations. Hybrid approaches that combine various optimization tactics have proven to render better outcomes [7], [8], [9], [10], [11], [12]. For instance, 2-Opt and 3-Opt type local search methods can identify an exact tour alteration and provide near-optimal solutions. These local search methods can then be followed by Slide and Swap, which increase the area of search and enhance the results [6].

This paper proposes the Hybrid Local Search Algorithm (HLSA) algorithm, including different multi-local searches such as Slide, Swap, 2-Opt, and 3-Opt in solving TSP. This algorithm employs a dynamic weighting scheme, with the assignment of percentage weight to each local search method: 20% for Swap, 20% for Slide, 30% for 2-Opt, and 30% for 3-Opt, and then uses the roulette wheel for the selection of operators for every function call. In this methodology, the performance of the solution for TSP will be improved.

This paper consists of six chapters. Section 2 gives the overall review related to the TSP optimization and related concepts, identifying the key methodologies and approaches from earlier works. Section 3 describes the TSP problem in depth, entailing a description of the operators used, dataset selection, and details about the HLSA algorithm. Section 4 depicts the experimental results, considering HLSA against other optimization methods. Section 5 discusses the results in detail. Finally, Section 6 concludes the study by summarizing the essential findings and contributions of HLSA and highlighting the effectiveness of HLSA in delivering better solutions with lower computational costs. Furthermore, some of the possible avenues for future research are outlined, which may improve the algorithm's performance.

2. Related works

Deep learning and machine learning are widely used in solving various problems, such as optimization problems, data classification, object recognition, and natural language processing, among many others. These technologies have been applied in a wide range of applications, from post-pandemic education continuity [13] to hygiene monitoring systems [14] and waste management solutions [15]. Moreover, effective results have been achieved in complex tasks such as biological image classification [16], license plate recognition [17], transfer learning [18], and sign language recognition [19]. Successful results have also been obtained with AI-based models in medical tasks such as brain tumor classification [20].In particular, deep learning and AI-based methods have succeeded significantly in combinatorial optimization problems such as TSP. TSP is one of the basic problems of combinatorial optimization and is generally used to reach a solution to logistics, transportation, and planning problems. Within this context, in this section, we draw upon some critical works on TSP in the literature and introduce the different solution methods.

Khan and Maiti proposed an enhanced version of the ABC algorithm for tackling with TSP problems, exploiting some update rules along with K-opt operations [21]. All eight different update rules proposed in this work are done in a way that these update rules may help the swap operation on the sequence of cities in order to improve the quality of the solution paths. Bees update the solution in the algorithm using a randomized rule selection process via the roulette wheel selection process. At the same time, some stagnant solutions undergo K-opt operations to further improve the best solution quality. The algorithm's performance was evaluated on TSPLIB benchmark datasets, demonstrating satisfactory accuracy and consistency when compared to existing methods for symmetric and asymmetric TSP problems.

Zhao et al. proposed a hybrid approach that encompasses concepts of the simulated annealing method and local search in solving the TSP [12]. A valid tour is considered to be one that visits each city exactly once and then returns to the original city. The algorithm was enhanced by the addition of a simulated annealing component as well as a localized search technique. The computational tests performed on datasets that had never undergone any preprocessing returned promising results, confirming the effectiveness of this hybrid method.

Tongur and Ülker enhanced the MBO algorithm by adding neighborhood operators to obtain more effective solutions for TSP and TSP and asymmetric TSP (ATSP) problems [22]. While the basic Migrating Birds Optimization (MBO) algorithm was shown to be quite effective for solving quadratic assignment types of problems, in other more complex problems such as TSP and ATSP, the performance remains far from optimal. The experiment showed that the neighborhood operators had a very significant impact, up to 36%, on improving the performance of the algorithm. Experimental results from TSPLIB datasets confirmed that the modified MBO algorithm successfully solved TSP problems when optimized with different neighborhood operators.

Voudouris et al. consider the TSP to be one of the most famous problems in the field of combinatorial optimization [23]. They investigate how the techniques of Guided Local Search (GLS) and Fast Local Search (FLS) can be applied to such a problem. GLS builds on local search heuristics and aims to enable these processes to explore large search spaces of combinatorial optimization problems efficiently and effectively. GLS can be combined with the neighborhood reduction scheme of FLS, significantly speeding up the algorithm's operations. Combining GLS and FLS with TSP local search heuristics of different efficiency and effectiveness is investigated to determine the dependence of GLS on the local search heuristics used. Furthermore, comparisons are made with some of the best TSP heuristic algorithms and general optimization techniques, showing the advantages of GLS over alternative heuristic approaches proposed for the problem.

Gouveia and Paias et al. consider TSP with spatial consistency constraints [24]. Spatial consistency constraints mean that the location of the next visit point constrains a visit point. These constraints can arise, for example, when a healthcare provider, while visiting a patient, also wants to visit other patients in the same region [24]. The paper presents an algorithm for solving TSP with spatial consistency constraints. The algorithm is based on the tabu search algorithm and uses a set of tabu rules to find a route that satisfies the spatial consistency constraints.

Günay-Sezer and Çakmak et al. proposed a method to optimize drone movements using a hybrid meta-heuristic solution [11]. This solution combines the capabilities of genetic algorithm (GA) and tabu search (TS) algorithms. GA generates and refines travel routes, while TS facilitates continuous exploration without getting stuck in local optima. The proposed hybrid meta-heuristic solution outperforms classical methods in terms of solution quality.

Mosayebi and Sodhi, as described in their article, studied the complexities of the business time traveling salesman problem (TSPJ), in which a preset amount of time should be allotted to each customer [25]. Unlike the classic TSP, TSPJ necessitates optimizing travel routes and time spent with each customer. The method suggested in the article resulted in better outcomes than traditional methods.

Shi and Zhang conducted research on various solution techniques for TSP using artificial neural networks (ANN) [26]. The authors objectively found that ANN has significant potential for solving TSP. Their proposed method demonstrated superiority over classical methods and effectively solved TSP using ANNs.

Voudouris and colleagues conducted a study on the effectiveness of GLS and FLS techniques for TSP[23]. Gouveia and Paias introduced a heuristic algorithm for TSP that includes spatial consistency constraints [24]. Günay-Sezer and Çakmak proposed a hybrid meta-heuristic solution method involving drones and reported superior outcomes [11]. Mosayebi and Sodhi solve the TSPJ problem with customer lead times and reach a more skilled solution than what was provided by classical techniques [25]. Shi and Zhang applied ANNs to solve the TSP problem and showed that this approach can often ascertain better solutions than those obtained by using traditional techniques [26]. Both these papers indicate that optimization methods can be applied to different applications which at its core involves TSP and its variants. In the end, the literature review concerning the Traveling Salesman Problem demonstrates that extensive research has led to the development of numerous algorithms and techniques for solving this significant optimization problem.

A wide range of methods in literature has been applied to solve TSP and related problems. Local search algorithms were found particularly useful for tackling TSP [27], while the hybridization of techniques has been used in finding better solutions to combinatorial problems [28].

Lancia and Dalpasso introduced one new cubic-time 3-Opt algorithm to solve TSP, which yields substantial improvements in finding an optimum against traditional techniques [29]. They designed one new algorithm that turns out to be faster and more efficient for TSP challenges and returned an effective performance about the optimization of TSP.

Schmidt and Irnich introduce novel neighborhoods and iterated local search (ILS) algorithms to handle the generalized traveling salesman problem (GTSP), which is an extension of TSP involving an associated cost function for each customer [30]. The authors' approach surpasses conventional ILS methods on multiple GTSP test problems.

Gao et al. suggest utilizing a Chaotic Differential Evolution algorithm for optimization problems [31]. By integrating a chaotic local search component into the classical differential evolution algorithm (DE), Chaotic DE enhances the algorithm's capability to evade local optima and maintain the search process, resulting in an effective optimization tool that has yielded successful outcomes with these methods.

Aly and Guadagni et al. demonstrated how local search algorithms can optimize ANNs [32]. By eliminating the need for derivative calculations, these algorithms effectively reduce the complexity of ANNs. The authors proved the efficiency of local search algorithms in reducing ANN complexity.

Various methods and approaches have been proposed for the solution of TSP and related problems. In summary, Lancia and Dalpasso proposed a 3-Opt motion detection algorithm for TSP, which is cubic-time efficient and achieves expedited outcomes [29]. Schmidt and Irnich proposed a new ILS algorithm for GTSP, resulting in superior results [30]. Furthermore, Gao et al. had shown that chaotic local search-based differential evolution algorithms constitute an effective method to deal with optimization problems [31]. Aly and Guadagni applied ANN optimization using local search algorithms to bring out the effectiveness of the approach [32]. These research contributions provided new techniques for solving TSP and similar problems, further enriching the field.

There are many diverse practical areas of application of TSP and other combinatorial optimization problems. Their solution ensures a reduction in cost and provides more productivity in the business sector, which is a great advantage. However, the intricate nature of these problems renders the application of traditional methods an arduous and ineffective process. A review of the literature reveals that hybrid techniques can yield superior outcomes compared to traditional methods. Consequently, hybrid approaches have the potential to effectively address TSP and related problems, and their applicability can be extended to more complex problems in the future.

This paper proposes a hybrid local search algorithm that can be regarded as an incorporating approach of several local searches for the solution of TSP optimization problems. In HLSA, a dynamic weighting mechanism is applied to distribute certain percentages of every local search method. This mechanism enables the algorithm to track the adaptation of an instance in any problem and selects the most effective technique at each iteration. The roulette wheel method determines the local calling operator for each function call. The primary objective of this method is to enhance the solution efficiency of TSP and other related combinatorial optimization problems.

3. Materials and Methods

3.1. Travelling Salesman Problem

The Traveling Salesman Problem is one of the most traditional problems in combinatorial optimization, which falls under the main fields of computer science and operational research. It is a type of problem in which a salesman is tasked with visiting a given number of cities exactly once and returning to the starting city (or a given starting point) while minimizing the total distance traveled. If cities are represented by nodes and roads by lines, the problem corresponds to finding the minimum cost closed path on the graph [33]. As the number of nodes in the problem increases, the time spent solving the problem increases exponentially; the exponential increase of the solution time in TSP with an increasing number of nodes is shown in Table 1 [33].

Table 1. Evaluation of Hamiltonian Cycles.

Number of Nodes	Number of Cycles (n-1)!	Time
12	39.916.800	0.004 second
13	479.001.600	0.04 second
14	6.227.020.800	1 second
15	87.178.291.200	9 second
16	1.307.647.368.000	2 second
17	(17-1)!	35 second

18	(18-1)!	10 hours
19	(19-1)!	7,5 days
20	(20-1)!	140 days
21	(21-1)!	7,5 years
22	(22-1)!	160 years
23	(23-1)!	3.500 year
24	(24-1)!	2 million years

When Table 1 is analyzed, it is seen that the number of cycles increases rapidly with the number of nodes, and the time required to complete this process increases dramatically. The computation times become quite long, especially when increases beyond 15 nodes are considered. This shows that Hamiltonian loops and similar problems become more challenging to solve as the number of nodes increases. The mathematical formulation of a TSP is given below.

Constraints:

$$\sum_{j=1, j \neq i}^{n} x(i,j) = 1, i = 1,2,\ldots,n \tag{1}$$

$$\sum_{i=1, i \neq j}^{n} x(i,j) = 1, j = 1,2,\ldots,n \tag{2}$$

$$\sum_{i,j \epsilon S}^{n} x(i,j) \leq |S| - 1, \forall\, S \subset \{1,2,\ldots,n\} \tag{3}$$

$$x(i,j) = \begin{cases} 1, & \textit{if going from point i to point j} \\ 0, & \textit{if going from point i to point j} \end{cases} \tag{4}$$

Here, the distance between points i and j is denoted as d(i,j), while x(i,j) represents the distance from point i to point j. Equations 1 and 2 ensure that each point is visited only once. Equation 1 ensures that each point is exited only once, while equation 2 guarantees that each point is visited only once. Equation 3 serves as a sub-round elimination constraint, eliminating any sub-rounds that may occur. In Equation 4, a value of 1 for x(i,j) indicates that a trip has been made from point i to point j, while a value of 0 indicates that no trip has been made.

Objective Function:

$$\sum_{i=1}^{n} \sum_{j=1, j \neq i}^{n} x(i,j).d(i,j) \tag{5}$$

The objective function of the TSP, also presented in Equation 5, minimizes the tour length by finding the shortest possible route that visits each city exactly once and returns to the origin city.

3.2. Local search

Local search is a strategy for finding a better solution to an optimization problem by making small changes around an existing solution [34]. Slide, Swap, 2-Opt, and 3-Opt are standard methods to perform these modifications [35]. This section will provide an explanation of the Swap, Slide, 2-Opt, and 3-Opt techniques.

3.2.1. Swap operator

The swap operator is a technique to solve the TSP problem by randomly selecting two cities and swapping their positions within the tour sequence. This action modifies the sequence of visits. Repeatedly using this operator allows exploration of various shorter tour configurations [36]. By adopting this approach, conflicts at edges are resolved, leading to the identification of promising solutions and, ultimately, a decrease in tour length. Although not guaranteeing the optimal solution, the swap operator remains a valuable tool for efficiently enhancing existing solutions [36]. Figure 1 illustrates how the swap operator operates.

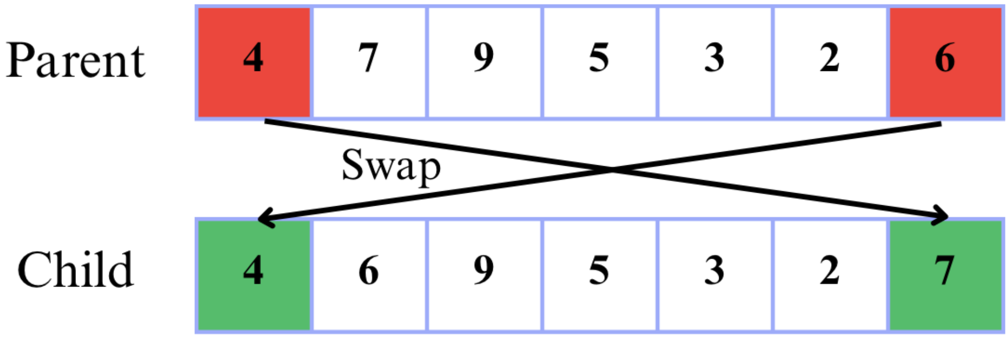


Fig. 1. The swap operator working principle.

3.2.2. Slide operator

The Slide is a fundamental operator used in TSP solutions. The Slide Operator moves a city to a new location without breaking the order of other cities. This maneuver explores shorter routes and prevents conflicts between edges to boost the tour's overall length [36]. The Slide Operator offers an alternative to the Swap Operator, allowing for examining new potential solutions by making slight changes to the arrangement of cities in the tour. This aids in enhancing the quality of the TSP solution by eliminating conflicts. However, like the Swap Operator, the Slide Operator does not ensure the best solution [37]. Figure 2 illustrates the working logic of the Slide Operator.

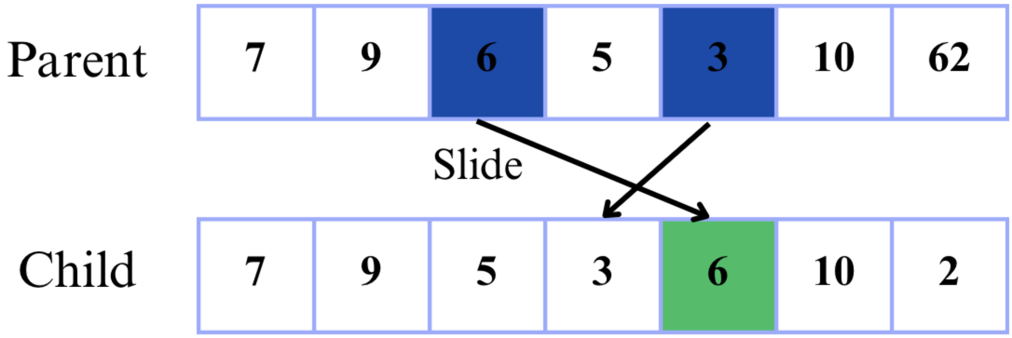


Fig. 2. The Slide operator working principle.

3.2.3. 2-Opt operator

The 2-Opt operator is utilized in TSP problem-solving methods to enhance solution quality by minimizing the total length of the corresponding tour. Unlike Swap or Slide operators, 2-Opt concentrates on eliminating diagonal edges present in the tour [38]. The 2-Opt operation identifies two such diagonal edges in the current TSP round. Once identified, these two edges are removed, which split the tour into two parts. The process of eliminating diagonals involves combining the two sections and reversing the order of city visits between the edges [39]. This procedure is repeated until no further improvements can be achieved. Although the 2-Opt technique is not infallible, it provides a systematic method to remove cross edges. The operational logic of this technique is depicted in Figure 3.

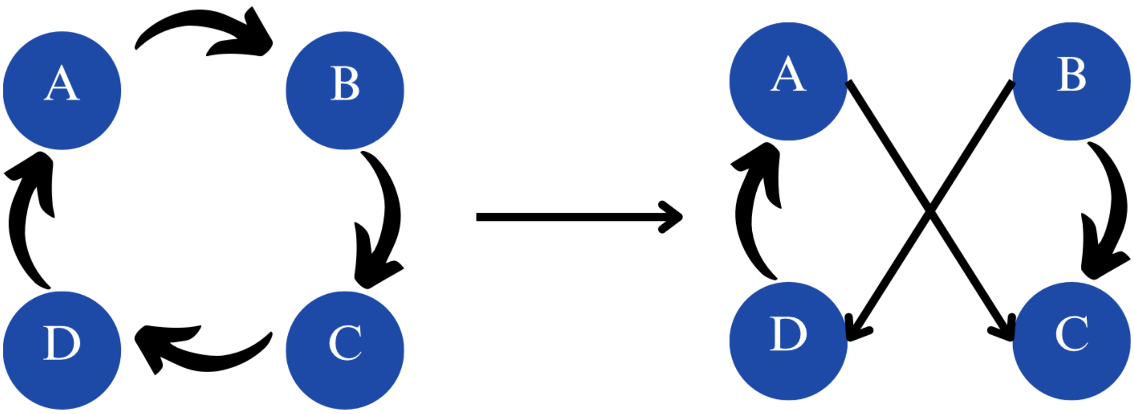


Fig. 3. The 2-Opt operator working principle.

3.2.4. 3-Opt operator

The 3-Opt operator is a method utilized for TSP problem-solving. The 3-Opt technique entails identifying three diagonal edges within the ongoing TSP round. By eradicating these three edges, the round is separated into multiple parts. The order of visiting the cities between these edges is reorganized to remove the diagonals, and the segments are then combined to create a novel tour. To explore various tour configurations, the same procedure is repeated multiple times [40]. It explores more tour possibilities than 2-Opt, but it does not guarantee the best solution of the TSP like other optimization processes. However, it is an essential tool used to improve the solution quality of TSP in the context of local search and optimization. Cheng et al. showed the working logic of the 3-Opt technique in their work [41].

3.3. Data set

The dataset for the study, which was retrieved from Traveling Salesman Problem Library (TSPLIB) [42], [43]. Introduction TSPLIB is a library of sample instances for the TSP (and related problems) from various sources and of varying size, which have been compiled over several years. TSPLIB instances have city coordinates, distance information between cities, and other necessary info to model TSP, which researchers employ for evaluating numerous algorithms/heuristics of TSP in comparison [42]. The data obtained from the TSPLIB library and the number of cities are presented in Table 2.

Table 2. Cities used in the study.

Dataset	Eil76	Berlin52	Rat99	St70	KroA100	Lin105	KroA200
# of regions	76	52	99	70	100	105	200

3.4. Proposed model

Various optimization operators are used to solve TSP problem [21]. The Slide operator moves one city without disturbing the order of the others and allows to find shorter routes. Swap tries to optimize the total route length by swapping two cities within the tour. 2-Opt reverses the order of cities when removing two diagonal edges in the tour, while 3-Opt does the same to remove three diagonal edges. These optimizations are used to improve SP's existing solutions quickly, but they do not guarantee finding the best solution. After testing all four methods, we created a hybrid method, HLSA, which combines all four methods (Slide, Swap, 2-Opt, 3-Opt). The distribution of 20% for Swap, 20% for Slide, 30% for 2-Opt, and 30% for 3-Opt was selected after a series of preliminary experiments. These proportions were determined based on the performance of each operator during these trials. The 2-Opt and 3-Opt operators, being more computationally intensive, were assigned higher weights as they consistently yielded superior results for larger datasets. On the other hand, the Swap and Slide operators, which execute faster but offer incremental improvements, were assigned equal yet lower proportions to maintain diversity in search space exploration. Such a balance enables the algorithm to explore global and local optima effectively, preventing it from incurring high computational costs. A more uniform distribution, such as 25% for each operator, was tested but yielded suboptimal results, as the algorithm could not fully leverage the strengths of the 2-Opt and 3-Opt techniques. The roulette wheel method determines the algorithm to choose, and what is its likeness. Roulette wheel method: This is a simple and effective probability selection algorithm that chooses from among many items or options [44], [45], [46]. This is used mostly in optimization problems. Its simple idea is to design a system where the chance of picking each option will be directly proportional to some quality or criteria amongst all options.

A random number between 0 and 1 is generated at the beginning of our algorithm. Based on the value of this number, the following selection process occurs:

- If the number is between 0 and 0.2, the Swap operator is selected.
- If the number is between 0.2 and 0.4, the Slide operator is selected.
- If the number is between 0.4 and 0.7, the 2-Opt operator is selected.
- If the number is between 0.7 and 1, the 3-Opt operator is selected.

Then, the distance is calculated and saved. Then, the next iteration is started, and the above operations are repeated. If the new solution is better than the one stored in memory, the new solution is saved; otherwise, the new solution is forgotten. This way, the process continues until the end state is reached and the shortest path is found. The structure of our HLSA model is given in Fig. 4.

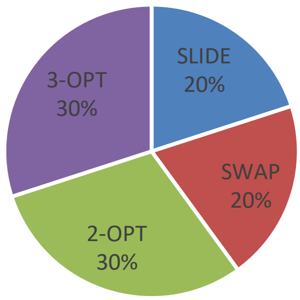


Fig. 4. Structure of the HLSA model.

To make the functioning of the HLSA algorithm clearer, the pseudo-code of the HLSA algorithm is given below.

pseudo-code of the HLSA algorithm
Initialize: - Define dataset from TSPLIB. - Set initial route and memory solution. While stopping criteria is not met: - Generate a random number between 0 and 1. - If 0 ≤ random number < 0.2, apply Swap operator. - If 0.2 ≤ random number < 0.4, apply Slide operator. - If 0.4 ≤ random number < 0.7, apply 2-Opt operator. - If 0.7 ≤ random number ≤ 1, apply 3-Opt operator. - Evaluate new route. - If new route is better than current memory solution, update memory. End While: - Return the best route found.

4. Results

This study combines the local search algorithm operators to solve the TSP, and the HLSA algorithm is created. The proposed HLSA model was run in MATLAB version R2023a on the Windows 11 operating system and a computer with Intel i5, 2.5 GHz processor, and 8 GB RAM. Table 3 shows the results of Slide, Swap, 2-Opt, 3-Opt and HLSA methods on eight different datasets. All algorithms were run under the same conditions and on the same datasets. The results in the table are measured in path length, which is the total length of the route determined by each algorithm. A lower path length means that the algorithm performs better and completes the trip in a shorter distance. The shortest paths (most successful results) are shown in bold.

Table 3. Local search optimization algorithms (Swap, Slide, 2-Opt, 3-Opt, and HLSA).

Data Set	Swap	Slide	2-Opt	3-Opt	HLSA
Eil76	690.65	652.65	600.52	590.45	**571.43**
Eil51	520.85	490.35	470.95	460.92	**449.96**
Berlin52	7920.65	7880.65	7850.96	7840.80	**7821.64**
Rat99	1450.55	1420.53	1398.12	1396.24	**1388.85**
St70	825.52	795.52	760.41	762.20	**743.92**
KroA100	24190.62	24160.52	24130.20	24140.20	**24129.07**
Lin105	16420.63	16400.63	16386.96	16381.20	**16380.93**
KroA200	32312.54	32302.24	32271.85	32270.40	**32253.39**

Table 3 presents the results of the Slide, Swap, 2-Opt, 3-Opt, and HLSA methods applied to eight different datasets. When analyzing these data, it is evident that the HLSA consistently produces the most successful results across all problems. Specifically, for datasets such as Eil76, Eil51, Berlin52, Rat99, St70, KroA100, Lin105, and KroA200, the HLSA generated shorter, more cost-effective paths than the standalone methods.

These findings demonstrate that the hybrid nature of HLSA combining multiple local search strategies provides a significant advantage over applying individual operators in isolation. For instance, while techniques like Swap or Slide might offer faster iterations, they tend to get stuck in local optima, which limits their effectiveness when dealing with more complex problem instances. On the other hand, although more computationally intensive, the 2-Opt and 3-Opt methods can explore a broader solution space, leading to better overall solutions.

Importantly, our assertion is not that HLSA will necessarily beat the standalone methods in speed. Certainly, operators like Swap or Slide can converge faster on their own because they are very simple. But the real power of HLSA is not in its speed but ratherin how good it can do and use that capability to produce high-quality solutions across diverse datasets. The hybrid algorithm is that it explores (through operators such as 2-Opt and 3-Opt) but exploits (Swap, Slide), subsequently producing better paths by preventing from becoming stuck into local optima. These findings are also corroborated by the consistently lower cost values provided by HLSA relative to other techniques.

Moreover, Fig. 5 illustrates the coordinates of the optimal paths identified by the HLSA algorithm across the eight datasets, underscoring the effectiveness and reliability of the approach. Moreover, the convergence behavior of the algorithm can be observed in the cost function plots in Fig. 6, which depict the function calls versus the cost for each dataset. These plots further demonstrate that while HLSA may require more iterations to converge in some cases, the solutions it produces are consistently superior to those obtained by standalone methods.

Ultimately, HLSA's effectiveness comes from its ability to combine the strengths of different operators, leveraging the global search capabilities of 2-Opt and 3-Opt alongside the faster but more localized improvements offered by Swap and Slide. This strategic combination enhances the search process, providing more optimal solutions for many TSP instances.

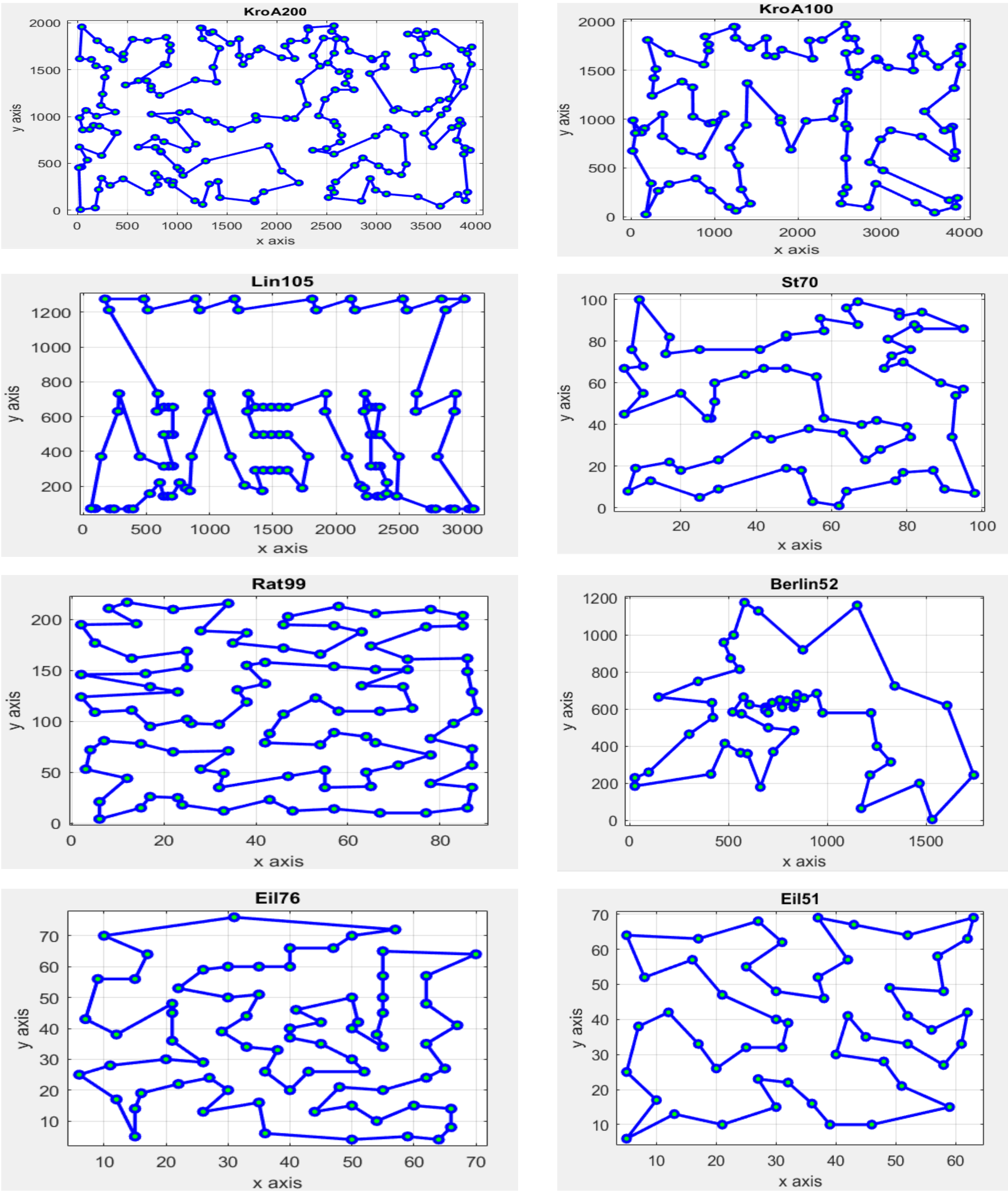
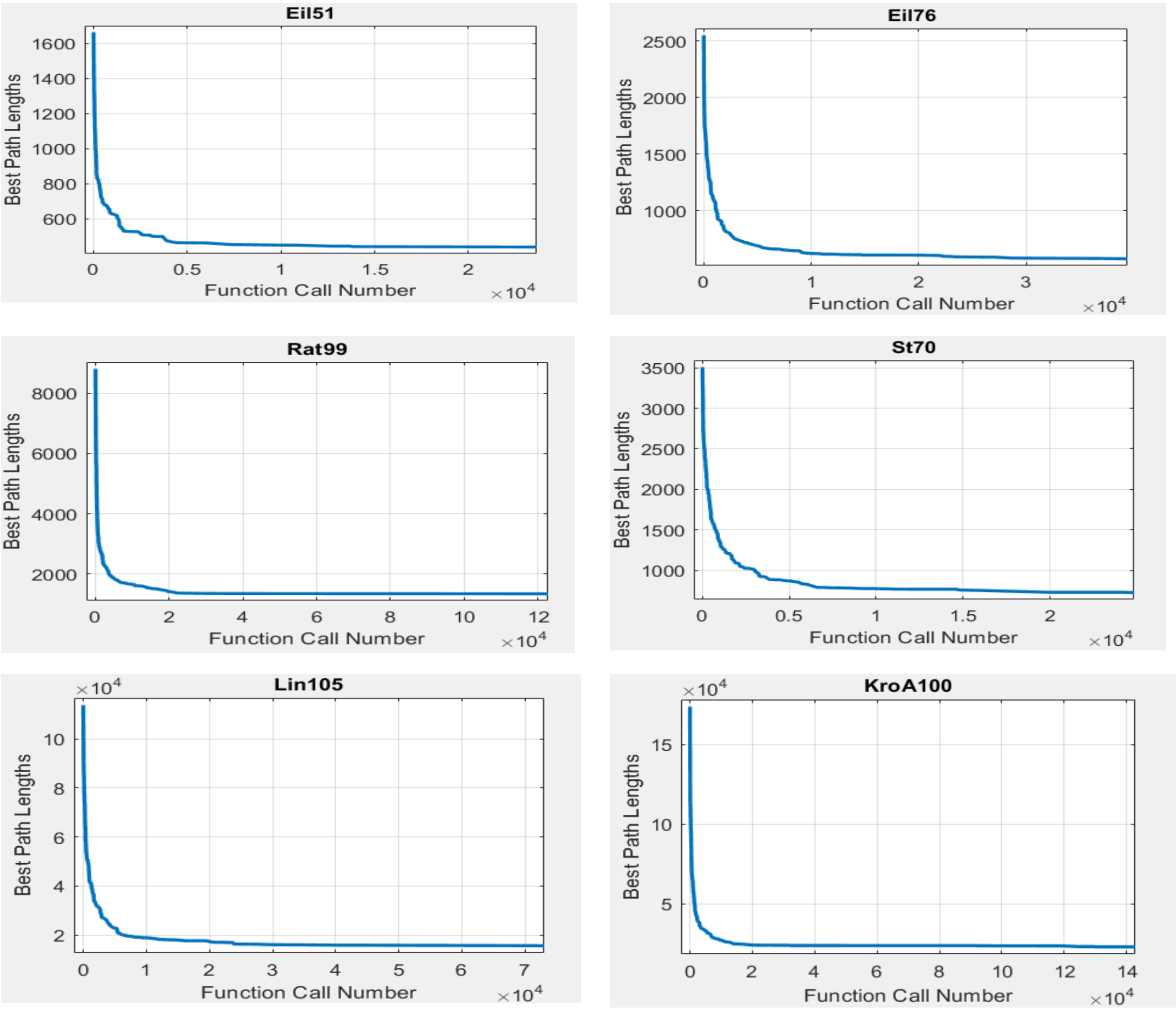


Fig. 5. The best coordinates found by HLSA.

Fig. 5 shows the best paths found by the HLSA algorithm for eight different datasets (Eil76, Eil51, Berlin52, Rat99, St70, KroA100, Lin105 and KroA200). In each panel, we can see the placement of the cities in the dataset on the shortest path and how this path was optimized. It is possible to observe from these visualizations how the HLSA algorithm provides an effective solution on different dataset. Fig. 6 presents the Cost Function plots of HLSA for eight different datasets.



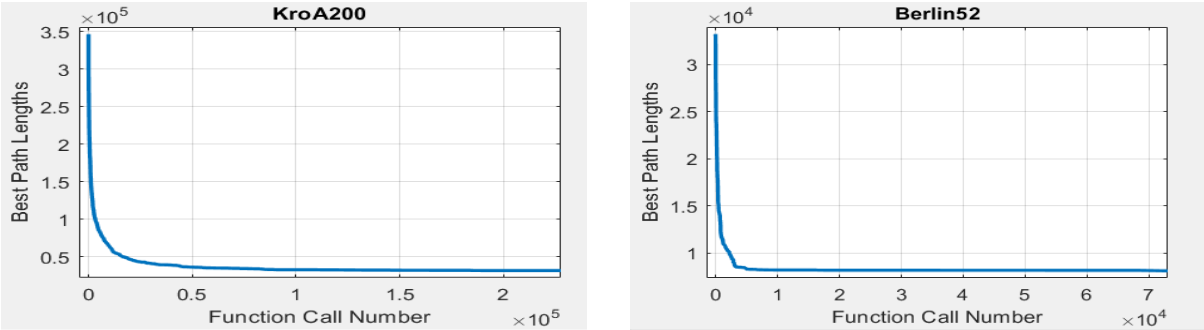


Fig. 6. Cost Function graphs of HLSA.

The performance of the HLSA algorithm was evaluated using eight TSP datasets. Analysis of the cost function graphs resulting from the algorithm operation reveals the following outcomes:

The algorithm identified the optimal solution for Eil51, Berlin52, and St70 datasets with 2000-7500 function calls. The algorithm demonstrated high performance for these datasets. For the Eil76 dataset, the algorithm determined the optimal solution with 10,000 function calls. The algorithm also performs well on this dataset. It achieved the optimal result with 20,000 to 50,000 function calls for the Rat99, Lin105, KroA100, and KroA200 datasets. However, the performance of these datasets is relatively inferior compared to the results obtained from other well-known optimization techniques such as Ant Colony Optimization (ACO), Genetic Algorithms (GA), and Artificial Bee Colony (ABC) [47], [48], [49], [50]. While these algorithms may outperform the proposed Hybrid Local Search Algorithm (HLSA) on specific large datasets, our method offers significant simplicity, computational efficiency, and hybrid adaptability advantages. By dynamically combining local search techniques (Swap, Slide, 2-Opt, and 3-Opt), HLSA balances exploration and exploitation, leading to more efficient problem-solving on medium-sized datasets with fewer computational resources. While other methods may offer better global solutions for highly complex problems, our approach is particularly suitable for real-world applications where computational time and simplicity are critical. Moreover, the hybrid nature of HLSA allows it to converge faster. It requires fewer parameters to be tuned compared to ACO, GA, or ABC, making it a more practical option in time-sensitive or resource-constrained environments.

The HLSA algorithm's performance varies with dataset size. It is concluded that as the number of cities or data points increases, the algorithm requires more computations, and the convergence process lengthens. Based on these findings, the HLSA algorithm proves highly efficient for the travel salesman problem. Nonetheless, the algorithm's performance dwindles as the size of the dataset increases.

5. Discussion

In this study, the HLSA aims to demonstrate the power of combining classical local search techniques by introducing a probabilistic structure for solving complex optimization problems, such as the TSP. The results indicate that HLSA offers more effective solutions than other methods across various datasets. However, specific key points should be emphasized to understand the findings better.

One of the primary advantages of HLSA is its ability to reduce the risk of being trapped in local optima through probabilistic operator selection. This enables the algorithm to explore a broader solution space, balancing local and global search capabilities. Particularly in larger datasets, such as KroA200, this probabilistic structure enhances the algorithm's capacity to find more effective solutions where other methods fall short. Such probabilistic approaches have been widely supported in the literature. Gu and Huang have demonstrated that combining various local search techniques increases the chances of reaching the global optimum [27]. Similarly, Kaabachi et al. highlight the effectiveness of hybrid methods in solving combinatorial problems like the TSP [51].

In the literature, there are metaheuristic methods that outperform others in solving TSP, such as Simulated Annealing, Genetic Algorithms, and Tabu Search, which may yield shorter path lengths for specific datasets. However, these methods' complexity and long computation times, especially for large datasets, present a significant disadvantage. In contrast, HLSA combines simpler and faster local search techniques, such as Swap, Slide, 2-Opt, and 3-Opt, into a hybrid structure, resulting in lower computational costs and faster solutions. This makes it more applicable compared to complex methods, offering an effective alternative, particularly for small and medium-sized problems. A number of hybrid approaches have been proposed that make use of local search techniques and it has been demonstrated by researchers that these methods can produce good results even on large datasets [52], [53], [54].

The probabilistic nature of HLSA helps reduce the risk of getting stuck in local optima while combining global and local search capabilities. This flexibility allows the algorithm to explore broader solution spaces where other methods are confined to a single search strategy. In the literature, methods like GRASP (Greedy Randomized Adaptive Search Procedures) have demonstrated how such diversity can improve solution quality [55].

Metaheuristic algorithms often require very long execution times, especially for large data sets, and the computational cost may put a burden that we can hardly afford them. HLSA takes advantage of the simplicity that classical local search operators provide and in turn provides faster results with a lower cost. For example, Ant Colony and Genetic Algorithms achieve good results with large datasets but they are also known for heavy computational usage [47], [48]. The method of using a HLSA, in contrast, provides an approach that is more reasonable and still removes most of these costs.

The purpose of this work is to show that the combination with local search methods in a probabilistic hybrid framework leads to better results. Using the hybrid techniques can improve and come up with a solution that is of good quality at relatively lower computational cost. To sum up, by pooling the resources of classical local search methods in a hybrid manner HLSA serves as providing lower computational costs and more give head for solution strategies. The results from this study confirm that HLSA is a new way of solving the TSP problem. The use of hybrid search models is a competitive and strong option for optimization issues as TSP. Future research could involve further enhancements and optimization strategies to improve the performance of HLSA.

6. Conclusion

This This paper presents evidence supporting the efficacy of the proposed HLSA for solving the TSP. The hybrid approach, which combines four distinct local search methods (Swap, Slide, 2-Opt, and 3-Opt), yielded shorter paths and lower-cost outcomes in experiments conducted on disparate datasets. The efficacy of HLSA is evidenced by its superior performance compared to other methods, particularly for medium-sized problems. Furthermore, the test results on different datasets demonstrated that HLSA can explore a vast solution space and circumvent the local optima trap.

One of the most significant advantages of the algorithm is its ability to strike a balance between global and local search capabilities by combining different local search methods. The methodology uses the efficiency of more detailed and slower approaches, like 2-Opt and 3-Opt, with faster ones such as swap and slide, so this concept represents a hybrid structure. Notably, despite the increase in computational time for large datasets, the solution quality offered by HLSA is quite satisfactory in comparison to more costly and time-consuming algorithms. In conclusion, this research demonstrates the potential of hybrid approaches to combinatorial optimization problems and proves that HLSA offers an innovative solution to challenging problems such as TSP.

For future work, it is recommended that HLSA be combined with new optimization techniques to improve its performance, especially on larger datasets and more complex problems. Moreover, a more advanced adaptive structure, in which the weights of different local search methods are dynamically adjusted, can further improve the efficiency of the algorithm.

Author contribution

C.H.A., H.T., S.D, and D. Ö. actively participated in conducting the experimental studies and writing the manuscript

Acknowledgements

The authors have no individuals or organizations to acknowledge.

References

[1] M. Englert, H. Röglin, and B. Vöcking, "Worst case and probabilistic analysis of the 2-Opt algorithm for the TSP," *Algorithmica*, vol. 68, no. 1, pp. 190–264, 2014.
[2] A. H. Halim and Ija. Ismail, "Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem," *Archives of Computational Methods in Engineering*, vol. 26, pp. 367–380, 2019.
[3] G. F. Hertono and B. D. Handari, "The modification of hybrid method of ant colony optimization, particle swarm optimization and 3-OPT algorithm in traveling salesman problem," in *Journal of Physics: Conference Series*, IOP Publishing, 2018, p. 012032.
[4] X. Yang *et al.*, "A review: machine learning for combinatorial optimization problems in energy areas," *Algorithms*, vol. 15, no. 6, p. 205, 2022.
[5] M. Mahi, Ö. K. Baykan, and H. Kodaz, "A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem," *Appl Soft Comput*, vol. 30, pp. 484–490, 2015.
[6] S. Singh and E. A. Lodhi, "Study of variation in TSP using genetic algorithm and its operator comparison," *International Journal of Soft Computing and Engineering*, vol. 3, no. 2, pp. 264–267, 2013.
[7] N. Yagmur, I. Dag, and H. Temurtas, "A new computer-aided diagnostic method for classifying anaemia disease: Hybrid use of Tree Bagger and metaheuristics," *Expert Syst*, p. e13528, 2023.
[8] N. Yagmur, İ. Dag, and H. Temurtas, "Classification of anemia using Harris hawks optimization method and multivariate adaptive regression spline," *Neural Comput Appl*, pp. 1–20, 2024.
[9] S. Dörterler, H. Dumlu, D. Özdemir, and H. Temurtaş, "Hybridization of Meta-heuristic Algorithms with K-Means for Clustering Analysis: Case of Medical Datasets," *Gazi Mühendislik Bilimleri Dergisi*, pp. 1–23.
[10] S. Dörterler, H. Dumlu, D. Özdemir, and H. Temurtaş, "Hybridization of K-means and Meta-Heuristics Algorithms for Heart Disease Diagnosis," in *NEW TRENDS IN ENGINEERING AND APPLIED NATURAL SCIENCES*, Duvar Publishing, 2022, p. 55.
[11] N. S. Gunay-Sezer, E. Cakmak, and S. Bulkan, "A hybrid metaheuristic solution method to traveling salesman problem with drone," *Systems*, vol. 11, no. 5, p. 259, 2023.
[12] D. Zhao, W. Xiong, and Z. Shu, "Simulated annealing with a hybrid local search for solving the traveling salesman problem," *J Comput Theor Nanosci*, vol. 12, no. 7, pp. 1165–1169, 2015.
[13] F. Aydemir and S. Arslan, "A System Design With Deep Learning and IoT to Ensure Education Continuity for Post-COVID," *IEEE Transactions on Consumer Electronics*, 2023.
[14] F. Aydemir and S. Arslan, "Covid-19 pandemi sürecinde çocukların el yıkama alışkanlığının nesnelerin interneti tabanlı sistem ile izlenmesi," *Mühendislik Bilimleri ve Araştırmaları Dergisi*, vol. 3, no. 2, pp. 161–168, 2021.
[15] V. Kaya, "Classification of waste materials with a smart garbage system for sustainable development: a novel model," *Front Environ Sci*, vol. 11, p. 1228732, 2023.
[16] C. Arslan and V. Kaya, "Classification of Plant Species from Microscopic Plant Cell Images Using Machine Learning Methods," *International Research Journal of Engineering and Technology (IRJET)*, vol. 11, no. 05, pp. 853–861, May 2024.
[17] G. Arslan, F. Aydemir, and S. Arslan, "Enhanced license plate recognition using deep learning and block-based approach," *Journal of Scientific Reports-A*, no. 058, pp. 57–82, 2023.
[18] V. Kaya, "A perspective on transfer learning in computer vision," 1st ed., vol. 1, Platanus, 2023, ch. 17, pp. 332–359.
[19] N. N. Arslan, E. Şahin, and M. Akçay, "Deep learning-based isolated sign language recognition: a novel approach to tackling communication barriers for individuals with hearing impairments," *Journal of Scientific Reports-A*, no. 055, pp. 50–59, 2023.
[20] E. Şahin, D. Özdemir, and H. Temurtaş, "Multi-objective optimization of ViT architecture for efficient brain tumor classification," *Biomed Signal Process Control*, vol. 91, p. 105938, 2024.
[21] I. Khan and M. K. Maiti, "A swap sequence based artificial bee colony algorithm for traveling salesman problem," *Swarm Evol Comput*, vol. 44, pp. 428–438, 2019.
[22] V. Tongur and E. Ülker, "The analysis of migrating birds optimization algorithm with neighborhood operator on traveling salesman problem," in *Intelligent and Evolutionary Systems: The 19th Asia Pacific Symposium, IES 2015, Bangkok, Thailand, November 2015, Proceedings*, Springer, 2016, pp. 227–237.
[23] C. Voudouris and E. Tsang, "Guided local search and its application to the traveling salesman problem," *Eur J Oper Res*, vol. 113, no. 2, pp. 469–499, 1999.
[24] L. Gouveia, A. Paias, and M. Ponte, "The travelling salesman problem with positional consistency constraints: An Application to healthcare services," *Eur J Oper Res*, vol. 308, no. 3, pp. 960–989, 2023.
[25] M. Mosayebi, M. Sodhi, and T. A. Wettergren, "The traveling salesman problem with job-times (tspj)," *Comput Oper Res*, vol. 129, p. 105226,

2021.
[26] Y. Shi and Y. Zhang, "The neural network methods for solving Traveling Salesman Problem," *Procedia Comput Sci*, vol. 199, pp. 681–686, 2022.
[27] J. Gu and X. Huang, "Efficient local search with search space smoothing: A case study of the traveling salesman problem (TSP)," *IEEE Trans Syst Man Cybern*, vol. 24, no. 5, pp. 728–735, 1994.
[28] K. Panwar and K. Deep, "Discrete Grey Wolf Optimizer for symmetric travelling salesman problem," *Appl Soft Comput*, vol. 105, p. 107298, 2021.
[29] G. Lancia and M. Dalpasso, "Finding the best 3-OPT move in subcubic time," *Algorithms*, vol. 13, no. 11, p. 306, 2020.
[30] J. Schmidt and S. Irnich, "New neighborhoods and an iterated local search algorithm for the generalized traveling salesman problem," *EURO Journal on Computational Optimization*, vol. 10, p. 100029, 2022.
[31] S. Gao, Y. Yu, Y. Wang, J. Wang, J. Cheng, and M. Zhou, "Chaotic local search-based differential evolution algorithms for optimization," *IEEE Trans Syst Man Cybern Syst*, vol. 51, no. 6, pp. 3954–3967, 2019.
[32] A. Aly, G. Guadagni, and J. B. Dugan, "Derivative-free optimization of neural networks using local search," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, IEEE, 2019, pp. 293–299.
[33] Y. Dumas, J. Desrosiers, E. Gelinas, and M. M. Solomon, "An optimal algorithm for the traveling salesman problem with time windows," *Oper Res*, vol. 43, no. 2, pp. 367–371, 1995.
[34] J. Gu and X. Huang, "Efficient local search with search space smoothing: A case study of the traveling salesman problem (TSP)," *IEEE Trans Syst Man Cybern*, vol. 24, no. 5, pp. 728–735, 1994.
[35] L. Sengupta, R. Mariescu-Istodor, and P. Fränti, "Which local search operator works best for the open-loop TSP?," *Applied Sciences*, vol. 9, no. 19, p. 3985, 2019.
[36] P. Singamsetty, J. Thenepalle, and B. Uruturu, "Solving open travelling salesman subset-tour problem through a hybrid genetic algorithm," *Journal of Project Management*, vol. 6, no. 4, pp. 209–222, 2021.
[37] R. Hossain, S. Magierowski, and G. G. Messier, "GPU enhanced path finding for an unmanned aerial vehicle," in *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*, IEEE, 2014, pp. 1285–1293.
[38] Y. Harrath, A. F. Salman, A. Alqaddoumi, H. Hasan, and A. Radhi, "A novel hybrid approach for solving the multiple traveling salesmen problem," *Arab J Basic Appl Sci*, vol. 26, no. 1, pp. 103–112, 2019.
[39] I. Mavroidis, I. Papaefstathiou, and D. Pnevmatikatos, "Hardware implementation of 2-opt local search algorithm for the traveling salesman problem," in *18th IEEE/IFIP International Workshop on Rapid System Prototyping (RSP'07)*, IEEE, 2007, pp. 41–47.
[40] A. F. Tuani, E. Keedwell, and M. Collett, "Heterogenous adaptive ant colony optimization with 3-opt local search for the travelling salesman problem," *Appl Soft Comput*, vol. 97, p. 106720, 2020.
[41] B. Cheng, H. Lu, Y. Huang, and K. Xu, "An improved particle swarm optimization algorithm based on Cauchy operator and 3-opt for TSP," in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, IEEE, 2016, pp. 177–182.
[42] G. Reinelt, "Tsplib95," *Interdisziplinäres Zentrum für Wissenschaftliches Rechnen (IWR), Heidelberg*, vol. 338, pp. 1–16, 1995.
[43] G. Reinelt, "TSPLIB—A traveling salesman problem library," *ORSA journal on computing*, vol. 3, no. 4, pp. 376–384, 1991.
[44] D. Özdemir and S. Dörterler, "An adaptive search equation-based artificial bee colony algorithm fortransportation energy demand forecasting," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 4, pp. 1251–1268, 2022.
[45] D. Özdemir, S. Dörterler, and D. Aydın, "A new modified artificial bee colony algorithm for energy demand forecasting problem," *Neural Comput Appl*, vol. 34, no. 20, pp. 17455–17471, 2022.
[46] O. Çıtlak, M. Dörterler, and İ. Dogru, "A hybrid spam detection framework for social networks," *Politeknik Dergisi*, vol. 26, no. 2, pp. 823–837, 2022.
[47] N. M. Razali and J. Geraghty, "Genetic algorithm performance with different selection strategies in solving TSP," in *Proceedings of the world congress on engineering*, International Association of Engineers Hong Kong, China, 2011, pp. 1–6.
[48] J. Yang, X. Shi, M. Marchese, and Y. Liang, "An ant colony optimization method for generalized TSP problem," *Progress in natural science*, vol. 18, no. 11, pp. 1417–1422, 2008.
[49] L. Li, Y. Cheng, L. Tan, and B. Niu, "A discrete artificial bee colony algorithm for TSP problem," in *Bio-Inspired Computing and Applications: 7th International Conference on Intelligent Computing, ICIC 2011, Zhengzhou, China, August 11-14. 2011, Revised Selected Papers 7*, Springer, 2012, pp. 566–573.
[50] S. P. Tiwari, S. Kumar, and K. K. Bansal, "A survey of metaheuristic algorithms for travelling salesman problem," *International Journal Of Engineering Research & Management Technology*, vol. 1, no. 5, 2014.
[51] I. Kaabachi, D. Jriji, and S. Krichen, "A DSS based on hybrid ant colony optimization algorithm for the TSP," in *Artificial Intelligence and Soft Computing: 16th International Conference, ICAISC 2017, Zakopane, Poland, June 11-15, 2017, Proceedings, Part II 16*, Springer, 2017, pp. 645–654.
[52] V. Kelner, F. Capitanescu, O. Léonard, and L. Wehenkel, "A hybrid optimization technique coupling an evolutionary and a local search algorithm," *J Comput Appl Math*, vol. 215, no. 2, pp. 448–456, 2008.
[53] A. Sharifi, J. K. Kordestani, M. Mahdaviani, and M. R. Meybodi, "A novel hybrid adaptive collaborative approach based on particle swarm optimization and local search for dynamic optimization problems," *Appl Soft Comput*, vol. 32, pp. 432–448, 2015.
[54] G. D'Angelo and F. Palmieri, "GGA: A modified genetic algorithm with gradient-based local search for solving constrained optimization problems," *Inf Sci (N Y)*, vol. 547, pp. 136–162, 2021.
[55] M. G. C. Resende and C. C. Ribeiro, "Greedy randomized adaptive search procedures: Advances, hybridizations, and applications," *Handbook of metaheuristics*, pp. 283–319, 2010.