JOIN (Jurnal Online Informatika)
p-ISSN: 2528-1682, e-ISSN: 2527-9165
Volume 11 Number 1 | June 2026: 82-97
DOI: 10.15575/join.v11i1.1539

Vehicle Routing Problem: A Performance Comparison of Hybrid Evolutionary Algorithm with Local Search Strategies

Maria Ulfah Siregar[1], Thoriq Firdaus Arifin[2], Muhammad Javier Badruttamam[3], Maulida Suryaning Aisha[4], Ibnu Raju Humam[5], Muhammad Hafiz[6], Siti Mutmainah[7]
[1,2,3,4,5,6,7]Department of Informatics, Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta, Indonesia

Article Info

Article history:

Received December 28, 2024
Revised March 22, 2025
Accepted May 5, 2025
Published April 25, 2026

Keywords:

Genetic Algorithm
Local Search
Metaheuristic Hybridization
Particle Swarm Optimization
Simulated Annealing
Vehicle Routing Problem

ABSTRACT

The Vehicle Routing Problem (VRP), one of the most challenging problems in logistics and transport, has been an area of optimization solutions to minimize costs and optimize the operational process. This study examines a hybrid of metaheuristic algorithms that are combinations of the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), and Local Search (LS) to tackle various complexities of VRP. The hybrid approach offered better exploration and exploitation by integrating global explorations with GA and PSO and local refinement with SA and LS. The performance was performed using real datasets and generated randomly with problem sizes ranging from 9 to 100 customers. PSO-LS and GA-LS are LS-based hybrids that produce lower standard deviations, showing a stable and consistent result for small to medium problems. For example, PSO-LS computed 3.31 for 9 customers and 5.76 for 50 customers. However, SA-based hybrids, such as PSO-SA and GA-SA, presented more variability, with SA-GA reaching 100 customers as much as 7.83. These findings highlight key trade-offs while optimizing VRP between stability, efficiency, and problem scale.

Corresponding Author:

Thoriq Firdaus Arifin
Informatics Department, Faculty of Science & Technology, Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta
Jl. Laksda Adisucipto, Depok, Sleman, Daerah Istimewa Yogyakarta 55281
Email: thoriqfarf@gmail.com

Maria Ulfah Siregar
Informatics Department, Faculty of Science & Technology, Universitas Islam Negeri (UIN) Sunan Kalijaga Yogyakarta
Jl. Laksda Adisucipto, Depok, Sleman, Daerah Istimewa Yogyakarta 55281
Email: maria.siregar@uin-suka.ac.id

1. INTRODUCTION

Metaheuristics are a set of generic constructed algorithms that are not reliant on any specific problem and are intended to solve many optimization tasks [1][2]. This is one of the newer advances that have attracted much interest in recent decades due to its applicability in complex scenarios where standard searching methods will only work in one area.

On the other hand, one of the most challenging issues: the so-called vehicle routing problem (VRP), which was raised by [3], placing it in the class of significant problems in distribution network optimization [4]. The solution includes finding the minimum cost of the customer service within the vehicle route associated with finding such a minimum cost service, which is classified as NP-hard [2][5]. It is crucial, however, to balance this coordination with the quality of the solution.

Metaheuristics have developed the multi-metaheuristic approach to pragmatically solve the many versions of the VRP, which broadened the targeting of the hybrid multicriteria genetic algorithm and all it aimed at [6]. As the research [1] demonstrates, by merging the Honey Badger Algorithm (HBA) with Sand Cat Swarm Optimization (SCSO), the combined application of approaches performs better than using them separately, which enables one to solve the problem on time and arrive at the optimal solution. Similarly, paper [7] brings to the fore the I-HFPSO algorithm, which has surpassed average 'traditional' alternatives, such as ArcGIS, for instance, and linear programming in both speed and performance. In other research, the hybridization has paired a genetic algorithm and a method from the tree boosting system, namely XGBoost [8], in which the hybridization yields better results than the solely use of each algorithm; the genetic algorithm was used to obtain the best parameters to optimize the random forest [9], the hybridization between brute force algorithm and genetic algorithm to optimize data search process [10].

It is concluded in [11] that the integration of global exploration from genetic algorithms or other similar algorithms, with local exploitation using metaheuristic algorithms, is able to increase the stability of solution quality in complex timetable problems.

Work [12] extends the research on the path planning of vehicles with the proposed blending of the Ant Colony and Particle Swarm Algorithms (ACPSA). This method incorporates planning for the shortest route while responding to real-time and robust performance issues. Also, research [13] uses Solomon's set of data to demonstrate the effect of the technique called metaheuristic link prediction (MLP) and Ant Colony Optimization based on GA on the travel expenses computation time and the number of vehicles used as compared with other algorithms such as Genetic Algorithm-based ACO, Simulated Annealing models, and Memetic Models. Research [14] deals with the problem of the logical design of a blood supply network by describing the blood supply based on the use of unmanned aerial vehicles, Genetic Algorithm, and Greedy Search, which helps in reducing the overall blood supply cost and is also more efficient in terms of computation time compared to the CPLEX solver.

Unlike previous studies, this research provides a wider performance benchmark for Hybrid Evolutionary Algorithms, focusing on combining Genetic Algorithm and Particle Swarm Optimization with further guidance from Local Search methods. We would like to assess the performance of these approaches on the VRP in terms of solution quality, computation time, and applicability in different operational environments. The uniqueness of this study lies in comparing these hybrid methods to address the complex nature of the VRP and improve solution efficiency and scalability for distribution network optimization.

2. METHOD

Regarding optimization problems, especially NP-hard ones like the Multiple Objective Vehicle Routing Problem (MOPRV), metaheuristics are general algorithms that are made to help. This necessitates a balance between exploring the solution space broadly and refining solutions in promising regions [15]. This paper suggests a mixed metaheuristic that works well with global and local optimization methods to meet these needs. As illustrated in Figure 1, we combine the exploration strengths of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), which can explore various solution areas. Simultaneously, we apply Simulated Annealing (SA) and Local Search (LS) to improve the accuracy of solutions obtained through searches within the local neighborhood, ultimately boosting the efficacy of the solution progression of our approach.

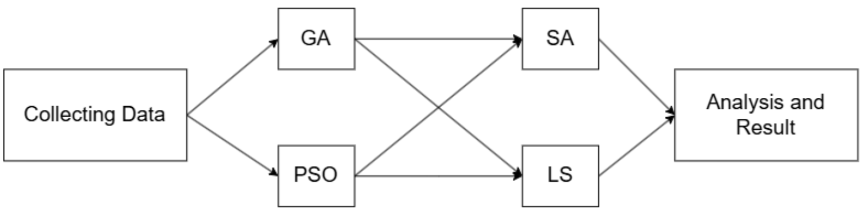


Figure 1. Flow Chart Hybrid Method

This mix of algorithms has a fair trade-off: the global search algorithms look across the whole solution space, while the local search algorithms focus on a small part. Our approach consists of collecting specific data and an incremental and collaborative application of these metaheuristics. The following sections will detail GA, PSO, SA, and LS and discuss how integrating these techniques provides a strong framework for the VRP solution.

2.1. Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is the most important combinatorial optimization problem for logistics, and it is used to find a set of low-cost routes for a fleet of vehicles delivering goods to a set of customers from a central warehouse [16]. It is an NP-hard problem, and the computational burden becomes more serious with increased data scales, making it challenging to obtain optimal solutions. Various heuristics/prioritized agendas were developed to help promote a more efficient problem-solving process. Several exceptional cases of the VRP have emerged to solve different logistics demands [17]. However, new algorithms and models are still emerging to fill the gaps in what is known. Considering the integration of algorithms and optimization processes is crucial, which will lead to the acceleration and effectiveness of solution generation, especially when faced with large and complex real-world problems. These models do not only solve identified problems; they will also discover new ways to optimize routes.

In the present work, we put together VRP instances from publicly available data on Kaggle, for which we have the area coordinates of customers, distances, and total demand volume of customers. This dataset is a close simulation of actual logistics conditions and allows a comprehensive evaluation of the performance of the presented algorithm. Various parameters, such as the number of vehicles and customers' positions, were changed to evaluate the algorithm's efficiency for different degrees of complexity. These variants aim to analyze the algorithm properties and explore the possibilities of improved performance under different sets of problems, especially when addressing different problem classes, as the VRP is given by Equation 1.

$$Fitness = TotalDistance + (5 \cdot numVehicle \cdot \sum(Difference\ in\ Load)) \quad (1)$$

The fitness function, which has two main components, is used to assess the quality of solutions in VRP. $TotalDistance$ denotes the sum of distances accumulated over all routes for which it is $TotalDistance$ for the travel operations costs, such as fuel and time. Load balance is quantified as $numVehicle$, which is the summation of the deviation of actual load from the capacity per vehicle, a measure of the deviation of actual load from the height of a vehicle, given the number of used vehicles. A penalty constant of 5 punishes this form's lack of load balance and improves the optimization quality.

2.2. Genetic Algorithm

Genetic algorithms (GA) are utilized in various studies as a part of the hybrid metaheuristic method. Often, these strategies are hybridized with other techniques to improve the performance of the GA further in solving the Vehicle Routing Problem (VRP). For example, a hybrid GA without trip delimiters is suggested to eliminate the necessity for repair procedures and enhance the solution quality [18]. Genetic Algorithm, besides being the most commonly used algorithm for optimization search problems, is a member of the class of evolutionary algorithms based on the principles of biological evolution. The GA process involves several stages, from the simplest to the most complex, as illustrated in Figure 2 [19], [20], [21].

1. **Population Initialization**: The process starts with an initial population of individuals, meaning a collection of strings of bits referred to as solution encodings. This representation is a string or a list of numbers corresponding to the parameters in the solution. We create this population randomly to ensure initial diversity.
2. **Feasibility Evaluation**: All solutions or chromosomes in the population are then evaluated according to a feasibility measure or function, which indicates the solution's efficiency with respect to the problem. The higher the level of feasibility, the higher its probability of being chosen for the next generation.

3. **Selection**: The selection process selects chromosomes from the current population to be the next generation's parents. If its feasibility value passes the constraint, cycling selects the chromosome or solution with the most considerable value. Roulette wheel selection and tournament selection are some of the methods that are frequently used.
4. **Crossover**: Creating one or more offspring involves using two selected parent chromosomes. It is like sexual reproduction aimed at combining the parents' best qualities. Single-point crossovers, two-point crossovers, and other variants can be present.
5. **Mutation**: Mutations are changes in the species' chromosomes that occur to ensure genetic diversity and stop population stagnation. Typically, mutations occur with low probability and result in a replacement (substitution) of one or a few genes on the chromosome.
6. **Population Update**: Children generated from this process constitute the new population after crossover and mutation. This process is repeated so that the next generation is the product of selection, crossover, and mutation being applied to the new population. The above procedure continues until the population viability is acceptable or a fixed number of generations is met.

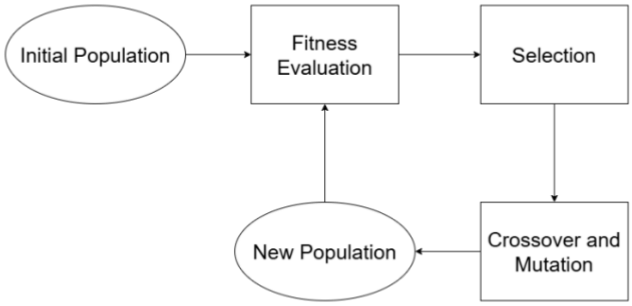


Figure 2. Genetic Algorithm Procedure

Genetic Algorithm (GA), the backbone of the hybridization procedure, has significantly improved overall VRP optimization performance. GA is an attractive browsing tool because it can browse large and even complex solution spaces based on its evolutionary technique. Using schemes like selection, crossover, and mutation, GAs can create pools of diverse candidate solutions that can be iterated, modeled, and tailored for the problem. This flexibility is important for addressing other aspects of the VRP that may be difficult to solve by conventional optimization methods. Further, other heuristics or optimization schemes can be included in the GAs effortlessly to synergize the strengths of different approaches. While using the best-known solutions, the large number of possible combinations helps the GAs avoid getting stuck in local optima, allowing them to find high-quality solutions quickly and effectively handle complex and changing routing problems [22].

2.3. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population-based optimization algorithm motivated by the social interactions of birds and fish. The optimization field widely accepts its fast convergence and ease of implementation. Nevertheless, PSO tends to have difficulty if the problem is high-dimensional; sometimes, it converges to local optima too soon and has difficulties finding the global optimal solution. To counter the drawback, PSO is usually hybridized with other techniques like GA and SA to improve its global exploration capability and to help achieve better solutions [23], [24], [25]. The PSO (iteration) learning rules are given by the following expressions (Equation 2-4) [26]:

$$v_i(t+1) = \omega \cdot v_i(t) + c_1 \cdot r_1 \cdot \left(p_{best,i} - x_i(t)\right) + c_2 \cdot r_i \cdot \left(g_{best} - x_i(t)\right) \quad (2)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

$$Probability = \frac{v_i}{maxVelocity} \quad (4)$$

The inertia factor ω is considered the most important parameter, as it affects the contribution of previous velocities to the current velocities, striking an appropriate balance of exploration (to look for new regions) and exploitation (to refine the current best). The cognitive and social coefficients, c_1 and c_2, control individual particles to update their location, considering the relationship between their

p-ISSN: 2528-1682
e-ISSN: 2527-9165

personal best position (p_{best}) and the global best position (g_{best}), discovered by everyone in the swarm. Adding random values r_1 and r_2Ranging in [0,1] can improve system randomness and make particles escape from the local optimum.

The new velocity of the particle is referred to as $v_i(t+1)$ and is affected by the previous position p_{best} of the particle and the g_{best}. Therefore, in the new position ,$x_i(t+1)$ we will see how each particle visited the search space in the next iteration. This modified velocity vector makes the mean vector balance the distance and direction of movement, and particles can traverse the search space well enough to help them search for near-optimal solutions.

The added value of PSO is its ability to utilize the social behavior of a group of particles for joint exploration in various space regions. This feature not only assists in finding a good set of solutions time-efficiently but also makes PSO a good choice to be hybridized with other optimization methods. By improving the global scanning ability and avoiding premature convergence towards local solutions, PSO is a good building block for hybridizing metaheuristics. The structure and dynamics of the PSO generally allow it to be a valuable tool for resolving complicated optimization issues in a variety of domains [27].

2.4. Simulated Annealing

Simulated Annealing (SA) is a probabilistic method for finding the suboptimal solution to an optimization problem. Its effectiveness is derived from the fact that it can escape from local maxima and minima through uphill moves, making exploring the solution space possible. This property makes SA particularly useful for studies of rugged landscapes, where local optima may prevent global optimization. SA gets frequently hybridized with other algorithms to improve its local search adoption. For example, hybrid algorithms that integrate SA with Particle Swarm Optimization (PSO) exploit the global exploration ability of PSO as well as the rapid convergence ability of SA and, therefore, have better optimization performance [23][28][29].

The foundation of Simulated Annealing involves evaluating potential solutions based on their fitness difference Δ and a temperature parameter, which controls the acceptance of new solutions. The acceptance probability P is given by Equation 5 [30], [31], [32]:

$$P = \exp\left(-\frac{\Delta}{T}\right) \tag{5}$$

The subsequent temperature adjustment follows as Equation 6:

$$T_{next} = T \cdot coolingRate \tag{6}$$

Here, the fitness difference is defined as $\Delta = Fitness_{neighbor} - Fitness_{current}$, and T represents the current temperature, which decreases exponentially during the process. Solutions are accepted if the neighboring solution shows better fitness ($\Delta < 0$). Conversely, if the neighboring solution is less favorable ($\Delta > 0$), it may still be accepted with a probability P, allowing the algorithm to escape local optima.

SA hybridized with GA and PSO offers unique advantages. In combination with GA, the SA that makes the global search mechanism with new probability acceptances of solutions can enable it to keep solution diversity and prevent the population from premature local optima convergence. This is an invaluable advantage, especially for multi-modal optimization space, since there are often many local optima. The combination of SA and PSO benefits from PSO's ability to quickly find good areas to search while also using SA's method of improving solutions by sometimes accepting worse ones as the temperature cools. These methods combine to allow an in-depth exploration and exploitation strategy, which can improve the solution quality and optimization efficiency. At last, SA has become an important tool in hybrid algorithms, improving the reliability and efficiency of optimization in various problems [33].

2.5. Local Search

Local Search (LS) algorithms are methods that improve the solution of an optimization problem by looking at the current solution's neighborhood in the solution space. These iterative algorithms serve as post-processes, effectively refining the results generated by other methods. LS algorithms can also be integrated into bigger optimization structures, such as PSO, to manage the search technique, lower redundancy, and improve feature selection [23][34]. LS methods, such as the 2-opt heuristic, also improve local solutions by swapping chunks of a route and evaluating new fitness values, leading to performance jumps. This operation is mathematically defined as,

$$NewRoute = Reverse(Route[i{:}\, j + 1]) \quad (7)$$

Here is the variable. The variable $Route$Equation 7 represents the current route under scrutiny, and i and j indicate the beginning and ending indexes of the segment to be reversed. The $Reverse$ operation takes the points from the index i to $j + 1$ of the route and creates a new route named $NewRoute$. We evaluate the newly generated route to get its fitness value, which is typically measured by the total travel distance or the effectiveness of load balance on the established route. Around the current solution, the above process is carried out iteratively to look for the neighborhood and the curvature of the solution space that is detected, leading to an improved quality of solution [35][36].

Great benefits derive from the hybrid of Local Search and Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). It is used together with the GA to facilitate the convergence of solutions through the genetic operation; the tendency to premature convergence gets reduced, and population diversity is promoted [24][34]. This synergy facilitates the algorithm in searching a more expansive solution space rather than local searching, simultaneously refining the quality of the individuals in the population. Similarly, combining LS with PSO provides a straightforward method to improve how quickly the group finds better solutions by gradually improving the options they create. The specific search for the best LS can effectively facilitate the promising regions generated by PSO and thus accelerate the convergence speed of the solution. Hence, Local Search drives the hybridization procedures, advancing optimization algorithms' overall efficiency and robustness in various challenging problem instances.

2.6. Hybridization Process

The hybridization process entails merging different optimization algorithms to maximize their combined strengths while minimizing shortcomings. This technique is beneficial in the solution of complicated optimization problems. For example, the combination of PSO with GA or SA has been reported to perform better in balancing exploration and exploitation, improving solutions' accuracy and efficiency [37][38]. These hybrid techniques are utilized in different areas, including feature selection, task scheduling, and image encryption, proving their functionality and efficiency [25], [28], [29].

Genetic Algorithm (GA) with Simulated Annealing (SA) synergizes GA and SA by using SA's probabilistic acceptance mechanism to diversify GA. In this hybrid, we first generate the initial population by selecting, crossing, and mutating in GA to yield various solutions. SA is then employed for all offspring, which further fine-tunes the solutions by examining the neighboring routes and possibly accepting them according to the fitness difference and the temperature trend. In this way, the solution set can escape local optima since the GA will put previous solutions in those previously described in other and less stressed areas (Algorithm 1).

GA of LS seeks to preserve GA's global exploration ability while utilizing LS's ability to refine a local part of the search space. Using GA to get a diverse initial population, LS is then performed in the child solution to search and enhance local areas of the solution space. The local improvements by local approaches (including the 2-opt heuristic) are tested for the fitness value; only good or better individuals are stored and used to replace them as early fitness improvement solutions. This strategy results in smoother searches and is particularly effective at problem instances of moderate size and complexity (Algorithm 2).

Particle Swarm Optimization (PSO) combined with Simulated Annealing (SA) applies the global search power of PSO with SA's local optimization ability. To start, PSO generates a swarm of particles that explore the solution space and change their positions according to their own and global bests. As the optimization proceeds, SA is applied to the best swarm solutions, which explore the neighboring

p-ISSN: 2528-1682
e-ISSN: 2527-9165

solutions based on probabilistic rules. Mainly to avoid premature convergence in the multidimensional space, as is usual for the PSO, and to permit a deeper search over the solution space (Algorithm 3).

Finally, the PSO-LS hybrid addresses the trade-off between global search effectiveness and local optimization improvement. It starts with the swarm navigating the solution space according to the PSO framework, where the motion of each particle is affected by its individual optimum and the swarm's global optimum. Following PSO iterations, we apply the LS method to the optimal solutions, enabling the local refinement of route segments. This approach improves the quality and stability of the total solution and yields good results, especially for multi-scale problems (Algorithm 4).

Algorithm 1. Proposed Hybrid GA-SA Pseudocode

Algorithm1 GA-SA

Input: Population size parameters λ, μ; termination criteria; stagnation criteria; initial temperature T_0; minimum temperature T_{min}

Output: Best global solution and its fitness history

Initialize the $Population$ with random solutions
while $Termination$ criteria not met **do**
  $ParentPairs \leftarrow SelectParentPairs(Population)$
  $Children \leftarrow \bigcup_{pair \in ParentPairs} Crossover(pair)$
  Apply SA **for** each $child \in Children$ **do**
    $T \leftarrow T_0$
    while $T > T_{min}$ **do**
      $neighbor \leftarrow SelectNeighbor(child)$
      $f_{neighbor} \leftarrow Fitness(neighbor)$
      if $f_{neighbor} > f_{child}$ **then** $child \leftarrow neighbor$
      else
        accept with probability $p \leftarrow e$^$((f_{neighbor} - f_{child})/T)$
        if $random(0,1) < p$ **then** $child \leftarrow neighbor$ **end if**
      $T \leftarrow \alpha \cdot T$ where $0 < \alpha < 1$
    end while
  end for
  for each $child \in Children$ **do**
  $f_{child} \leftarrow Fitness(child)$
  end for
  if $|Population| > (\lambda + \mu)$ **then**
    $Population \leftarrow DiscardWorst(Population, \lambda + \mu)$
  end if
  $BestSolutionHistory \leftarrow UpdateBestSolution(Population, BestSolutionHistory)$
  if $CheckStagnation(BestSolutionHistory)$ **then** $Terminate\ Algorithm$ **end if**
end while

Algorithm 2. Proposed Hybrid GA-LS Pseudocode

Algorithm 2 GA-LS

Input: Population size parameters λ, μ; termination criteria; stagnation criteria

Output: Best global solution and its fitness history

Initialize the $Population$ with random solutions
while $Termination$ criteria not met **do**
  $ParentPairs \leftarrow SelectParentPairs(Population)$
  $Children \leftarrow \bigcup_{pair \in ParentPairs} Crossover(pair)$
  Apply LS **for** each $child \in Children$ **do**

Algorithm 2 GA-LS

6. **for** each $route \in child$ **do**
7. $neighborRoute \leftarrow LocalChange(route)$
8. $f_{neighborRoute} \leftarrow Fitness(neighborRoute)$
9. **if** $f_{neighborRoute} > f_{route}$ **then** $route \leftarrow neighborRoute$ **end if**
10. **end for**
11. **end for**
12. **for** each $child \in Children$ **do** $route \leftarrow Fitness(child)$ **end for**
13. $Population \leftarrow Population \cup Children$
14. **if** $|Population| > (\lambda + \mu)$ **then**
15. $Population \leftarrow DiscardWorst(Population, \lambda + \mu)$
16. **end if**
17. $BestSolutionHistory \leftarrow UpdateBestSolution(Population, BestSolutionHistory)$
18. **if** $CheckStagnation(BestSolutionHistory)$ **then** $Terminate\ Algorithm$ **end if**
19. **end while**

Algorithm 3. Proposed Hybrid PSO-SA Pseudocode

Algorithm 3 PSO-SA

Input: Swarm size parameters N; termination criteria; stagnation criteria; initial temperature T_0; minimum temperature T_{min}

Output: Best global solution and its fitness history

1. **for** each particle $i \in \{1,2,\dots,N\}$ **do** $x_i \leftarrow RandomPosition()$; $v_i \leftarrow RandomVelocity()$; $personal\ best\ P_i \leftarrow x_i$, $f_i \leftarrow Fitness(x_i)$; $personal\ best\ fitness\ f_{Pi} \leftarrow f_i$
2. **end for**
3. $g \leftarrow \arg\max_{Pi} f_{Pi}$, $f_g \leftarrow f_{Pi}\ corresponding\ to\ g$; $BestSolutionHistory \leftarrow g, f_g$
4. **while** $Termination$ criteria not met **do**
5. **for** each particle $i \in \{1,2,\dots,N\}$ **do**
6. $v_i \leftarrow \omega \cdot v_i + c_1 \cdot r_1 \cdot (p_i - x_i) + c_2 \cdot r_i \cdot (g_{best} - x_i)$
7. $x_i \leftarrow UpdatePosition(x_i, v_i)$
8. $f_i \leftarrow Fitness(x_i)$
9. **if** $f_i > f_{Pi}$ **then** $P_i \leftarrow x_i$; $f_{Pi} \leftarrow f_i$ **end if**
10. **end for**
11. $g_{new} \leftarrow \arg\max_{Pi} f_{Pi}$; $f_{gnew} \leftarrow f_{Pi}\ corresponding\ to\ g_{new}$
12. **if** $f_{gnew} \leftarrow f_g$ **then** $g \leftarrow g_{new}$; $f_g \leftarrow f_{gnew}$ **end if**
13. Apply SA **for** $SA(g, f_g)$ **do**
14. $T \leftarrow T_0$
15. **while** $T > T_{min}$ **do**
16. $g_{neighbor} \leftarrow SelectNeighbor(g)$
17. $f_{neighbor} \leftarrow Fitness(g_{neighbor})$
18. **if** $f_{neighbor} > f_g$ **then** $g \leftarrow g_{neighbor}$; $f_g \leftarrow f_{neighbor}$
19. **else**
20. accept with probability $p \leftarrow e$^$((f_{neighbor} - f_g)/T)$
21. **if** $random(0,1) < p$ **then** $g \leftarrow g_{neighbor}$; $f_g \leftarrow f_{neighbor}$ **end if**
22. **end if**
23. $T \leftarrow \alpha \cdot T$
24. **end while**
25. **return** g, f_g
26. $BestSolutionHistory \leftarrow UpdateBestSolution(BestSolutionHistory,\ g, f_g)$
27. **if** $CheckStagnation(BestSolutionHistory)$ **then** $Terminate\ Algorithm$ **end if**
28. **end while**

Algorithm 4. Proposed Hybrid PSO-LS Pseudocode

Algorithm 4 PSO-LS

Input: Swarm size parameters N; termination criteria; stagnation criteria

Output: Best global solution and its fitness history

1. **for** each particle $i \in \{1,2,\dots,N\}$ **do** $x_i \leftarrow RandomPosition()$; $v_i \leftarrow RandomVelocity()$; $personal\ best\ P_i \leftarrow x_i$, $f_i \leftarrow Fitness(x_i)$; $personal\ best\ fitness\ f_{Pi} \leftarrow f_i$
2. end for
3. $g \leftarrow \arg\max_{Pi} f_{Pi}$, $f_g \leftarrow f_{Pi}\ corresponding\ to\ g$; $BestSolutionHistory \leftarrow g, f_g$
4. **while** $Termination$ criteria not met **do**

Algorithm 4 PSO-LS

for each particle $i \in \{1,2,\ldots,N\}$ **do**
  $v_i \leftarrow \omega \cdot v_i + c_1 \cdot r_1 \cdot (p_i - x_i) + c_2 \cdot r_i \cdot (g_{best} - x_i)$
  $x_i \leftarrow UpdatePosition(x_i, v_i)$
  $f_i \leftarrow Fitness(x_i)$
  if $f_i > f_{Pi}$ **then** $P_i \leftarrow x_i; f_{Pi} \leftarrow f_i$ **end if**
end for
$g_{new} \leftarrow {}^{\arg max}_{Pi} f_{Pi}; f_{gnew} \leftarrow f_{Pi}\ corresponding\ to\ g_{new}$
if $f_{gnew} \leftarrow f_g$ **then** $g \leftarrow g_{new}; f_g \leftarrow f_{gnew}$ **end if**
Apply LS **for** each $route\ r \in g$ **do**
  $r_{neighbor} \leftarrow LocalChange(r)$
  $f_{r_{neighbor}} \leftarrow Fitness(r_{neighbor})$
  if $f_{r_{neighbor}} > f_r$ **then** $r \leftarrow r_{neighbor}; f_r \leftarrow f_{r_{neighbor}}$ **end if**
  end for
  $f_g \leftarrow Fitness(g)$
  return g, f_g
$BestSolutionHistory \leftarrow UpdateBestSolution(BestSolutionHistory,\ g, f_g)$
if $CheckStagnation(BestSolutionHistory)$ **then** $Terminate\ Algorithm$ **end if**
end while

3. RESULT AND DISCUSSION

The VRP is a prime optimization puzzle within the logistics and transportation industries. The VRP is quite complicated yet practically relevant. The present work conducts an empirical investigation into the synergetic influences exerted by a plethora of parameters on the VRP optimization process and any effects these parameters might have on the complexity of the problem and the possible solution space. The study of VRP instances included cases with varying numbers of vehicles and customers; these are well known to directly influence the extent and complexity of the optimization problems they address. Key algorithmic parameters considered include population size, number of generations, cooling rate, mutation rate, crossover rate, and stagnation count. The results aim to do justice to the interactions developed by the various parameters on the nature of the VRP solutions that help understand the balancing between computational efficiency and solution quality.

3.1. Data Description

The dataset used in this research is a very heterogeneous set of VRP instances with variability in the number of vehicles and clients. The base dataset, sourced initially from Kaggle (Vehicle Routing Dataset), consists of problem instances with 9 customers. To create a broader dataset for this study, additional random data was generated by scaling the original dataset to create instances with 25, 50, and 100 customers. The number of vehicles in each problem instance varies from 2, 3, and 5, and the dataset reflects both small and medium-sized problem instances. The number of vehicles introduces the scaling factor, influencing the complexity of the solution space. In contrast, the interaction between vehicles and customers represents a problem that must be solved.

Table 1. Hyperparameter Values for GA, PSO, SA, and LS

Algorithm	Hyperparameter	Value
Genetic Algorithm (GA)	population_size	20
	max_generations	100
	mutation_rate	0.01
	crossover_rate	0.7
Particle Swarm Optimization (PSO)	w	0.5
	c1	1.0
	c2	1.0
	max_iterations	100
	population_size	20
Simulated Annealing (SA)	initial_temperature	1000

	cooling_rate	0.95
Local Search (LS)	max_local_iterations	20
	improvement_threshold	0.001

The optimization process is informed by several important algorithmic parameters shown in Table 1. For stagnation, the count monitors the number of iterations when no improvement has been seen; within most examples, the cutoff is 50 iterations. This cutoff signals the algorithm to know when and how to start evolving to search for new solutions. These parameters characterize the degree and efficiency of the optimization process since they set limits of computational effort against the quality of the solution found. The results from this experiment, outlining how parameters may influence the performance of the VRP optimization algorithms, are detailed within this segment.

3.2. Cross-Analysis of Number of Vehicles and Customers

This section performs cross-analysis with a different number of vehicles (2, 3, and 5) and customer sizes (9, 25, 50, and 100). The analysis is based on the performance of four hybrid algorithms (GA-SA, GA-LS, PSO-SA, and PSO-LS) concerning computation time, distance, fitness, and stability, according to Table 2-5.

1. Computation Time (ct): It can be observed that there is a significant growth in computation time with an increase in both the number of vehicles and the number of customers. For example, with a problem size of 9 customers, PSO-SA has the best computation time of 0.038 seconds with 2 vehicles, as shown in Table 2. This is contrary to what GA-SA has, which is 0.336 seconds. Despite increasing its customers to 100, PSO-SA still maintains a competitive edge in computation time at 5.124 seconds, compared to GA-SA at 10.109 seconds in Table 5. This tendency supports PSO-SA's superiority regarding larger problem sizes. All algorithms are affected by the growing problem size; an increasing computational burden is required while still maintaining the edge in speed, with PSO-SA as the fastest algorithm.
2. Distance: As with other heuristics, LS-based heuristics perform better regarding the total distance traveled when "local search" is included in the algorithm, especially in PSO-LS and GA-LS. For instance, in the case of 50 customers, PSO-LS achieved a distance of 1167, while PSO-SA (1383) and GA-SA (1310) had worse outcomes. This proves that LS-based algorithms are more efficient in refining the solution concerning the distance metric, mostly in larger problem sizes. With the increase in customers, PSO-LS is dominant in distance performance, indicating that the algorithm effectively minimizes total travel distance even when faced with complex problems.
3. Fitness: These fitness scores align with the distance results as there is consistency, with LS achieving the highest fitness value in all problem sizes. For 100 customers, as indicated in Table 5, PSO-LS achieved fitness scores of 5.206, which is better than the GA-SA and even PSO-SA. This indicates that the algorithm not only minimizes the distance but does so while achieving optimal solutions across various problem sizes. The distributed performance corroborates the previous findings that LS-based algorithms are more efficient in providing better solutions as the problem size increases.
4. Standard Deviation (std) and Variance: The consideration of std and variance suggests that algorithms incorporating LS, such as PSO-LS and GA-LS, are more stable. For instance, as presented in Table 5 with 100 customers, the standard deviation of PSO-LS is lower than that of PSO-SA, with a value of 0.405 compared to 1.862. This means that PSO-LS provides more consistent results than PSO-SA. Hence, it is more accurate for large-scale problem instances, which is preferred. On the other hand, GA-SA and PSO-SA tend to exhibit greater variance and standard deviation, implying that these algorithms might have issues with consistency, particularly when dealing with larger datasets.

Table 2. Experimental Results of GA-SA, GA-LS, PSO-SA, and PSO-LS with 9 Customers

	ct			Distance			fitness			iteration converge			std			variance		
	V2	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**
GA-SA	0.336	0.355	0.327	72	91	113	82	91	138	6	6	5	3.447	4.329	3.474	11.884	18.743	0.336
GA-LS	0.510	0.371	0.365	72	88	103	82	88	128	9	6	5	3.500	3.749	4.116	12.248	14.056	0.510
PSO-SA	0.038	0.038	0.048	69	87	111	79	87	136	5	5	5	3.645	3.897	3.515	13.289	15.188	0.038
PSO-LS	0.063	0.064	0.077	73	93	104	83	93	129	5	5	5	3.310	3.631	3.977	10.959	13.188	0.063

Table 3. Experimental Results of GA-SA, GA-LS, PSO-SA, and PSO-LS with 25 Customers

	ct			distance			fitness			iteration converge			std			variance		
	V2	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**
GA-SA	0.106	0.071	0.076	202	266	332	212	281	332	5	5	5	6.215	7.813	7.945	38.620	61.036	0.106
GA-LS	0.084	0.086	0.075	198	241	289	208	256	289	5	5	5	5.484	7.133	7.863	30.074	50.881	0.084
PSO-SA	1.582	0.561	2.990	226	270	262	236	285	262	9	6	36	6.068	7.197	6.875	36.826	51.801	1.582
PSO-LS	1.041	0.708	0.509	242	216	271	252	231	271	6	6	6	7.824	6.518	8.187	61.221	42.490	1.041

Table 4. Experimental Results of GA-SA, GA-LS, PSO-SA, and PSO-LS with 50 Customers

	ct			distance			fitness			iteration converge			std			variance		
	V2	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**
GA-SA	0.109	0.125	0.138	568	589	731	568	694	731	5	5	5	7.703	7.883	7.703	59.340	62.138	0.109
GA-LS	0.088	0.133	0.233	546	424	449	546	424	449	5	6	5	7.878	6.977	6.039	62.058	48.682	0.088
PSO-SA	1.731	2.008	1.230	567	557	690	587	572	690	7	10	7	7.887	7.540	8.250	62.202	56.854	1.731
PSO-LS	1.503	1.248	1.085	517	449	438	517	464	438	6	6	6	7.838	7.472	5.768	61.439	55.834	1.503

Table 5. Experimental Results of GA-SA, GA-LS, PSO-SA, and PSO-LS with 100 Customers

	ct			distance			fitness			iteration converge			std			variance		
	V2	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**	**V2**	**V3**	**V5**
GA-SA	0.232	0.263	0.301	1167	1112	1415	1167	1127	1515	5	5	5	8.641	8.171	8.948	74.658	66.764	0.232
GA-LS	0.202	0.134	0.240	1202	1366	1445	1202	1381	1445	5	5	5	8.093	8.824	8.817	65.502	77.863	0.202
PSO-SA	5.124	3.525	7.334	1183	1237	1363	1183	1252	1363	6	6	16	7.944	7.878	8.911	63.103	62.068	5.124
PSO-LS	5.206	3.474	2.862	1332	1279	1383	1332	1294	1383	6	6	6	8.509	8.642	7.833	72.408	74.690	5.206

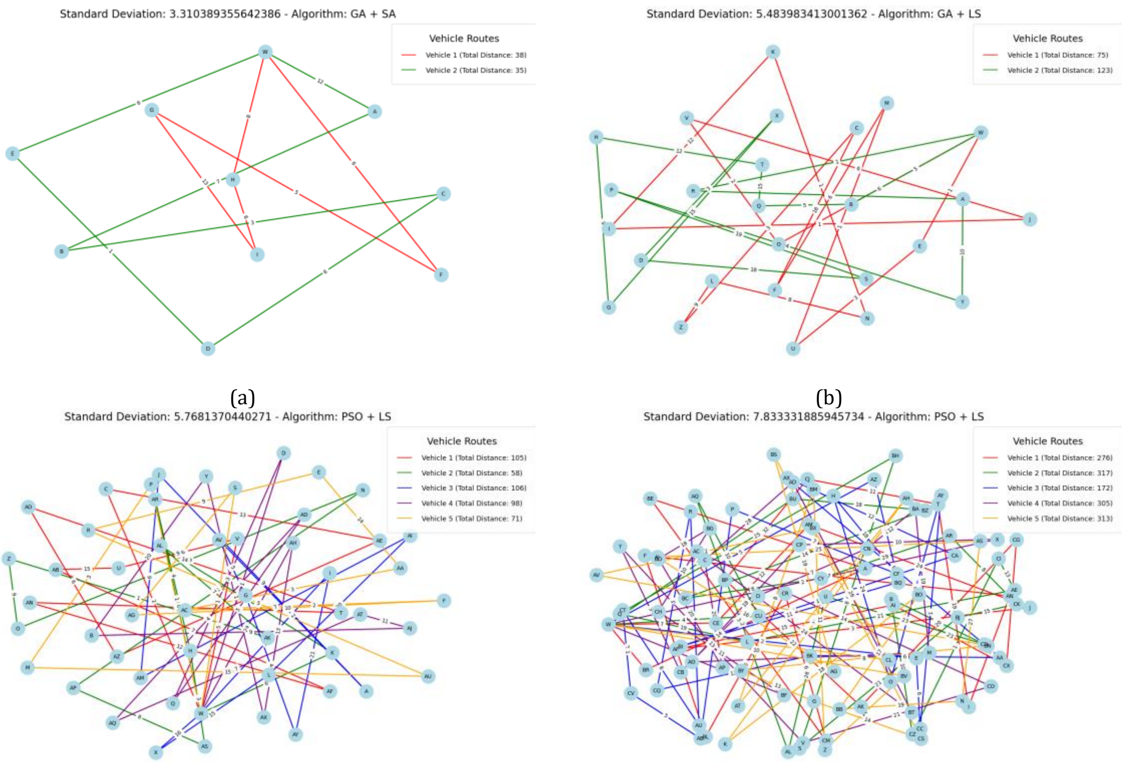
3.3. *Discussion and Implications*

Hybrid optimization approaches are, therefore, illustrated in the vehicle routes outlined in Figures 3(a) through 3(d), while the particular complexities of the problems are reflected in Tables 2 through 5. Problem instances with nine customers, as shown in Figure 3(a), have very low complexity, allowing approaches such as PSO-LS to perform effectively in distributing routes with a low standard deviation of 3.31. The routes are pretty straightforward, with moderate overlaps, indicating the efficacy of the PSO-LS on relatively simple problems. Although GA-SA worked, it had a slightly higher deviation, suggesting that probing minor problems may be limited in its search power.

As indicated in Figure 3(b), with growing complexity as indicated in Table 3, we expanded the customer base to 25. GA-LS emerges as an important algorithm since it skillfully considers the trade-off between the time it takes to reach the solution and the quality of that solution, and the standard deviation of 5.48 is commendable. The routes exhibit moderate intersection, indicating the increasing complexity involved in the loading-distributing procedures. PSO-LS remains at a reasonable pace but with slight deviations, showing itself thirsty for larger problem sizes.

Under 50 customers, as in Figure 3(c), presents a different picture as the determination of efficient routes gets even tougher. The results in Table 4 show that GA-LS is still encrypted in managing such problems, returning a low standard deviation of 5.76. PSO-LS, exhibiting the capability of minimizing total travel distance, still shows increased variability due to increased computational demands. The routes in these figures are more entangled, underlining that solution spaces get complex with increasing customers.

An instance of the typical problems with large-scale instances, Figure 3(d) shows the difficulties with 100 customers. PSO-LS may land decent fitness values; however, with an unstable standard deviation of 7.83, it is the highest in this scenario, as in Table 5. The packed and crossing trips of routes depicted by this figure show the problematic part of being efficient at such a scale. Compared to that, GA-LS delivers more or less stable performance, although at the cost of being less efficient. This emphasizes the need to select optimization methods compatible with the particular nature of the problem at hand.



(c) (d)

Figure 3. (a) Vehicle Route for Number Customers 9; (b) Vehicle Route for Number Customers 25; (c) Vehicle Route for Number Customers 50; (d) Vehicle Route for Number Customers 100

Optimization in Larger Problem Sizes

We obtained several findings from the results of our analysis using Hybrid Metaheuristic algorithms to solve the Vehicle Routing Problem (VRP). One key observation is that while PSO-SA offers the fastest computation times, it does not necessarily yield the best quality solutions, particularly in minimizing the distance or achieving high fitness values. This is shown in Figure 3(a), where PSO-SA's vehicle routes (red and green) have moderate overlaps with a standard deviation of 3.31 for 9 customers, indicating its efficiency in handling more minor problems. However, the quality of the solution may be compromised as the problem size increases.

In contrast, including Local Search (LS), as seen in PSO-LS and GA-LS, significantly improves the solution quality, especially for larger instances. As demonstrated in Figure 3(b), with 25 customers, PSO-LS and GA-LS provide more refined solutions with lower variance in their respective vehicle routes, indicating their ability to optimize distance more effectively. These algorithms are particularly well-suited for larger VRP instances (i.e., with more vehicles and customers), where the balance between global search and local refinement is crucial for achieving better outcomes. Thus, hybrid algorithms that combine global search methods like PSO or GA with local refinement techniques such as LS are recommended for large-scale VRP problems, as they tend to deliver superior solutions. These algorithms are especially effective when solution quality and stability are paramount, even if they require slightly more computational time.

Computation Time vs. Solution Quality

The study also highlights a trade-off between computation time and solution quality. As Figure 3(a) shows, PSO-SA consistently demonstrated faster computation times for 9 customers but achieved less optimal fitness scores and higher variance in the solution, particularly in the routes (represented by red and green lines). The total distances for each vehicle in PSO-SA (Vehicle 1: 38, Vehicle 2: 35) were higher than those of PSO-LS and GA-LS. This suggests that PSO-SA may be preferred in real-time or dynamic routing applications, where computation time is critical.

However, for applications where optimality and reliability are of greater concern—such as in strategic planning or long-term optimization—algorithms like PSO-LS and GA-LS, as seen in Figures 3(b) and 3(c), are more suitable. Despite their higher computation times, these algorithms deliver better-quality solutions with lower variance in their results. For instance, PSO-LS in Figure 3(c) minimizes travel distances effectively, as shown in the vehicle routes (total distance for Vehicle 1: 105, Vehicle 2: 58).

Stability and Consistency

The lower standard deviation and variance for PSO-LS and GA-LS indicate that these algorithms provide better-quality solutions and are more stable and consistent. In Figure 3(d), for 100 customers, PSO-LS and GA-LS exhibit significantly lower standard deviations (PSO-LS: 0.405 vs. PSO-SA: 1.862), indicating better stability and fewer fluctuations in the vehicle routes, even as the problem size grows.

This consistent performance is critical in operational settings, where predictability and reliability are important. As demonstrated in Figure 3(d), high variability in PSO-SA and GA-SA suggests that these algorithms may require more iterations to stabilize and provide consistent results. The higher standard deviation and variance observed in these algorithms imply that their solutions are more prone to fluctuations across different runs, which may not be ideal in environments that demand reliable and stable outcomes.

In practical terms, these findings suggest that the choice of algorithm depends on the specific requirements of the VRP application. PSO-SA may be the preferred choice when computational efficiency is prioritized, such as in real-time applications, as it provides faster solutions. However, when solution quality, stability, and consistency are more important—such as in strategic logistics planning or large-scale optimization—PSO-LS and GA-LS are the more appropriate choices due to their superior fitness scores, lower variance, and greater stability.

3.4. Result Overview

Solution quality in relationship to computational efficiency is one of the considerations in solving Vehicle Routing Problems (VRP), especially while integrating hybrid optimization methods: Particle Swarm Optimization combined with Local Search (PSO-LS), Particle Swarm Optimization combined with Simulated Annealing (PSO-SA), Genetic Algorithm combined with Local Search (GA-LS), and Genetic Algorithm combined with Simulated Annealing (GA-SA). This section presents results drawn from tests based on varying numbers of vehicles and customers in specific algorithm configurations, constitutes how these parameters affect solution quality and computational time. The insights from this analysis illustrate the trade-offs involved and guide the parameter selection for use in logistics and transportation.

As it is present in the analysis depicting an ongoing trade-off between solution quality and its computational efficiency, increasing the number of users for a particular algorithm, defined through its parameters and structures, and subsequently examining the results, reveals scalability issues at focus in Table 6, which specifies not just the algorithm structure but also hybrid optimization techniques.

Table 6. Comparison of Proposed Hybrid Algorithm

Metric	GA-SA	GA-LS	PSO-SA	PSO-LS
Computation Time	High	Low	High	Moderate
Fitness	High	Moderate	Moderate	High
Iteration Converge	Slow (500-1000)	Moderate (300-400)	Slow (500-1000)	Fast (100-200)
Stability (Std)	High	Low	High	Moderate

The analysis presents a variety of key observations relating to the performance of these hybrid techniques. First, hybrid approaches that use LS are likely to perform better, given LS's capability to achieve local refinements in a smaller solution space, thus allowing for a better exploration of possible solutions. In this way, SA methods tend to perform differently depending on their cooling rates. Although the SA methods may converge relatively slowly, they save themselves from local minima--an advantage that may prove substantial in mathematical programming.

In computational time domains, GA-based methods are generally less tedious on computational resources because GA applies purely mutation and crossover operations compared to the other swarm-based models, which are plenty taxing. The major hit on SA-based methods is the cost incurred because of extensive iterations during refinement, where the scouting moves more slowly, gaining more temperature as larger sets are explored in the solution space.

Stability is recommended as a comparison factor for these methods. PSO-based methods are variable, reflecting a dynamic trade-off between exploration and exploitation. On the other hand, GA-based techniques yield consistent results in lower standard deviation, hence are most suited in applications where reliability matters.

4. CONCLUSION

This study focuses on the application of hybrid strategies using Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA), and Local Search (LS) to solve the Vehicle Routing Problem (VRP). Along with other algorithms, this problem was diagnosed with metaheuristic techniques. The experiments found that combinations of PSO-LS and GA-LS performed better than PSO-SA and GA-SA. These two combinations were also more efficient in smaller to mid-sized problems. Concerning computation time, total distance, and overall solution quality, hybrid algorithms built on Local Search catalyzed had lower standard deviations than PSO-LS and GA-LS performed, meaning they were more stable for customers from 9-50. Simulated Annealing SA-based algorithms had faster computation times but offered higher variability, making them less robust when exposed to larger problem sizes.

This work underlined the importance of selecting a suitable optimization approach based on the problem. Focusing on VRP, PSO-LS, and GA-LS were suggested for problems that seek the best solutions but compromise efficiency; for problems where computation speed is needed, SA-PSO could be leveraged, but at the cost of solution quality. This study also sheds light on the balance between the time taken to compute and the quality of solutions offered on VRP problems, emphasizing the impact of hybridization in algorithms to exploit the best features of different approaches for optimal outcomes.

ACKNOWLEDGEMENTS

The Informatics Department, Faculty of Science and Technology of Universitas Islam Neger (UIN) Sunan Kalijaga Yogyakarta, funded the research. We have further expressed our appreciation to our colleagues in the Faculty of Science and Technology for their constructive criticism and suggestions that improved our approach to a great extent. We greatly appreciate their assistance, but the insights and conclusions contained in this paper are solely the authors.

REFERENCES

[1] A. Seyyedabbasi, W. Z. Tareq Tareq, and N. Bacanin, "An Effective Hybrid Metaheuristic Algorithm for Solving Global Optimization Algorithms," *Multimed. Tools Appl.*, vol. 83, no. 37, pp. 85103–85138, May 2024, doi: 10.1007/s11042-024-19437-9.

[2] A. Bogyrbayeva, M. Meraliyev, T. Mustakhov, and B. Dauletbayev, "Machine Learning to Solve Vehicle Routing Problems: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 4754–4772, 2024, doi: 10.1109/TITS.2023.3334976.

[3] G. B. Dantzig and J. H. Ramser, "The Truck Dispatching Problem," *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, 1959.

[4] M. Drexl, "Synchronization in Vehicle Routing—A Survey of VRPs with Multiple Synchronization Constraints," *Transportation Science*, vol. 46, no. 3, pp. 297–316, Aug. 2012, doi: 10.1287/trsc.1110.0400.

[5] S. F. Roselli, M. Fabian, and K. Åkesson, "Conflict-free electric vehicle routing problem: an improved compositional algorithm," *Discret. Event Dyn. Syst.*, vol. 34, no. 1, pp. 21–51, Mar. 2024, doi: 10.1007/s10626-023-00388-6.

[6] M. Oliveira Machado, E. F. Gouvea Goldbarg, M. Cesar Goldbarg, G. De Araujo Sabry, I. F. Costa Fernandes, and T. Soares Marques, "Heuristic Hybridization for CaRSP, a multilevel decision problem," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, Dec. 2021, pp. 01–08. doi: 10.1109/SSCI50451.2021.9660015.

[7] S. Kaya, "A hybrid firefly and particle swarm optimization algorithm with local search for the problem of municipal solid waste collection: a real-life example," *Neural Comput. Appl.*, vol. 35, no. 9, pp. 7107–7124, 2023, doi: 10.1007/s00521-022-08173-6.

[8] M. U. Siregar *et al.*, *Housing Price Prediction Using a Hybrid Genetic Algorithm with Extreme Gradient Boosting*, vol. 1, no. 1. Association for Computing Machinery, 2022. doi: 10.1145/3575882.3575939.

[9] M. U. Siregar, "Optimized Random Forest Classifier Based on Genetic Algorithm for Heart Failure Prediction," *The 2022 International Conference on Intelligent Computing (ICIC 2022)*, vol. 1, no. 1, 2022.

[10] Y. Riwanto, M. T. Nuruzzaman, S. Uyun, and B. Sugiantoro, "Data Search Process Optimization using Brute Force and Genetic Algorithm Hybrid Method," *IJID (International Journal on Informatics for Development)*, vol. 11, no. 2, pp. 222–231, 2023, doi: 10.14421/ijid.2022.3743.

[11] E. Ranolo *et al.*, "A Hybrid Approach to Automatic Timetabling Using Self-Organizing Maps, Secure Convex Dominating Sets, and Metaheuristics," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 8, pp. 50–58, 2025, doi: 10.14569/IJACSA.2025.0160805.

[12] L. Cai, "Decision-making of transportation vehicle routing based on particle swarm optimization algorithm in logistics distribution management," *Cluster Comput.*, vol. 26, no. 6, pp. 3707–3718, 2023, doi: 10.1007/s10586-022-03730-z.

[13] J. K. C. Revanna and N. Y. B. Al-Nakash, "Metaheuristic link prediction (MLP) using AI based ACO-GA optimization model for solving vehicle routing problem," *International Journal of Information Technology (Singapore)*, vol. 15, no. 7, pp. 3425–3439, 2023, doi: 10.1007/s41870-023-01378-5.

[14] H. Saleh, M. Sayad, Y. Almoghathawi, A. Alghazi, and K. Al-Shareef, "A drone-based logistics network for blood supplies: a genetic algorithm based on greedy search," *Soft comput.*, vol. 2, 2024, doi: 10.1007/s00500-024-10373-2.

[15] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, *Metaheuristic algorithms: A comprehensive review*. Elsevier Inc., 2018. doi: 10.1016/B978-0-12-813314-9.00010-4.

[16] G. Erdoğan, "An open source Spreadsheet Solver for Vehicle Routing Problems," *Comput. Oper. Res.*, vol. 84, pp. 62–72, Aug. 2017, doi: 10.1016/j.cor.2017.02.022.

[17] R. Elshaer and H. Awad, "A taxonomic review of metaheuristic algorithms for solving the vehicle routing problem and its variants," *Comput. Ind. Eng.*, vol. 140, p. 106242, Feb. 2020, doi: 10.1016/j.cie.2019.106242.

[18] C. Prins, "A simple and effective evolutionary algorithm for the vehicle routing problem," *Comput. Oper. Res.*, vol. 31, no. 12, pp. 1985–2002, 2004, doi: 10.1016/S0305-0548(03)00158-8.

[19] R. F. Syahputra and Yahfizham, "Menganalisis Konsep Dasar Algoritma Genetika," *Bhinneka: Jurnal Bintang Pendidikan dan Bahasa*, vol. 2, no. 1, pp. 120–132, Dec. 2024, doi: 10.59024/bhinneka.v2i1.643.

[20] A. Lambora, K. Gupta, and K. Chopra, "Genetic Algorithm- A Literature Review," in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, IEEE, Feb. 2019, pp. 380–384. doi: 10.1109/COMITCon.2019.8862255.

[21] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 22–37, 1996, doi: 10.1109/79.543973.

[22] F. Zhang, Y. Xie, and L. Zheng, "Application and Efficiency Analysis of Genetic Algorithm in Multi-Vehicle Path Optimisation Models Under Time Window Constraints," in *2024 IEEE 7th International Conference on Information Systems and Computer Aided Education (ICISCAE)*, IEEE, Sep. 2024, pp. 938–944. doi: 10.1109/ICISCAE62304.2024.10761568.

[23] A. A. Mousa, M. A. El-Shorbagy, and W. F. Abd-El-Wahed, "Local search based hybrid particle swarm optimization algorithm for multiobjective optimization," *Swarm Evol. Comput.*, vol. 3, pp. 1–14, 2012, doi: 10.1016/j.swevo.2011.11.005.

[24] E. Mirsadeghi and S. Khodayifar, "Hybridizing particle swarm optimization with simulated annealing and differential evolution," *Cluster Comput.*, vol. 24, no. 2, pp. 1135–1163, Jun. 2021, doi: 10.1007/s10586-020-03179-y.

[25] F. Javidrad and M. Nazari, "A new hybrid particle swarm and simulated annealing stochastic optimization method," *Appl. Soft Comput.*, vol. 60, pp. 634–654, Nov. 2017, doi: 10.1016/j.asoc.2017.07.023.

[26] B. Fu, Y. He, Q. Guo, and J. Zhang, "An improved competitive particle swarm optimization algorithm based on de-heterogeneous information," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 6, 2023, doi: 10.1016/j.jksuci.2022.12.012.

[27] L. Vanneschi and S. Silva, "Particle Swarm Optimization," L. Vanneschi and S. Silva, Eds., Cham: Springer International Publishing, 2023, pp. 105–111. doi: 10.1007/978-3-031-17922-8_4.

[28] M. Tanha, M. H. Shirvani, and A. M. Rahmani, *A hybrid meta-heuristic task scheduling algorithm based on genetic and thermodynamic simulated annealing algorithms in cloud computing environments*, vol. 33, no. 24. Springer London, 2021. doi: 10.1007/s00521-021-06289-9.

[29] H. L. Shieh, C. C. Kuo, and C. M. Chiang, "Modified particle swarm optimization algorithm with simulated annealing behavior and its numerical verification," *Appl. Math. Comput.*, vol. 218, no. 8, pp. 4365–4383, 2011, doi: 10.1016/j.amc.2011.10.012.

[30] W. Ben-Ameur, "Computing the Initial Temperature of Simulated Annealing," *Comput. Optim. Appl.*, vol. 29, no. 3, pp. 369–385, Dec. 2004, doi: 10.1023/B:COAP.0000044187.23143.bd.

[31] J. J. Schneider and M. Puchta, "Investigation of acceptance simulated annealing — A simplified approach to adaptive cooling schedules," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 24, pp. 5822–5831, Dec. 2010, doi: 10.1016/j.physa.2010.08.045.

[32] B. Hajek, "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research*, vol. 13, no. 2, pp. 311–329, May 1988, doi: 10.1287/moor.13.2.311.

[33] Y. Dong-mei, "A Hybrid Algorithm of Simulated Annealing and Particle Swarm Optimization," *Computer Simulation*, 2008.

[34] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Applied Soft Computing Journal*, vol. 43, pp. 117–130, 2016, doi: 10.1016/j.asoc.2016.01.044.

[35] F. Uddin *et al.*, "An Improvement to the 2-Opt Heuristic Algorithm for Approximation of Optimal TSP Tour," *Applied Sciences*, vol. 13, no. 12, p. 7339, Jun. 2023, doi: 10.3390/app13127339.

[36] S. B. Sarathi Barma, J. Dutta, and A. Mukherjee, "A 2-opt guided discrete antlion optimization algorithm for multi-depot vehicle routing problem," *Decision Making: Applications in Management and Engineering*, vol. 2, no. 2, Oct. 2019, doi: 10.31181/dmame1902089b.

[37] P. Aivaliotis-Apostolopoulos and D. Loukidis, "Swarming genetic algorithm: A nested fully coupled hybrid of genetic algorithm and particle swarm optimization," *PLoS One*, vol. 17, no. 9, pp. 1–24, 2022, doi: 10.1371/journal.pone.0275094.

[38] S. Yin and H. Li, "GSAPSO-MQC:medical image encryption based on genetic simulated annealing particle swarm optimization and modified quantum chaos system," *Evol. Intell.*, vol. 14, no. 4, pp. 1817–1829, 2021, doi: 10.1007/s12065-020-00440-6.