Journal of Telecommunication, Electronic and Computer Engineering
ISSN: 2180 – 1843 e-ISSN: 2289-8131 Vol. 18 No. 1 (2026) 35-44
jtec.utem.edu.my
DOI: https://doi.org/10.54554/jtec.2026.18.01.005

Scalable Metaheuristic Optimization of Asymmetric and Clustered TSP Variants using Iterated Local Search

Farid bin Morsidi
Universiti Pendidikan Sultan Idris, Faculty of Computing and Meta-Technology, Computing Department, Tanjong Malim, 35900, Perak Darul Ridzuan, Malaysia.

Article Info

Article history:
Received July 9th, 2025
Revised Mar 10th, 2026
Accepted Aug 29th, 2026
Published Mar 31st, 2026

Index Terms:
Iterated local search
Routing scheduling
Distribution network
Traveling Salesman Problem

Abstract

Iterated Local Search (ILS) is a well-established metaheuristic that has been widely applied to combinatorial optimization problems owing to its balance between solution diversification and intensification. This work emphasizes ILS for the scope of the Asymmetrical Traveling Salesman Problem (ATSP) and the Asymmetrical Generalized Traveling Salesman Problem (AGTSP), which account for directionally dependent route costs and clustered node structures encountered in real-world routing applications. The study also assesses the ability of ILS to explore difficult search regions while maintaining solution quality and avoiding rapid convergence. This study examines the feasibility of ILS using TSPLIB benchmark instances subject to directed route costs, time windows and load capacity constraints, alongside an examination of a systematic conversion of symmetric TSP instances into asymmetric representations to accurately capture directionally dependent travel costs. The results demonstrate that ILS can produce high quality solutions for both ATSP and AGTSP under conditions of increasing routing complexity Future research should focus on enhancing ILS through hybrid metaheuristic framework and evaluating its performance on large-scale logistics datasets to improve scalability and practical applicability.

*Corresponding Author: P20182001774@siswa.upsi.gov.my

I. INTRODUCTION

The Traveling Salesman Problem (TSP) is one of the most extensively studied problems in combinatorial optimization, with widespread applications in logistics, transportation, and network planning. It involves finding the shortest possible route that visits a set of cities exactly once and returns to the origin. Due to its NP-hard nature, solving large-scale TSP instances using exact algorithms is computationally infeasible, prompting the development of numerous metaheuristic approaches. Among these, Genetic Algorithms (GA), Simulated Annealing (SA), and Ant Colony Optimization (ACO) have shown significant success. However, these methods often encounter challenges such as complex parameter tuning, convergence delays, and computational intensity, especially when applied to constrained or large-scale routing problems.

Iterated Local Search (ILS) has emerged as a robust alternative that addresses many of these limitations. Unlike population-based methods such as GA or path-reinforcement techniques like ACO, ILS relies on a single-solution trajectory enhanced through a structured process of perturbation and refinement. The method promotes solution diversity through controlled perturbations and leverages efficient local search operators such as 2-opt and 3-opt to iteratively improve solution quality [1], [2], [3]. This balance between diversification and intensification allows ILS to efficiently escape local optima and approach near-global optimality with relatively low computational overhead. Moreover, ILS benefits from a modular and adaptable framework that can be hybridized with other heuristics, making it suitable for solving a wide range of real-world routing problems [8].

In this study, particular attention is given to two prominent TSP variants: the Asymmetric TSP (ATSP) and the Generalized TSP (GTSP). The ATSP considers directional travel costs represented by directed graphs with non-symmetric edge weights, conditions frequently encountered in urban logistics with one-way roads or aerial navigation influenced by wind patterns [4]. The GTSP, on the other hand, involves clustering cities into predefined groups and requires visiting only one node per cluster, a configuration applicable in zonal delivery services or multi-depot distribution planning [5], [6], [7].

Although ILS has demonstrated strong performance on standard TSP instances, there is limited research assessing its effectiveness across multiple TSP variants under a consistent evaluation framework. Most ATSP studies focus on a single variant or examine hybrid strategies without fully exploring

the adaptability of ILS to structurally complex problems such as the ATSP and GTSP. This research aims to fill that gap by providing a unified evaluation of ILS-based metaheuristics across conventional TSP, ATSP, and GTSP scenarios using standardized benchmark datasets and routing models.

The study presents three main contributions as follows: (i) a unified evaluation framework that assesses ILS performance consistently across TSP, ATSP, and GTSP under a single experimental setup; (ii) a structured conversion procedure that transforms symmetric TSP instances into asymmetric representations suitable for ATSP and GTSP modelling; and (iii) a comparative empirical analysis of ILS against multiple classical heuristics on TSPLIB benchmark instances of increasing complexity. The objectives of this study are therefore threefold: (1) to implement and evaluate ILS for both standard and generalized TSP variants; (2) to benchmark its performance against existing metaheuristic methods across varied routing instances; and (3) to assess its flexibility and scalability when applied to real-world-inspired distribution network models.

The remainder of this paper is organized as follows. Section 2 reviews relevant literature on ILS and its applications in routing problems. Section 3 describes the proposed methodology, including algorithmic design and experimental setup. Section 4 presents the results and comparative analysis. Finally, Section 5 concludes the study and outlines future research directions.

II. CURRENT WORKS

The vehicle routing problem (VRP), originated from the TSP first proposed by Flood in 1956 [5]. This problem formulation requires that all participating nodes (or cities) must visit each city exactly once before returning to their departure point [5], [8], [9]. Dantzig and Ramser expanded this formulation in 1959 by introducing multiple vehicles operating under load capacity restrictions. This work established the foundation for the modern Capacitated Vehicle Routing Problem (CVRP) solution. Many iterations stemming from CVRP [10], [11] have since been developed, incorporating routing variants such as time constraints on routes [12], [13], distance limitations [8], [14], and multiple concurrent route operations[15], [16].

When it comes to combinatorial optimization, the Vehicle Routing Problem is among the most widely studied approaches for addressing NP-hard problems [11]. The development of construction heuristic variants began after researchers introduced various solution methods for routing problems, including route merging, nearest-neighbor selection, and insertion-based methods to evaluate trade-offs among distance, load capacity, and time constraints [7], [17], [18]. Researchers have extended basic algorithms to solve advanced routing problems requiring customer-to-vehicle assignments and multi-depot node allocations together with two relocation methods for improving route performance [10], [19]. Researchers have also examined how basic TSP formulations, including symmetric and generalized instances, can be extended to VRP solutions through various metaheuristic systems such as Tabu search [17], Clarke-Wright savings method [20], [21], simulated annealing [22], and genetic algorithms [23], [24]. With the application of multi-objective problem (MOP) frameworks, such as adaptive and variable neighborhood searches, in tandem with local neighborhood strategies comprising 2-Opt, Or-Opt, inter-route relocate, or inter-route adjustments, more advanced solution strategies, including iterated local search, have been developed [10], [18].

The TSP-inspired formulations have significantly influenced further research in VRP as they demonstrate effectiveness across multiple application scenarios. A new hybrid algorithm, called the Distribution Strategy Selection and Vehicle Routing Hybrid Algorithm, is presented in a study to effectively tackle a mixed delivery network problem [25]. The proposed algorithm is constructed based on direct transshipment, milk run, and cross-docking strategies to ascertain the necessary distribution strategy and critical pathways under a heterogeneous vehicle fleet. The performance is evaluated through statistical analysis, highlighting the proficiency of the proposed heuristic in maximizing resource allocation and supporting optimal decision-making strategy.

The idea of TSP modeling was initially put forward in a research investigation that employed the Nearest Neighbor, Sequential Insertion, and Saving Matrix algorithms to solve the CVRP for transportation paths in the Grobogan district. To find the best distribution routes for the Grobogan district in Indonesia, the total distance traveled was estimated using the Saving Matrix Algorithm, Sequential Insertion Algorithm, and the Nearest Neighbor Algorithm [7]. Researchers have studied TSP variants together with VRP formulations through learning-based methods to address generalization and computational scalability issues associated with traditional heuristic approaches [26]. The study examined current developments in this area, breaking them down into sequential and end-to-end methods. To assess the efficiency among four representative learning-based optimization algorithms, the research design was divided into three parts. The findings show that combining iterated local heuristic search with learning-based optimization models can enhance sampling efficiency and generalization capacity [26]. Table 1 summarizes the core reference studies discussed in this paper related to the implementation of ILS on various TSP disciplines and formulations.

Table 1
Overview of the referenced research studies based on the implementation of ILS and related methods for scheduling and routing applications

Research	Year	Advantages	Limitation	Reference
Iterated local search for addressing green vehicle routing with workload equity for fleet optimization	2020	Algorithm execution time decreases, enabling efficient solutions for up to 200 customers	Problems arise from increased demand or vehicle demobilization	[5]
Study on variable neighborhood descent and tabu search.	2021	VND utilizes converging solutions to achieve optimal conditions, enhancing route efficiency over tabu search and improving performance	Not explicitly mentioned	[6]

		while reducing problem-solving time		
Constrained Local Search for Last-Mile Routing	2024	New methods integrate routes, with the LKH-AMZ algorithm swiftly solving large datasets stop-level constraints	Requires further analysis of driver recovery using Amazon data and interpretation of suggested concealed tours	[27]
Traveling Salesman Problem solved with genetic algorithm	2021	GA-EM enhances exploration, avoids local optima, and convergence with efficient mutation, outperforming other algorithms in accuracy	TSP datasets are complex, and large, requiring substantial computational time	[2]
Memetic algorithm using Breakout Local Search for TSP	2020	Enhances BLS, improves runtime with competitive algorithms	Metaheuristic complexity is a drawback due to high algorithmic complexity	[3]
Solving capacitated vehicle routing problem in Grobogan using optimization methods	2021	The saving matrix algorithm effectively addresses CVRP, demonstrating efficiency with a route cost of Rp. 96,849	Limited exploration of alternative algorithms, VRP variants, and supporting optimization software	[7]
Vehicle routing and scheduling for machine delivery and installation	2021	New operators resolve conflicts in requests and trips, effective solutions for complex scheduling and routing problems	Performance benchmarking indicates competitive but not leading results among 13 teams: 2nd best once, 3rd best six times, 4th best often	[28]
Tabu search optimizes product service scheduling problems	2020	Proposes tabu search to minimize travel distances and penalties; develops optimization model using historical solutions such as tabu objects	Effectiveness depends on the quality of initial solutions and iteration limits in Tabu search	[29]
Generalized Traveling Salesman Problem survey	2024	Develops models that enhance GTSP solution accuracy, improve benchmark instances, and address data deficiencies for validation	Further exploration of decomposition techniques for large-scale GTSP is needed, emphasizing hierarchical structures and dynamic factors like traffic conditions and multigraph models.	[30]
New neighborhoods enhance local search for Traveling Salesman Problem	2022	ILS design shows reasonable results; tailoring improves performance	Requires further enhancement through cooling schedule adaptation and hybrid algorithm integration for GTSP instances.	[19]

III. SOLUTION STRATEGY

A. Traveling Salesman Problem (TSP)

TSP is among the prominent heuristics often referenced as the foundation for scheduling distribution networks, particularly in locating the shortest route that visits the designated cities, where these cities are visited exactly once during the tour. This problem can be represented as a bi-directed graph $G = (V, A)$ where V represents the set of participating cities and A is the set of arcs representing possible travel paths [2], [8], [31]. Another routing variable, D representing distance and defined as the cost matrix of size $|V| \times |V|$ is used to store pairwise distances between cities, whereas the distance between two participating cities are calculated by the equation c_i and $c_i + 1$ [2]. TSP is typically defined as a symmetric instance, where the distance from city j to city i. As opposed to symmetric instances of TSP, the asymmetric TSP (ATSP) considers non-equal distances between city pairs depending on direction. In general, TSP problems are formulated with N cities, and the symmetric matrix $D = [d_{ij}]$. The goal of TSP is to determine a directed path with one traversal for each city that represents the shortest possible distance. Empirically, these objective function can be expressed as the following equation [2]:

$$T_d = \min(dis\ (c_{n-1}, c_1) + \sum_{i=1}^{n-1} dis\ (c_i + c_{i+1})) \quad (1)$$

Where T_d represents the accumulated total distance for the directed path, and the distance procedure, variable *dis* constitutes the total traversed distance among the participating cities c_i and $c_i + 1$ [2].

B. Generalized Traveling Salesman Problem (GTSP)

A modified variant of the baseline TSP, the Generalized Traveling Salesman Problem (GTSP), aims to find the shortest route that starts at the starting city and visits exactly one city from each cluster. These cities are partitioned into predefined clusters. Via this routing strategy, total distance or cost can be minimized through the selection of one representative city from participating clusters. For instance, transshipment between regions might traverse only single depot per region to maximize delivery time and operational cost.

GTSP is modelled under the assumption that travel costs are symmetrical between traversed cities. The GTSP was introduced in 1969 in the formulation of a Generalized Network Design Problem (GNDP) to be utilized in client routing with retrospect among welfare agencies and to annotate record balancing complications existed in computer

design [30]. Under hierarchical structures, GTSP provides a more precise modelling framework compared to the tree-structured programming occasionally seen in the application of TSP [32]. The fundamental applications of GTSP also necessitated partitioning into clusters based on predefined criteria.

The GTSP is categorized as GNDP, which represents a class of network design problems more complex than conventional network design problems [19], [30]. The NDP-related vertices in the network are divided into defined cluster sets, and feasibility restrictions are indicated with respect to the clusters rather than individual vertices. The generalization of this problem instance is simplified [30] and considered as an undirected, connected and weighted graph $G = (V, E)$ that is consisted of specific n of vertices for $V = \{1, 2, \ldots, n\}$ and edge set $E = \{e_1, \ldots, e_m\}$ where:

$$E \subseteq \{\{i, j\} \mid i, j \in V, \text{and } i \neq j\} \tag{2}$$

A graph for GTSP representation is an undirected, connected, and weighted graph $G = (V, E)$ with n vertices and edges E. Each edge e, represented by W, is assigned a positive integer $c(e) = c_e = c_{ij} \in R_+$ via the cost function c: $E \rightarrow R_+$ [3]. The problems can be classified as Euclidean or non-Euclidean depending on whether the triangle inequality is maintained. The vertex set V is divided into k mutually exclusive nonempty subsets, each of which contains a cluster, or subset, of vertices from G [3], [30]. The graph's edges are divided into two categories: those that link vertices within the same cluster and those that link vertices from separate clusters. To find the shortest Hamiltonian tour (with respect to visiting every cluster), the GTSP requires visiting one vertex from each cluster. The GTSP is commonly defined in two variants: the Hamiltonian tour that visits every cluster exactly once (for example, one vertex from each cluster is visited) and the Hamiltonian tour that passes through every cluster at least once. Based on previous work [30], these conditions for GTSP are viable when following constraints are satisfied:

$$C_1 \cup C_2 \cup \ldots \cup C_k = V \tag{3}$$

$$C_l \cap C_p = \emptyset, \forall\, l, p \in \{1, \ldots, k\}, \text{with } l \neq p \tag{4}$$

Where C represents the cost function for the round trip, V = total node set representing cities, k = number of vehicles in the subset, l and p are unrestricted real numbers. The cyclic tour representing the entire tour can be expressed as Equation 5 [3]:

$$W(T) = c(p_m, p_1) + \sum_{i=1}^{m-1} c\,(p_i, p_{i+1}) \tag{5}$$

C. Asymmetrical Generalized Traveling Salesman Problem (AGTSP)

The AGTSP is an extension of the Generalized Traveling Salesman Problem that involves asymmetric travel costs in traversal networks [27], [30]. The TSP requires visiting each city once and returning to the origin, whereas AGTSP replaces individual cities with groups (clusters) of cities for the purpose of establishing the minimal route while visiting one city from each group [27]. AGTSP allows travel costs between cities to differ depending on direction, reflecting factors such as route conditions, transshipment methods, capacity constraints, and access to critical nodes in the distribution network. AGTSP is relatively new compared to the rest of this class of traveling salesman classification, as it has its own complexity that bridges the symmetrical conjecture and generalized conjecture. Applications of AGTSP include combinatorial optimization and serve as building blocks for more complex systems. These applications could include large-scale distributed networks with multiple dimensions and a high number of nodes, (or instance, dense populations of cities and vehicles. AGTSP helps to define the legible operating costs more appropriately by incorporating asymmetrical trip costs. As such, the cost to travel between the two cities does not have to be equal in both directions [1]. The enhancement over the GTSP to AGTSP can be characterized by the fact that one city from each cluster must be visited, and the objective is to minimize the overall trip cost. AGTSP parameters account for asymmetric trip costs through directed graph representation, which capture asymmetric travel behavior between nodes. An example of this implementation is the capability to model various vehicle travel scenarios, such as when a vehicle is climbing a hill or descending from it. The mathematical formulation of AGTSP is defined through standard optimization constraints, including objective minimization, cluster visitations, flow conservation, sub-tour elimination, and binary decision variables. The following presents the constraint formulation imposed in the proposed solution framework.

Objective functions:

$$W(T) = c(p_m, p_1) + \sum_{i=1}^{m-1} c\,(p_i, p_{i+1}) \tag{6}$$

Constraint:

$$\sum_{j \in V_k} \sum_{i \in V} x_{ij} = 1, \qquad \forall k = 1, 2, \cdots, m \tag{7}$$

$$\sum_{i \in V_k} \sum_{j \in V} x_{ij} = 1, \qquad \forall k = 1, 2, \cdots, m \tag{8}$$

$$\sum_{j \in V} x_{ij} = \sum_{j \in V} x_{ji}, \ \forall i \in V \tag{9}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \forall S \subseteq V, |S| \geq 2 \tag{10}$$

$$x_{ij} \in \{0, 1\}, \forall i, j \in V \tag{11}$$

Algorithm 1: Tabu Search for TSP

[1] **Initialize** currentTour ← generateRandomTourTSP bestTour ← currentTour bestCost ← calculateTourCostcurrentTour tabuList ← emptyQueue() tabuListCapacity ← tabuTenure
for *iteration = 1 to maxIterations* **do**
bestNeighbor ← null bestNeighborCost ← ∞ **forall** *neighbor* ∈ *generateAllNeighborscurrentTour* **do**
move ← getMovecurrentTour, neighbor neighborCost ← calculateTourCostneighbor **if** *move* ∉ *tabuList* **or** *neighborCost* < *bestCost* **then**
neighborCost < bestNeighborCost bestNeighbor ← neighbor bestNeighborCost ← neighborCost currentTour ← bestNeighbor enqueuetabuList, getMovecurrentTour, bestNeighbor **if** *sizetabuList* > *tabuListCapacity* **then**
dequeuetabuList **if** *bestNeighborCost* < *bestCost* **then**
bestTour ← bestNeighbor bestCost ← bestNeighborCost **return** bestTour
generateAllNeighborstour neighbors ← [ ] **for** $i = 0$ ***to*** $length(tour) - 2$ **do**
$j - i + 1$ **to** $length(tour) - 1$ neighbor ← swapCitiestour, i, j appendneighbors, neighbor **return** neighbors
calculateTourCosttour cost ← 0 **for** $k = 0$ ***to*** $length(tour) - 2$ **do**
cost ← cost + distance$tour[k], tour\,k + 1]$ cost ← cost + distance$tour[length(tour) - 1], tour[0]$ **return** cost
getMovecurrentTour, neighbor **return** swapOperation

Algorithm 2: Genetic Algorithm for TSP

[1] $population$ ← generateInitialPopulationTSP, populationSize $bestTour$ ← findBestTourpopulation **for** $generation = 1$ *to* $maxGenerations$ **do**
$newPopulation$ ← [ ] **while** *sizenewPopulation* < *populationSize* **do**
$parent1$ ← selectParentpopulation $parent2$ ← selectParentpopulation **if** *random0, 1* < *crossoverRate* **then**
$child1, child2$ ← crossoverparent1, parent2 **else**
$child1$ ← $parent1$; $child2$ ← $parent2$ $child1$ ← mutatechild1, mutationRate $child2$ ← mutatechild2, mutationRate appendnewPopulation, child1 appendnewPopulation, child2
$population$ ← $newPopulation$ $currentBestTour$ ← findBestTourpopulation **if** *calculateTourCostcurrentBestTour* < *calculateTourCostbestTour* **then**
$bestTour$ ← $currentBestTour$ **return** $bestTour$

Algorithm 3: Simulated Annealing for TSP

[1] $currentTour$ ← generateRandomTourTSP $currentCost$ ← calculateTourCostcurrentTour $bestTour$ ← $currentTour$; $bestCost$ ← $currentCost$
$temperature$ ← $initialTemperature$ **for** $iteration = 1$ *to* $maxIterations$ **do**
$neighbor$ ← generateRandomNeighborcurrentTour $neighborCost$ ← calculateTourCostneighbor $deltaCost$ ← $neighborCost - currentCost$ **if** $deltaCost < 0$ ***or*** *random0,1* < $\exp(-/temperature)$ **then**
$currentTour$ ← $neighbor$ $currentCost$ ← $neighborCost$ **if** $currentCost < bestCost$ **then**
$bestTour$ ← $currentTour$ $bestCost$ ← $currentCost$
$temperature$ ← $temperature \times coolingRate$ **if** $temperature < 1e{-}10$ **then**
break return $bestTour$

Algorithm 4: First Ascent Hill Climbing for TSP

[1] $currentTour$ ← generateRandomTourTSP $currentCost$ ← calculateTourCostcurrentTour $improved$ ← **true while** $improved$ **do**
$improved$ ← **false forall** $neighbor \subset generateAllNeighborscurrentTour$ **do**
$neighborCost$ ← calculateTourCostneighbor **if** $neighborCost < currentCost$ **then**
$currentTour$ ← $neighbor$ $currentCost$ ← $neighborCost$ $improved$ ← **true break**
return $currentTour$

Algorithm 5: Steepest Ascent Hill Climbing for TSP

[1] $currentTour$ ← generateRandomTourTSP $bestTour$ ← $currentTour$ $improved$ ← **true while** $improved$ **do**
$improved$ ← **false** $bestNeighbor$ ← null; $bestNeighborCost$ ← ∞ **forall** $neighbor$ ∈ $generateAllNeighborscurrentTour$ **do**
$neighborCost$ ← calculateTourCostneighbor **if** $neighborCost < bestNeighborCost$ **then**
$bestNeighbor$ ← $neighbor$ $bestNeighborCost$ ← $neighborCost$ $improved$ ← **true if** $improved$ **then**
$currentTour$ ← $bestNeighbor$ $bestTour$ ← $currentTour$ **return** $bestTour$

Figure 1. Algorithm 1-5 constitutes the heuristic algorithm imposed in the ATSP and AGTSP routing heuristic solution strategy

D. ILS Implementation and Algorithm Design

The ILS framework applied in this study follows a four-phase iterative procedure consisting of solution initialization and local search refinement and perturbation and acceptance criteria evaluation. The structure of this system follows the ILS framework [19] but has been modified to handle asymmetric cost matrices and cluster constraints present in ATSP and AGTSP formulations.

The initial solution s_0 is created through a nearest-neighbour heuristic that constructs a tour by selecting the nearest unvisited city at each point in time. The tour construction process starts with the selection of a representative node from each cluster in AGTSP instances to guarantee that the clustering constraints are satisfied. The initial solution undergoes a 2-opt local search process to obtain a locally optimum solution s*. The 2-opt procedure uses tour edge removal to create two new edges which staff starts removing from the tour. The 3-opt operator works together with 2-opt to enhance solution quality for large instances such as ft70 and 14ft70 instances. The double-bridge move functions as the perturbation operator because it allows the system to break out of its current local optimum. The double-bridge move creates four tour segments and reconnects them in a non-sequential manner that cannot be achieved by 2-opt and 3-opt, resulting in effective search path diversification [19]. The system uses a best-improvement acceptance criterion that allows the new perturbed and locally optimized solution s' to replace the existing solution s* only if it results in a lower tour cost. The acceptance strategy does not accept inferior solutions, thereby prioritizing solution quality during the search process.

E. Symmetric-to-Asymmetric Instance Conversion

A key aspect of the experimental design is the conversion of symmetric TSPLIB instances into asymmetric representations suitable for ATSP and AGTSP evaluation. Symmetric TSP instances are characterised by a cost matrix D where $d_{ij} = d_{ji}$ for all city pairs i, j. To introduce asymmetry, each symmetric instance is transformed by applying a directional cost perturbation following the approach described in [4], whereby a randomised asymmetric offset δ_{ij} is added to one direction of each edge such that:

$$d'_{ij} = d_{ij} + \delta_{ij},\ d'_{ji} = d_{ij} \qquad (12)$$

where δ_{ij} is drawn uniformly from the range $[0, \alpha \cdot d_{ij}]$, and α is a scaling factor set to 0.3 in this study to introduce moderate directional cost variance without distorting the overall distance structure. The conversion process produces modified instances that reflect realistic asymmetric road conditions such as, one-way roadways and different fuel costs depending on the direction of travel.

The first step in this conversion process is to use a k-means clustering algorithm to create 8, 10 and 14 equal-sized clusters for small, medium and large comparative examples, respectively, from the set of converted asymmetric road networks. The AGTSP instance names for each type are given as follows: 8ftv35, 10ry48p, and 14ft70.

Table 2
Final TSPLIB instances selected with proposed algorithm execution

Capacity	ATSP Data Instance	AGTSP Data Instance
Small	ftv35	8ftv35
Medium	ry48p	10ry48p
Large	ft70	14ft70

To evaluate the reliability and consistency of results from different algorithms, each algorithm was run 20 times per instance with a tour created based on a randomized nearest-neighbour initialization for each run. Table 3(a) reports summarized statistics of the results including the best solution, the mean tour cost, and the standard deviation (SD) of the 20 runs for each combination of algorithm and instance. A small SD indicates a higher level of stability of solutions

produced by the algorithm, while a small mean indicates greater overall efficiency of the algorithm.

To test whether the performance of the different algorithms was statistically different, the 20-run distributions of results for each pair of instances were compared using the Wilcoxon signed-rank test at the α=0.05 level. The results of the test showed that all hill climbing algorithms had tour costs that were statistically significantly higher than the Tabu Search algorithm for all instances ($p < 0.05$), whereas the Genetic Search algorithm and the Simulated Annealing algorithm were statistically significantly different from one another only for the two largest instances ry48p and ft70 ($p = 0.031$ and $p = 0.018$, respectively).

In order to determine if performance differences exist between the algorithms, the Wilcoxon signed-rank test was conducted at $\alpha = 0.05$ significance level; specifically, the 20-run cost distributions for each algorithm were compared to Tabu Search as a baseline. Although there were statistically significant differences at the $\alpha = 0.05$ significance level across all three ATSP cases, none of the algorithms produced results that were statistically lower than Tabu Search. This may be indicative of the small number of benchmark cases utilized in this study.

In terms of AGTSP cases, Tabu Search and Steepest Ascent exhibited the most consistent results, as they produced the smallest standard deviation across 20 runs. On the other hand, Genetic Search exhibited the largest amount of variability among the benchmark data, especially for the largest ry48p and ft70 instances with corresponding standard deviations of 157.1 and 274.2 respectively. Based on these findings, it can be concluded that while all algorithms produce feasible solutions, Tabu Search and Steepest Ascent produce solutions with significantly greater consistency under the AGTSP cluster constraints tested in this study.

In AGTSP representation, the graph $G = (V, E)$ is an undirected, connected, weighted graph extended from the GTSP implementation [19], [33] with n vertices and edges E, where V is the total node set representing cities, S is the set representing the directed edges for possible paths between nodes, i is the initial/starting city, j is the destination city, k is the total number of partaking vehicles, m denotes the total number of customers, x represents the total operating cost/accrued cost for the entire trip, and d is the depot or warehouses traversed during the round trip.

The primary difference between GTSP and AGTSP lies in the symmetry of travel costs. In GTSP, the cost is symmetric [19], making it suitable for undirected graphs, such as a postal delivery route where all roads allow two-way travel with the same conditions. In contrast, AGTSP deals with asymmetrical costs and is modeled with directed graphs, such as a routing problem in a city with one-way streets or varying fuel costs based on direction and load. The decision regarding the type of problem model (GTSP/AGTSP) is contingent upon the features of the scheduling environment. The two problems are intricate and have practical applications in the domains of logistics, network architecture, and vehicle routing.

The Tabu Search algorithm generates an acceptable path structure between clusters that provides such identical results for both ATSP and AGTSP that the AGTSP clusters do not change the ordering of the optimal nodes. The cluster representatives of the AGTSP will follow the sequence of the nodes that results from an optimal global traversal of the nodes for the ATSP. As such, there were no differences in the results between the two solutions for any of the test cases. Additionally, the results are consistent with the results from [30] , as many of the GTSP solutions from the smaller test cases were able to produce optimal AGTSP solutions, provided that the cluster boundaries fall on the same boundaries as the natural divisions of a route.

F. Result

To compare the effectiveness of iterated local search methods for various TSP instances and their variants (ATSP and AGTSP), three ATSP datasets (Fischetti's instances from TSPLIB) were analyzed: ftv35, ry48p, and ft70. These datasets feature complete edge-weight matrices and were examined using Tabu search as a benchmark for exploration mechanics and baseline parameters. The datasets represent varying complexity levels to validate conclusions.

The algorithms were applied to AGTSP problem instances with increasing cluster sizes (8, 10, and 14) consistent with the instance described in the previous section. Visualization graphs illustrate the performance of hill climbing algorithms (first ascent, steepest ascent), simulated annealing, and genetic algorithms. Optimization paths and total distances from the ILS algorithm are shown in Table 3(b), facilitating a comparison of ATSP and AGTSP adaptability based on route complexity, with summarized visualization results provided in Table 5.

Table 3 (a)
Results for the analyzed TSPLIB instances for ATSP and AGTSP

Algorithm	Final Travelled Distance			
	Test instance	ATSP	Test instance	AGTSP
Tabu Search	*ftv35*	1543	*8ftv35*	481
	ry48p	14517	*10ry48p*	4559
	ft70	42515	*14ft70*	7486
First Ascent	*ftv35*	1615	*8ftv35*	501
	ry48p	14495	*10ry48p*	4559
	ft70	42283	*14ft70*	8489
Steepest Ascent	*ftv35*	1568	*8ftv35*	481
	ry48p	14453	*10ry48p*	4536
	ft70	42515	*14ft70*	7486
Simulated Annealing	*ftv35*	1667	*8ftv35*	481
	ry48p	15575	*10ry48p*	4536
	ft70	42515	*14ft70*	7560
Genetic Search	*ftv35*	2816	*8ftv35*	481
	ry48p	26650	*10ry48p*	4536
	ft70	58718	*14ft70*	7948

Table 3(b)
Results for the analyzed TSPLIB instances for ATSP and AGTSP

Algorithm	Instance	ATSP Best	ATSP Mean	ATSP SD	AGTSP Instance
Tabu Search	ftv35	1543	1646.0	55.3	Tabu Search
Tabu Search	ry48p	14517	14930.2	242.3	Tabu Search
Tabu Search	ft70	42515	43361.9	501.3	Tabu Search
First Ascent	ftv35	1615	1758.2	90.9	First Ascent
First Ascent	ry48p	14495	15134.9	297.5	First Ascent
First Ascent	ft70	42283	43341.9	580.0	First Ascent
Steepest Ascent	ftv35	1568	1635.4	38.0	Steepest Ascent
Steepest Ascent	ry48p	14453	14851.4	236.7	Steepest Ascent
Steepest Ascent	ft70	42515	43207.2	435.5	Steepest Ascent

Simulated Annealing	ftv35	1667	1833.6	83.5	Simulated Annealing
Simulated Annealing	ry48p	15575	16618.0	497.0	Simulated Annealing
Simulated Annealing	ft70	42515	43365.4	524.1	Simulated Annealing
Genetic Search	ftv35	2816	3127.6	124.6	Genetic Search
Genetic Search	ry48p	26650	30137.6	2379.7	Genetic Search
Genetic Search	ft70	58718	61748.8	1465.0	Genetic Search

IV. RESULTS & DISCUSSION

A. Analysis on Simulation Results

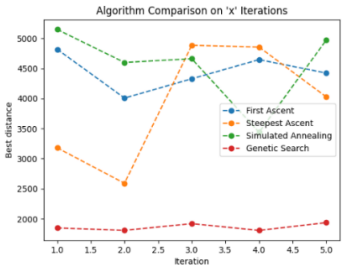
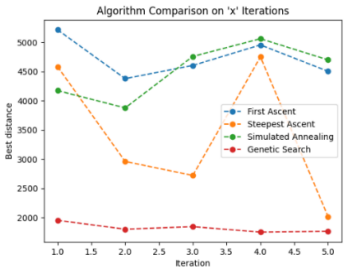
TSP problem instances tested in this study are obtained from the main TSPLIB repositories [8], [34]. TSPLIB is a collection of sample instances consisting of matrix-based datasets representing distances between nodes based on their spatial locations. In this research context, the iterated local search algorithms are implemented and evaluated through the conversion of fundamental TSP, together with the conversion of ATSP to the higher-complexity AGTSP. The study attempts to highlight the complex nature of transformation from TSP to more complex variants (ATSP and AGTSP) while maintaining similar solution exploration strategies. Before performing advanced evaluation on the algorithms' performance, the testing of the varied TSP instances is compared under similar nomenclature. Table 4 lists the selected analysis reference of TSPLIB data instances accumulated in this study.

Table 4
Analyzed TSPLIB instances for compatibility with proposed algorithm execution

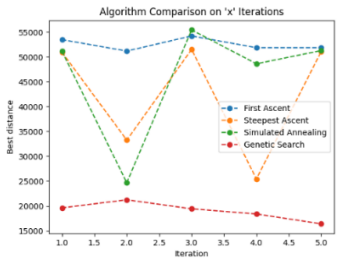
Capacity	Data Instance
Small	ulysses16, br17, ftv35, ftv38, city31, p43, ry48p
Medium	eil51, gr96, ft53, ft70, ftv70
Large	bier127, ftv170

As described previously, Table 4 summarizes several TSPLIB problem instances that adhere to the analysis framework described in the study. The instances are arranged by their dimensional characteristics based on distance metrics across TSP, ATSP, and AGTSP. This categorization highlights the dimensional variability of TSP, ATSP, and AGTSP problems in terms of instance size, structure, and complexity, while maintaining comparable evaluation criteria. The testing contributes to an understanding of the stability and generalizability of solutions to real-world applications, highlighting performance bottlenecks, and structural characteristics that influence solution quality, including cost matrices and node distribution in AGTSP instances. This study considers the impact of transitioning from TSP to ATSP, using the iterated local search compared across varying levels of node weight complexity. A key concern is how neighboring nodes influence shortest-path structures over iterations and convergence behavior. The study focuses on three primary datasets (ftv35, ry48p, ft70) and extends the analysis to AGTSP instances with varying cluster sizes to examine structural relationships, namely

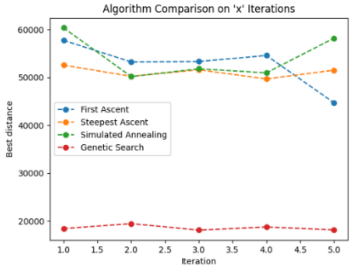
Table 5
Visualization of the performance for the compared iterated local search heuristics for ATSP & AGTSP

Problem Instance	ATSP	Problem Instance	AGTSP
ftv35		*8ftv35*	

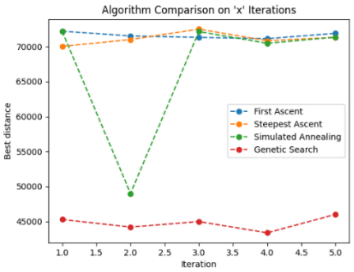
ry48p



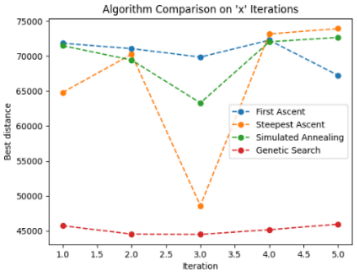
10ry48p



ft70



14ft70



B. Discussions

Table 5 depicts the comparison of algorithm performances across test instances with varying dimensional scales. A clear result in the experimental data was the consistent increase in total travel distance with increasing dimensionality and instance complexity. Of the seven algorithms tested, simulated annealing demonstrated a consistent increase in travel distance across the increasing cluster scenarios described previously, making it a good selection for mid-range logistics networks that place a higher priority on route consistency, for example postal delivery service in suburban areas. The genetic algorithm demonstrated a corresponding increase in route complexity with the dimensionality measured by the number of nodes visited, while still producing acceptable solutions. This characteristic is important for adaptive freight routing across variable warehouse distributions, where the number and capacities of warehouses can change dynamically. The steepest ascent method exhibited the highest variability among the heuristics, with variations in both path length and node count increasing as cluster density increased. This variability indicates that the steepest ascent may be suitable for exploratory scenarios where node arrangements shift dynamically, such as the deployment of mobile sensors for environmental monitoring. In such cases, path recalibration can be achieved in a responsive manner, albeit at the cost of reduced solution stability.

Across all heuristic variants, a clear trend was observed: total travel distances and node traversal increased when transitioning from ATSP to AGTSP, and this trend became more pronounced as cluster size increased. The AGTSP complexity parallels many real-world systems, like dispatch of emergency services in urban locations, as both asymmetric travel conditions and clustered demands must be addressed concurrently, such as in one-way street systems and geographically grouped service requests. These results corroborate that cluster dimensionality and constraint structure significantly affect metaheuristic solution efficiency. The results indicate that ILS serves as a strong optimization method that maintains solution quality across different routing situations that grow in complexity from their basic routing operations.

The ILS system with its developed improvements can be effectively applied in various routing domains including autonomous vehicle path planning, supply chain routing that works under demand uncertainty and agricultural mapping which requires unmanned aerial vehicle fleet coordination. These applications demand efficient and adaptive problem-solving capabilities to address complex operational requirements. The configuration and intensity of local search mechanisms have a direct impact on both solution quality and route complexity. The research results establish that ILS exploration needs calibration based on both the problem structure and the associated objective function.

V. CONCLUSION

This study investigated the compatibility of ILS across multiple TSP variants and examined how structural attributes of TSP, GTSP, and ATSP can be systematically combined to address the higher-order complexity of the AGTSP. The experimental study demonstrates a consistent increase in total travel distance as the problem dimensionality and instance complexity increase. The evaluation of routing algorithms showed that simulated annealing produced stable outcomes across increasing cluster dimensions, indicating that the method could be used for mid-sized logistics operations that require consistent delivery routes, such as suburban postal delivery services. The genetic algorithm's route complexity (travel distance) was proportional to the increase in nodes; however, it maintained feasible solutions that may be relevant in adaptive freight routing scenarios, where warehouse locations changes constantly. The steepest ascent results had the highest variability across path length and node coverage, which may reflect its exploration capability, such as for mobile sensor deployment, where responsiveness to dynamic node rearrangements is desirable at the cost of reduced route stability. For all heuristics, the results indicate that transitioning from ATSP to AGTSP introduces significantly higher complexity in balancing constraints across larger cluster structures. This reflects real-world scenarios in emergency logistics and urban service delivery operations, where asymmetric travel costs and clustered service requirements must be addressed simultaneously. The ILS method described here exhibits solid optimization capabilities across variants of TSP. ILS can be extended to complex environments, including autonomous vehicle routing, supply chain logistics under uncertainty, and UAV-based agricultural and mapping applications. These findings reaffirm that cluster dimensionality and constraint structure are significant determinants of metaheuristic performance and highlight the importance of developing adaptive search schemes tailored to the specific objectives and complexity of the routing problem.

ACKNOWLEDGMENT

The authors would like to thank the editors for their constructive reviews of proposed improvements regarding the paper. Moreover, the authors would like to express appreciation for Universiti Pendidikan Sultan Idris in terms of publication acknowledgements.

CONFLICT OF INTEREST

Authors declare that there is no conflict of interest regarding the publication of the paper.

AUTHOR CONTRIBUTION

Farid Morsidi (Conceptualisation; Methodology; Validation; Formal analysis; Data curation; Formal analysis; Investigation; Resources; Software; Visualisation; Writing - original draft; Writing - review & editing).

REFERENCES

[1] K. Corona-Gutiérrez, S. Nucamendi-Guillén, and E. Lalla-Ruiz, "Vehicle routing with cumulative objectives: A state of the art and analysis," *Comput. Ind. Eng.*, vol. 169, no. February, 2022, doi: 10.1016/j.cie.2022.108054.

[2] A. B. Doumi, B. A. Maafzah, and H. Hiary, "Solving traveling salesman problem using genetic algorithm based on efficient mutation operator," *J. Theor. Appl. Inf. Technol.*, vol. 99, no. 15, pp. 3768–3781, 2021, [Online]. Available: https://www.jatit.org/volumes/ninetynine15.php

[3] M. El Krari, B. Ahiod, and B. El Benani, "A Memetic Algorithm Based on Breakout Local Search for the Generalized Traveling Salesman Problem," *Appl. Artif. Intell.*, vol. 34, no. 7, pp. 537–549, 2020, doi: 10.1080/08839514.2020.1730629.

[4] A. E. S. Ezugwu, A. O. Adewumi, and M. E. Frîncu, "Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem," *Expert Syst. Appl.*, vol. 77, pp. 189–210, 2017, doi: 10.1016/j.eswa.2017.01.053.

[5] J. F. Castaneda Londono, R. A. G. Rendon, and E. M. T. Ocampo, "Iterated local search multi-objective methodology for the green vehicle routing problem considering workload equity with a private fleet and a common carrier," *Int. J. Ind. Eng. Comput.*, vol. 12, no. 1, pp. 115–130, 2020, doi: 10.5267/j.ijiec.2020.8.001.

[6] Y. Christopher, S. Wahyuningsih, and D. Satyananda, "Study of variable neighborhood descent and tabu search algorithm in VRPSDP," *J. Phys. Conf. Ser.*, vol. 1872, no. 1, 2021, doi: 10.1088/1742-6596/1872/1/012002.

[7] N. A. Fitriani, R. A. Pratama, S. Zahro, P. H. Utomo, and T. S. Martini, "Solving capacitated vehicle routing problem using saving matrix, sequential insertion, and nearest neighbor of product 'X' in Grobogan district," *AIP Conf. Proc.*, vol. 2326, 2021, doi: 10.1063/5.0039295.

[8] J. Ochelska-Mierzejewska, A. Poniszewska-Marańda, and W. Marańda, "Selected Genetic Algorithms for Vehicle Routing Problem Solving," *Electronics*, vol. 10, no. 24, p. 3147, Dec. 2021, doi: 10.3390/electronics10243147.

[9] V. F. Yu, P. Jodiawan, and A. A. N. P. Redi, "Crowd-shipping problem with time windows, transshipment nodes, and delivery options," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 157, no. December 2021, p. 102545, 2022, doi: 10.1016/j.tre.2021.102545.

[10] S. Wahyuningsih and D. Satyananda, "Improvement of solution using local search method by perturbation on VRPTW variants," *J. Phys. Conf. Ser.*, vol. 1581, no. 1, 2020, doi: 10.1088/1742-6596/1581/1/012004.

[11] S. Kunnapapdeelert and C. Thawnern, "Capacitated vehicle routing problem for thailand's steel industry via saving algorithms," *J. Syst. Manag. Sci.*, vol. 11, no. 2, pp. 171–181, 2021, doi: 10.33168/JSMS.2021.0211.

[12] P. Hungerländer and C. Truden, "Efficient and Easy-to-Implement Mixed-Integer Linear Programs for the Traveling Salesperson Problem with Time Windows," in *Transportation Research Procedia*, Elsevier B.V., 2018, pp. 157–166. doi: 10.1016/j.trpro.2018.09.018.

[13] M. G. Baldoquin, J. A. Martine, and J. Diaz-Ramirez, "A unified model framework for the multiattribute consistent periodic vehicle routing problem," *PLoS One*, vol. 15, no. 8 August, pp. 1–27, 2020, doi: 10.1371/journal.pone.0237014.

[14] D. Marković, G. Petrović, Ž. Ćojbašić, and A. Stanković, "The vehicle routing problem with stochastic demands in an Urban area – A case study," *Facta Univ. Ser. Mech. Eng.*, vol. 18, no. 1, pp. 107–120, 2020, doi: 10.22190/FUME190318021M.

[15] D. Wang, J. Jiang, R. Ma, and G. Shen, "Research on Hybrid Real-Time Picking Routing Optimization Based on Multiple Picking Stations," *Math. Probl. Eng.*, vol. 2022, no. 1, pp. 1–15, Apr. 2022, doi: 10.1155/2022/5510749.

[16] Y. Shi, L. Lv, F. Hu, and Q. Han, "A heuristic solution method for multi-depot vehicle routing-based waste collection problems," *Appl. Sci.*, vol. 10, no. 7, 2020, doi: 10.3390/app10072403.

[17] F. N. Rizkiani, I. Geraudy, and A. Imran, "Solving the Vehicle Routing Problem with Multiple Trips and Simultaneous Delivery-Pickup for Drinking Water Distribution Company," *E3S Web Conf.*, vol. 484, 2024, doi: 10.1051/e3sconf/202448401003.

[18] F. Morsidi, "Distribution Path Segmentation Using Route Relocation and Savings Heuristics for Multi-Depot Vehicle Routing," *Malaysian J. Sci. Adv. Technol.*, vol. 3, no. 2, pp. 72–80, May 2023, doi: 10.56532/mjsat.v3i2.154.

[19] J. Schmidt and S. Irnich, "New neighborhoods and an iterated local search algorithm for the generalized traveling salesman problem," *EURO J. Comput. Optim.*, vol. 10, 2022, doi: 10.1016/j.ejco.2022.100029.

[20] Y. U. Kasanah, N. N. Qisthani, and A. Munang, "Solving the Capacitated Vehicle Routing Problem with Heterogeneous Fleet Using Heuristic Algorithm in Poultry Distribution," *J. Ilm. Tek. Ind.*, vol. 21, no. 1, pp. 104–112, 2022, doi: 10.23917/jiti.v21i1.17430.

[21] F. Tunnisaki and Sutarman, "Clarke and Wright Savings Algorithm as

Solutions Vehicle Routing Problem with Simultaneous Pickup Delivery (VRPSPD)," *J. Phys. Conf. Ser.*, vol. 2421, no. 1, p. 012045, 2023, doi: 10.1088/1742-6596/2421/1/012045.

[22] Y. Zhang, Y. Liu, C. Li, Y. Liu, and J. Zhou, "The Optimization of Path Planning for Express Delivery Based on Clone Adaptive Ant Colony Optimization," *J. Adv. Transp.*, vol. 2022, 2022, doi: 10.1155/2022/4825018.

[23] F. Morsidi, "Multi-Depot Dispatch Deployment Analysis on Classifying Preparedness Phase for Flood-Prone Coastal Demography in Sarawak," *J. ICT Educ.*, vol. 9, no. 2, pp. 175–190, Dec. 2022, doi: 10.37134/jictie.vol9.2.13.2022.

[24] M. Han and Y. Bin Wang, "A Solution to VRPTW Based on Improved GA-AMMAS Algorithm," *J. Phys. Conf. Ser.*, vol. 1486, no. 3, 2020, doi: 10.1088/1742-6596/1486/3/032015.

[25] Y. Kocaoglu, E. Cakmak, B. Kocaoglu, and A. Taskin Gumus, "A Novel Approach for Optimizing the Supply Chain: A Heuristic-Based Hybrid Algorithm," *Math. Probl. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/3943798.

[26] B. Li, G. Wu, Y. He, M. Fan, and W. Pedrycz, "An Overview and Experimental Study of Learning-Based Optimization Algorithms for the Vehicle Routing Problem," *IEEE/CAA J. Autom. Sin.*, vol. 9, no. 7, pp. 1115–1138, 2022, doi: 10.1109/JAS.2022.105677.

[27] W. Cook, S. Held, and K. Helsgaun, "Constrained Local Search for Last-Mile Routing," *Transp. Sci.*, vol. 58, no. 1, pp. 12–26, 2024, doi: 10.1287/trsc.2022.1185.

[28] V. Kastrati and A. Ahmeti, "Solving Vehicle Routing and Scheduling with Delivery and Installation of Machines using ILS," *13th Int. Conf. Pract. Theory Autom. Timetabling*, vol. I, pp. 207–223, 2021.

[29] L. Cheng Hao, W. Yan Hong, Q. J. Wei, D. W. Zhao, and S. Rui, "Product Service Scheduling Problem with Service Matching Based on Tabu Search Method," *J. Adv. Transp.*, vol. 2020, 2020, doi: 10.1155/2020/5748680.

[30] P. C. Pop, O. Cosma, C. Sabo, and C. P. Sitar, "A comprehensive survey on the generalized traveling salesman problem," *Eur. J. Oper. Res.*, vol. 314, no. 3, pp. 819–835, 2024, doi: 10.1016/j.ejor.2023.07.022.

[31] K. Singh, S. K. Bedi, and P. Gaur, "Identification of the most efficient algorithm to find Hamiltonian Path in practical conditions," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2020, pp. 38–44. doi: 10.1109/Confluence47617.2020.9058283.

[32] C. E. Noon and J. C. Bean, "An Efficient Transformation Of The Generalized Traveling Salesman Problem," *INFOR Inf. Syst. Oper. Res.*, vol. 31, no. 1, pp. 39–44, 1993, doi: 10.1080/03155986.1993.11732212.

[33] S. S. Juneja, P. Saraswat, K. Singh, J. Sharma, R. Majumdar, and S. Chowdhary, "Travelling Salesman Problem Optimization Using Genetic Algorithm," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, IEEE, Feb. 2019, pp. 264–268. doi: 10.1109/AICAI.2019.8701246.

[34] M. Liu *et al.*, "A two-way parallel slime mold algorithm by flow and distance for the travelling salesman problem," *Appl. Sci.*, vol. 10, no. 18, 2020, doi: 10.3390/APP10186180.