VAWKUM Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2024, ISSN(e): 2308-8168, ISSN(p): 2411-6335
Volume 12, Number 2, July-December 2024 pp:01-15

A Hybrid Approach for Simultaneous Effective Automobile Navigation with DE and PSO

Syed Muhammad Daniyal [1*], **Mohsin Mubeen Abbasi** [1], **Dilbar Hussain** [1], **Usama Amjad** [1], **Abdul Basit Abro** [2], **Mehwish Naeem** [3]

[1]Faculty of Engineering Science and Technology, Iqra University Karachi, Pakistan; [2]Department of Computer System Engineering, Mehran University, Jamshoro, Pakistan; [3]School Polytechnic De Bruxelles, Brussels, Belgium

Keywords: *Automobile Navigation, Local Search, Differential Evolution, PSO.*

Journal Info:
Submitted:
September 05, 2024,
Accepted:
October 01 , 2022
Published:
October 06, 2024,

Abstract Automobile Navigation is fundamentally an optimization challenge focused on transportation logistics between a depot and various clients. In this paper, we specifically address the complex variant of Automobile Navigation that involves simultaneous pickup and delivery tasks, which must be executed concurrently at clients' locations. This dual requirement introduces significant complexity, as traditional exact approaches struggle to rapidly identify near-optimal solutions due to the problem's NP-hardness. Therefore, the objective of this research is to develop a novel hybrid algorithm that integrates Differential Evolution (DE) and Particle Swarm Optimization (PSO) to effectively solve the Automobile Navigation problem with simultaneous pickup and delivery. The proposed method uses the nearest neighbor heuristic to initially produce results. It is based on the iterated local search paradigm. Variable neighborhood descent is used to improve the search process by adding random sequences to the neighborhood structures for improved search space intensification. Furthermore, exploration across various sections of the search space is made possible by the perturbation process. This method solves the problem of different truck loads on every client visit because it does the pickup and delivery at the same time, which makes the Navigation strategy more effective.

***Correspondence author email address:** syed.daniyal@iqra.edu.pk
DOI: 10.21015/vtcs.v12i2.1914

1 Introduction

Route optimization is the process of finding the best (shortest and cheapest) route for Automobiles in a such way that we can optimize and find optimal solutions. A route optimization tool can find the most efficient routes to travel based on the constraints and goals of a business. This includes taking into account the number of stops required, their locations, and the time windows for deliveries. But it is not that simple. If one has a business, one may know that route planning includes deciding on the best way to get customers to the business. One might have had difficulty following the proper route. There should be challenges to face if one is not likely to take them more seriously. We are used to requesting multiple pick-ups and drop-off requests for our packages because we want them to be delivered to any location. Also, handling multiple clients may increase workload. Managing all these tasks one-handed can lead to serious bugs and problems. A carpenter cannot do the job of an electrician. People who are skilled at performing tasks quickly can do so effectively and with less time spent on the task. Likewise, Automobiles are also made for specific functions. Reefers are designed to carry cold items, trucks are made for carrying heavy items, and so on. Therefore, it is important to choose the right Automobile for the job. It is important to find a driver who is communicative and able to complete the process easily. The service time includes the travel time and a fixed stop time per customer. There is no one answer to this question since it depends on the individual's needs. It is difficult to determine which route and at what time a driver delivers the products. If an average time doesn't seem to be working, there might be a time crisis looming. Accounting is essential for ensuring that all financial matters are taken care of properly. It can be difficult to gather all of the details necessary for a report. Accounting is important for keeping track of the costs associated with business. If one neglects this part of business, it could bring the company down. Someone may have assets with their limitations. The volume, weight, and load meter are a part of the limitations of these devices. It is easy to manage the limitations of a single Automobile when doing so. However, when the number changes, the task becomes very tiring and difficult. We often change the delivery destination at the last minute. This can be a serious problem for logistics companies. Finding an optimized route at the last minute can be quite difficult. Moreover, there is a high chance of customer dissatisfaction if one fails to deliver goods on time. So, how can one overcome all these problems? All the above problems will be managed with the help of a Navigation optimization system.

1.1 Optimization

Maximizing or minimizing some function relative to some set, often representing a range of choices available in a certain situation. The function allows comparison of the different choices for determining which might be the best. Some Common goals of Optimization are minimal cost, maximal profit, minimal error, optimal design, and optimal management. Usually, optimization is of two types:

- Mathematical Optimization: Mathematical optimization is the process of finding the best possible solution to a problem by adjusting variables in a way that makes the problem as efficient as possible. Some variation of optimization is required for all deep learning models to function, whether using supervised or unsupervised learning. There are many optimization techniques to choose from, but all require a starting point and a goal. To optimize something, one needneedsdefine an objective function that will tell one what one wantwantsachieve. This function can produce a specific result or a probability threshold. Inputs can be either discrete or continuous.
- Heuristic Optimization: Basically, every supply chain planning and scheduling problem is at its core an optimization problem. The company's solution involves determining the best way to synchronize the supply and demand across the supply chain network to boost customer satisfaction

and bottom-line results. One common way businesses try to solve their supply chain planning and scheduling problems is by using heuristics. Simply put, a heuristic is a problem solution that uses a practical process (often referred to as a rule of thumb or best practice) to produce a workable solution good enough to quickly solve a specific problem and immediate goals - but not necessarily an optimal solution. In contrast, an optimization model uses an intelligent, automated process to create an optimal solution to a specific problem, given decision variables such as production, inventory, and shipment volumes, as well as constraints and key performance indicators (KPIs). Supply chain optimization solutions aim to improve the performance of procurement, production, inventory, and distribution operations so that one can achieve the best possible delivery performance and overall profitability

1.2 Route Optimization System

If one runs a logistics company, then one knows that the destination may not always be where one plans to take products. Even if it is not a pickup point, the delivery point can be multiple. How can one find the shortest route that covers all of the points? How can one know about the road conditions remotely? To optimize a route, one will need a route optimization system. Delivery route optimization software can help one find the shortest distance between two or more locations. Truck Navigation software can also analyze current dicey situations and business limitations, like the available Automobiles and drivers, traffic conditions, etc. This will help one get goods to their destination promptly. The best route optimization software generally follows three primary strategies: optimizing routes based on available data, optimizing routes based on user preferences, and optimizing routes to minimize travel time. The three types of Navigation are:

- Static- which is based on requirements such as quarterly or monthly plans.
- Dynamic- which is based on daily developments.
- Real-time dynamic Navigation – Developing routes based on current road and traffic conditions in real-time to help drivers on the ground.

1.3 Challenges

The delivery industry is growing massively as the new "online" trend has started due to the Covid. Most of the businesses have now shifted to "online" platforms. This increased the demand for the packages to be delivered. Therefore, finding an optimized route for delivering the packages has become more critical and harder than ever for delivery businesses. But today almost every business uses route optimization software from small cottage industries and stores to large B2B enterprises of all industries. In the delivery system, when done manually, finding the best route is nearly impossible. Even if we have a few Automobiles with 10 stops each there can be a million routes. While planning a route is very hard having hundreds of different routes without having the right tools. Also, the biggest challenge with last-mile delivery includes Delivery efficiency, margins, customer demands delivery agility, costs, end customer interactions, missed deliveries, etc. Companies also risk inflating their operational costs, often in the form of too many Automobiles in their fleet and/or wasted fuel and wages due to longer than necessary routes. Delivery businesses face these types of problems every day. Effective route planning would save fuel, time, cost, employee expenses, transport expenses, and maintenance expenses. So, if we use the proper route optimization mechanism we can overcome and save all the above things which will help businesses generate more revenue and let them make profitable. An effective route optimization solution will help delivery businesses minimize wages, driving time, and fuel consumption by finding the most efficient route for the entire fleet in a matter of minutes. Thus, we will formulate an algorithm that would optimize the route for a delivery company, hence minimizing the operational costs incurred by the company and the time taken to deliver a certain number of parcels. Nowadays many companies like Food Panda, UberEATS, Groceries, bottled beverages, CSA farm, and others are using software that helps them in planning their route for deliveries. These companies having an optimized route can create a valuable effect on the cost and time taken by a package to be delivered or collected, thus minimizing the operational cost for the delivery companies. As we are targeting optimization in route planning. Our main target is minimizing traveling time. To minimize traveling time by using optimization of all the routes and riders we will design an algorithm in such a way that when we try to plan a route our optimization system selects the best rider from all available riders in a particular area and assign deliveries to them. The major challenging part here is the data collection. To carry out experiments on our formulated algorithm, we need some real-time data. Getting this real-time data is very difficult as no company will be willing to provide its data due to security reasons. Our ultimate goal is to minimize the delivery time by minimizing the operational costs incurred in terms of travel time and fuel costs due to longer or ineffective routes or even due to an inexperienced ride.

1.4 Objectives

- To collect data related to Automobile Navigation i.e., Automobile Navigation Problem, that would contain the information regarding delivery and depot locations, the rider and delivery Automobile capacity, and other customer-related data.
- To apply different clustering techniques to divide the customers or the delivery locations according to their delivery needs or their locations or the like features.
- To minimize time by formulating an algorithm that would suggest the best possible route given the locations for delivery and the capacity of the delivery entity (i.e., the delivery van or the delivery boy).

2 RELATED WORK

Some scientists around the world have carried out a lot of work but the problem remains a critical subject in the field of optimization, especially when it comes to multidimensional optimization. This section reviews the major approaches and methodologies used earlier to solve this problem. This section reviews the major approaches and methodologies used earlier to solve this problem.

2.1 Heuristic Algorithms

There are two major categories of heuristic algorithms being widely used to solve route optimization i.e., Local Search and Evolutionary Search.

- Local Search: Starting from an initial solution, Local search moves from a current solution to another solution in the neighborhood. This is an iterative process in which the solution starts from a candidate node and exchanges nodes or local routes to gradually move towards a better solution. This iterative process sometimes gets trapped in the local optimum solution; therefore, to overcome this many other intelligent strategies have been formulated to improve overall solution quality. These include simulated annealing [1], iterated local search [2], large, variable neighborhood search [2] [3], and tabu search [4]. Many experiments on small and large-sized Automobile Navigations (i.e., for 25 - 100 customers), using Local Search Heuristics are found to have performed well.
- Evolutionary Search: In an Evolutionary Search the whole solution space is divided into many small solutions, and then the evolutionary algorithm concurrently optimizes many solutions even-

tually reaching high-quality solutions. It has four major steps involved: Representation, Selection, Combination, and Mutation A few most effective evolutionary algorithms for Automobile Navigation optimization are provided by [5]. Another author [6] used the standard evolutionary optimization framework to guide exploration in the Automobile NavigationTW solution space with solution initialization and evolution. They [7] parallelized the genetic algorithm and were the first to solve Automobile NavigationTW instances with up to 1000 customers. The only work on the Voronoi Diagram was done by Milthers [8], who split the Automobile NavigationTW into subproblems and then solved them with large neighborhood search heuristics. The Voronoi diagram was found to be effective in guiding the search process. However, this study only scoped the decomposition of the problem in the solution construction stage. It can be further improved.

2.2 Integration of VRO

A few effective approaches for Automobile Navigation have been integrated with GIS-based SDSS to cater to some real-world applications. Spatial data management, processing, and visualization tools are used to collect customer orders, and georeference-related data, activate the solving process, and display routes for VROs, as in GIS software such as ArcGIS and Trans CAD. Along with Local search, [9] was first introduced to the implementation of a tabu search heuristics approach in a GIS environment to deal with VRO for an American retailer. Another author [10] integrated a customized Navigation module, which improved solution quality with commercial solutions such as SAP/R3 and ArcGIS to cope with VROs in public utilities. Experiments on a real-world case in Bogotá, Colombia with 323–601 customers verified the effectiveness of evolutionary optimization and GIS software [11], developed a web-based user-friendly SDSS embedded with VRO to cater to a trash collection task in Coimbra [12], [13] used historical traffic information to present a cloud GIS-based spatial decision support framework with variable neighborhood search heuristics for dynamic Automobile Navigation. All this shows the dominance of VRO in real-life transportation applications. However, they also indicated that current spatial intelligence should be improved to cater to the ever-increasing number of customers in the different transportation sectors. The classical variant of the Automobile Navigation has limited capacity on their Automobiles. The goal of Automobile Navigation is to find the optimal routes that visit all customers at once, respecting the capacity of each Automobile, with routes starting and ending at the depot. The classical optimization task is to minimize the overall distance. For the full model, consider the work of [14]. Some mathematical formulations-based methods will thoroughly search for the values of the variables to find the mathematically proven optimal. However, the problem was proven to be NP-hard [15], and finding the best routes is a hard task. For instances with more than 200 customers, the problem is usually referred to as Large-Scale Automobile Navigation (LSAutomobile Navigation) [16]. This larger size requires extra care regarding the amount of search effort given. Therefore, such a heuristic method is needed to find good solutions in a faster time instead of an optimal solution. Heuristics are usually based on simple moves that can be searched fast and repeatedly. These are known as neighborhoods, perturbation heuristics, and local search, and they can be classified as intra-route and inter-route. Examples of such moves are the classics 2-Opt, and Cross Exchange, among others [17]. The most effective and efficient methods for solving the Automobile Navigation use a Local Search (LS) based approach [18]. LS heuristics quality is defined by the neighborhoods utilized, ranging from classic and simple moves to more elaborated ones. To start, the LS explores new solutions by making small moves, being robust across different problems and instances, as well as being able to find high-level solutions [19].

Work from [20] applied LS in combination with other techniques, such as Genetic Algorithm and Set

Partitioning, respectively, to lead the solution or set of solutions towards better solutions. However, when it comes to large scale, they lose their ability to efficiently solve the problem within a few minutes, reaching to several hours. This can be explained by applying LS to these scales, the large number of neighboring solutions makes it too costly for the full search in each neighborhood. To overcome this, some LS-based methods apply some kind of heuristic pruning, reducing the number of solutions searched.

A recent case of success can find solutions for up to 30000 customers within minutes of execution time, by considering move-specific pruning techniques. More methods apply limits to the search space, usually by grouping the customers or by some sort of threshold as reviewed in [21]. However, limiting the search space is not a trivial task, since if poorly done can prevent good solutions from being found.

Hyper-heuristics (HHs) reduce the level of domain knowledge to create a good heuristic. HHs have been applied to automate heuristic sequencing, planning systems, parameter control, and heuristic learning methods [22], with several cases of success. When learning heuristics, factors such as the types of components to be considered, the techniques used, and which parameters should this algorithm use, need to be considered. One popular approach for building HHs is the EC techniques, such as Genetic Algorithm (GA) and Genetic Programming (GP). GA has been applied for several search problems including searching for optimal heuristic sequencing, such as for the bin-packing problem. GP is more used for creating a heuristic rule that builds a solution, rather than improving it, such as in the Dynamic Job-Shop Scheduling [23].

For large-scale problems (focusing on the Automobile Navigation), however, HHs have not been well explored. One example of HH being applied to an LSAutomobile Navigation with Time Windows [24], is where the large problem size is handled before the solution is fed to the HH. The approach solves the problem and search space with a column generation technique. Guided Local Search (GLS) is a deterministic algorithm that attempts to escape the local optimum that LS algorithms inevitably fall [25]. GLS has a set of features that can be selected to penalize the current solution, moving it away from the pitfalls. This is done by using different objective functions to guide the solution, rather than changing the solution itself [26].

Knowledge Guided Local Search (KGLS) is presented in [27] where the authors apply the Guided Local Search (GLS) with a newly introduced operator and penalization functions. This was later adapted to large scale. The KGLS operates by sequential applications of a Local Search algorithm. These phases remove the undesired edges to find new solutions that can potentially lead to a better overall solution [28]. These penalization functions were based on a study by the same authors in which they investigated similarities across different Automobile Navigation solutions, according to several metrics. One of the most effective metrics was the width of the route's Clustering Analysis [29].

Through the grouping of items according to the degree of similarity between item sets, clustering in data mining solves problems. According to author [30], aiming to achieve maximum dissimilarity between clusters with the least similarity between each item, the clustering approach divides data into many clusters. A component of unsupervised learning is the clustering approach. This is so that each class is immediately categorized according to its characteristics rather than having a label thanks to clustering.

Many heuristic algorithms have been developed on the subject of route optimization, especially in the categories of Evolutionary Search and Local Search. The effectiveness of local search algorithms, including simulated annealing, iterated local search, and tabu search, is limited in complex scenarios like automobile Navigation with simultaneous pickup and delivery because they frequently become stuck in local optima, despite providing iterative gains. On the other hand, using techniques like selection, combination, and mutation, evolutionary search optimizes several solutions at once. Interestingly, classical approaches have proven effective for small to medium-sized Automobile Navigation, but the exponential growth of the solution space makes it difficult for them to handle bigger cases effectively. This study addresses the shortcomings of current heuristics by presenting a novel hybrid approach that combines Particle Swarm Optimisation (PSO) with Differential Evolution (DE). In comparison to conventional methods, our methodology allows for effective solutions in a substantially shorter amount of time by combining the benefits of both DE and PSO, improving computing efficiency while simultaneously improving solution quality. Moreover, the hybrid framework that has been suggested makes it easier to explore the solution space more thoroughly, proving its superiority in managing the complexity of large-scale Automobile Navigation a problem that has not been successfully handled by earlier heuristic techniques. This relative benefit highlights how important our method is to push the boundaries of Automobile Navigation optimization.

3 METHODOLOGY

Design and specification are an essential part of any implementation of a project. In this chapter, we will elaborate on our project deeply with the help of a diagram that is based on our research. We are going to explain our algorithm with the help of a flow chart to show the flow of the algorithm that will work in the optimization of the route. The purpose of making the flow chart diagram is to guide the direction of the algorithm and how the system will work in the implementation of all possible steps. This section provides a detailed explanation of the suggested algorithm, which is based on swarm particle optimization. PSO is a method of soft computing that typically draws influence from the theory of bird swarms. PSO is created through ongoing nonlinear function optimization. It considers not just the space to be found and the Automobile positions to be adjusted for the best positioning, but also the Automobile's previous position and the positions of its neighbours. VANETs are limited by some factors, including high Automobile speeds, clustering, predefined transport directions, interference, and frequent topology changes. Consequently, selecting the subsequent Automobile (CH) in VANETs is a crucial task.

To improve comprehension of the hybrid algorithm that blends Differential Evolution (DE) and Particle Swarm Optimisation (PSO), Figure 1 provides a flow of the algorithm that illustrates how these two algorithms interact. The processes and feedback loops that are part of the hybrid strategy are illustrated in this figure, which also explains how the particle movement and communication concepts of PSO work in tandem with the mutation and crossover mechanisms of DE.

3.1 Particle Swarm Optimization (PSO)

Kennedy and Eberhart introduced PSO in 1995. PSO gets its ideas from the social behaviors of animals, such as fish or birds going out and foraging. The process involves beginning a list of potential solutions to find the best one within a certain search space. The basis of PSO is the idea that every particle can be analogously compared to another particle. Dispersed across the search area, each particle represents a potential solution and comes together to form a collective known as a swarm. The basic pseudocode of PSO is shown in Figure 2.

Each particle is responsible for locating itself within the search space. Once the locations of particles are identified, they will initiate a process of intercommunication to enable the exchange of information along the entirety of their trajectory. Within the PSO algorithm, the process of movement is influenced by an individual cognitive (Personal best) desire to search and the group's collective action (Global best). There exists a categorization of data aggregation into two distinct types: position and velocity. The updated velocity of each particle depends on several critical parameters: the inertia weight, the

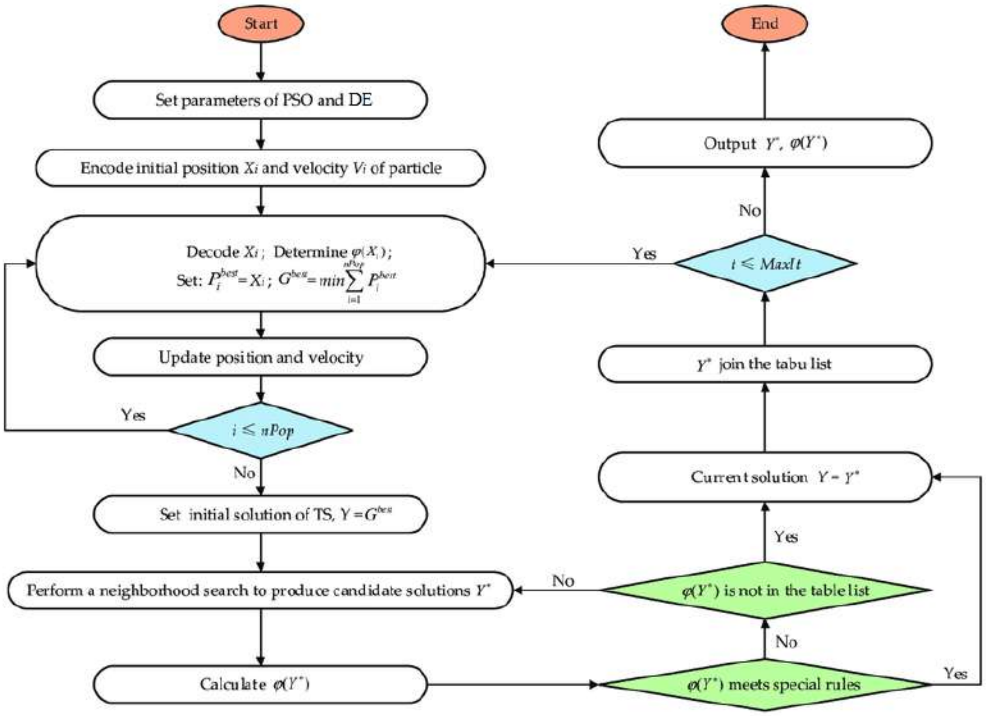

Figure 1. Flowchart of the Hybrid Algorithm Combining DE and PSO

coefficient for the personal best, and the coefficient for the global best. This updated velocity consequently leads to the derivation of a new position of a particle, which is acquired through the best position attributed to each particle and that of the entire swarm. This methodology resides within the framework of evolutionary algorithms.

3.2 Differential Evolution (DE)

In 1995, Storn and Price introduced DE. It belongs to the family of population-based optimization algorithms called evolutionary algorithms. Figure 3 illustrates how DE works. Essentially, it works with a population of potential solutions, or vectors. The derived solution of the method is expressed by each vector. To create unique trial solutions, DE uses a two-pronged strategy—mutation and crossover operations—to improve these candidate solutions throughout multiple generations. These trial solutions are then assessed about the values of the objective function. If a trial solution works better than the population solution, it replaces the related solution; if it doesn't work better, it's ignored. DE can be tailored and adapted to various problem areas by varying parameters including population size, mutation tactics, and crossover techniques.

The basic DE variation used in this work, called DE/rand/1, generates vectors by adding a weighted difference between two randomly selected vectors and a third randomly selected vector. We show our PSO-based QoSR Navigation method. This method for global optimization works well when the answer is an argument or an external variable in an n-dimensional space. PSO models the motion of a group of particles to find the best possible solution to the problem. A merit criterion is employed to assess the outcomes after these particles are moved in the reaction space. Particles in the same communication group with higher merit values tend to move faster over time. The strategy is flexible against the local optimal response problem due to the number of swarming particles. Any particle with a position is specified. The equation of this is given below [31]:

```
Particle Swarm Optimization (PSO)
Set the following parameters: population size, number of iterations, cognitive learning
factor (Cp), social learning factor (Cg), and inertia weight (W)
Initialize a swarm of particles with random position and zero velocity
For t = 1 to the maximum iteration
    For i = 1 to the swarm size
        Decode to obtain the customer and the vehicle sequence
        Consider the time window and capacity constraints
        Evaluate objective value or fitness value
        If particle's fitness value is better than its Pbest in history:
            Update Pbest to current position
        End if
        If particle's fitness value is better than Gbest:
            Update Gbest to current position
        End if
        For d = 1 to the total dimension
            Generate a random number R1 between 0 and 1
            Generate a random number R2 between 0 and 1
          Update particle's velocity using the formula:
          Velocity [d] = W* Velocity [d] + Cp * R1 *(Pbest – Current position[d]) +
        Cg * R2 *  (Gbest – Current position[d])
          Update particle's position using the formula:
            Position[d] = Position[d] + Velocity[d]
        End for
    End for
End for
Return Gbest as the optimal solution
```

Figure 2. Pseudocode of Basic PSO

$$V_i^{(t+1)} = w \cdot V_i^t + c_i \cdot r_1(y_i - x_i^t) + c_j \cdot r_2(y_j - x_i^t) \tag{1}$$

$$x_i^{(t+1)} = x_i^t + V_i^{(t+1)} \tag{2}$$

$$w = w_{\max} - \frac{t\,(w_{\max} - w_{\min})}{t_{\max}} \tag{3}$$

At iteration t, particle i travels through space at the speed of vi(t). Throughout their entire existence, particles retain the ideal positions for storage. Yi is the best experience a particle has or the best place a particle observes. The best-observed location in the entire group, known as is known to the particles. The starting weight, w, falls linearly from wmax to wmin, while the constants ci and cj remain constant. The maximum number of repetitions that the algorithm takes into consideration is called t-max. By using PSO to isolate the communication line between two Automobiles, it is expected that the QoS will be enhanced.

4 COMPUTATIONAL RESULT

Table 2 shows the numerical results of the transportation costs obtained from the exact method, the current practice, and metaheuristics. For metaheuristics, the best, average, and standard deviation (SD) values from 10 replicated runs for each instance are reported. Table 3's results make it clear that in certain cases (e.g., 1 to 6 when there are up to 13 clients), the accurate method quickly determines the global solution. Similar to this, while PSO and DE had numerous options, they were both able to identify the best answers for cases 1 through 6. In nearly every case of this magnitude, DE reliably offered a solution with an SD equal to zero. Conversely, PSO gives DE greater SD values in all scenarios.

```
Differential Evolution Algorithm (DE)
Set the following parameters: population size, number of iterations,
Scale factor (F), Crossover rate (Cr), and Crossover type.
Initialize a population of vector with randomly
For t = 1 to the maximum iteration
    For i = 1 to  the population size:
     Generate three distinct random parents (a, b, c) from the current population
     to generate a trail vector (u) by blending the solutions a, b, and c
     For d = 1 to the total dimension
         If a random number r from [0, 1] is less than CR or equal to d is randomly selected:
             u[d] = a[d] + F * (b[d] - c[d])
         Else:
             u[d] = Target vector[d]
         End if
     End for
     Decode to obtain the customer and the vehicle sequence
     Consider the time window and capacity constraints
     Evaluate objective value or fitness value
     If the trial vector is better than the current candidate solution
         Vector  = u
     Else:
         Vector  = Target vector
     End if
   End for
   Update global best vector.

End for
Return the best vector as the optimal solution
```

Figure 3. Pseudocode of Basic DE

This suggests that in terms of stability and dependability, DE performs better than PSO. It is also important to highlight that PSO and DE performed well in situations where the accurate technique faced challenges.

Table 1. Comparison Result of Total Transportation obtained from PSO and DE

Instance	Customers	Automobiles	Current Practice	Optimal Solution	PSO Best	PSO Avg.	PSO SD	PSO Time (Mins)	DE Best	DE Avg.	DE SD	DE Time (Mins)
1	8	2	-	1811	1811	1811	0.00	0.11	1811	1811	0.00	0.46
2	9	2	-	1816	1816	1816	0.00	0.06	1816	1816	0.00	0.47
3	10	2	-	1832	1832	1839	4.84	1.05	1832	1832	0.00	0.54
4	11	2	-	1838	1838	1912	111	1.55	1838	1838	0.00	0.52
5	12	2	-	1843	1843	1846	5.23	50.79	1843	1843	0.00	0.68
6	14	2	-	1860*	1860	1863	2.98	>600	1860	1860	0.00	1.51
7	15	2	-	1833*	1879	1881	2.31	>600	1879	1879	0.00	0.62
8	16	2	-	1880*	1880	1889	7.25	>600	1880	1880	0.09	0.47
9	22	2	2060	2039*	2068	2091	18.08	>600	2040	2048	5.56	0.52
10	22	2	2060	2039*	2068	2091	18.08	>600	2040	2048	5.56	0.52
11	23	2	2120	2032*	2077	2132	39.64	>600	2037	2048	14.02	1.20
12	24	2	2396	2142*	2293	2405	74.11	>600	2182	2208	18.09	0.69
13	25	2	2196	2108*	2204	2254	45.66	>600	2074	2103	19.73	0.56
14	27	2	2101	2064*	2173	2242	59.77	>600	2106	2130	12.88	0.59
15	27	2	2101	2064*	2173	2242	59.77	>600	2106	2130	12.88	0.59
16	27	2	2287	2142*	2270	2408	114.06	>600	2158	2178	17.51	0.56
17	27	2	2176	2071*	2162	2247	82.58	>600	2087	2103	21.25	1.38
18	27	2	2151	2062*	2159	2283	68.74	>600	2094	2115	17.46	1.46
19	29	2	2277	2093*	2231	2306	60.40	>600	2083	2146	28.88	1.43
20	31	2	2141	2079*	2242	2314	39.03	>600	2119	2139	15.73	0.60
21	31	2	2316	2179*	2338	2459	70.51	>600	2220	2256	20.66	0.60
22	31	2	2316	2189*	2338	2472	105.45	>600	2211	2252	27.67	1.39
23	31	2	2176	2045*	2122	2234	58.92	>600	2062	2100	14.40	0.60
24	32	2	2318	2239*	2440	2842	673.97	>600	2272	2312	20.97	0.65
25	32	2	2355	2258*	2344	2506	90.61	>600	2284	2308	13.70	0.61
26	32	2	2346	2078*	2220	2319	59.60	>600	2089	2134	23.07	0.61
27	35	2	2287	2176*	2320	2498	117.73	>600	2240	2287	27.92	0.67
28	36	2	2229	2124*	2350	2467	59.44	>600	2179	2239	31.42	0.67
29	36	2	2469	2302*	2502	2702	128.48	>600	2397	2436	29.63	0.66
30	39	2	2278	2139*	2355	2469	88.61	>600	2227	2262	28.78	0.72

Equations 4 and 5 state that a positive outcome means that the metaheuristic produces a worse solution than the one that is being evaluated [32]. On the other hand, a negative number suggests that the metaheuristic yields a better result than the solution used as a reference. The PD value comparison findings are displayed in Table 2.

$$PD = \frac{Sol_{metaheuristic} - Sol_{optimal}}{Sol_{optimal}} \times 100\% \tag{4}$$

$$PD = \frac{Sol_{metaheuristic} - Sol_{practice}}{Sol_{practice}} \times 100\% \tag{5}$$

Table 2. Comparison Result of PD Value

Instance	Customers	Automobiles	Current Practice	Optimal Solution	PSO Best	PSO Avg.	PSO SD	PSO Time (Mins)	DE Best	DE Avg.	DE SD	DE Time (Mins)
1	8	2	2	1811	-	-	0.00	-	1811	0.00	0.00	-
2	9	2	2	1816	-	-	0.00	-	1816	0.00	0.00	-
3	10	2	2	1832	-	-	0.00	-	1832	0.00	0.00	-
4	11	2	2	1838	-	-	0.00	-	1838	0.00	0.00	-
5	12	2	2	1840	-	-	0.00	-	1840	0.00	0.00	-
6	13	2	2	1843	-	-	0.00	-	1843	0.00	0.00	-
7	15	2	2	1833*	-	-	2.51	-	1879	2.51	2.51	-
8	16	2	2	1880*	-	-	0.00	-	1880	0.00	0.00	-
9	22	2	2	2039*	2060	2060	1.46	0.42	2040	0.06	0.06	-0.97
10	23	2	2	2032*	2120	2120	2.17	-2.07	2037	0.20	0.20	-3.96
11	24	2	2	2142*	2396	2396	7.05	-4.29	2182	1.85	1.85	-8.94
12	25	2	2	2108*	2196	2196	4.56	0.37	2074	-1.61	-1.61	-5.56
13	27	2	2	2064*	2101	2101	5.27	3.44	2106	2.00	2.00	0.23
14	27	2	2	2064*	2101	2101	5.27	3.44	2106	2.00	2.00	0.23
15	27	2	2	2142*	2287	2287	5.99	-0.72	2158	0.75	0.75	-5.63
16	27	2	2	2142*	2287	2287	5.99	-0.72	2158	0.75	0.75	-5.63
17	27	2	2	2071*	2176	2176	4.36	-0.64	2087	0.78	0.78	-4.06
18	27	2	2	2062*	2151	2151	4.70	0.35	2094	1.57	1.57	-2.64
19	29	2	2	2093*	2277	2277	6.57	-2.03	2083	-0.48	-0.48	-8.51
20	31	2	2	2079*	2141	2141	7.87	4.72	2119	1.95	1.95	-1.02
21	31	2	2	2179*	2316	2316	7.32	0.94	2220	1.90	1.90	-4.16
22	31	2	2	2189*	2316	2316	6.82	0.97	2211	1.03	1.03	-4.51
23	31	2	2	2045*	2176	2176	3.77	-2.49	2062	0.85	0.85	-5.23
24	32	2	2	2239*	2318	2318	9.00	5.29	2272	1.47	1.47	-1.99
25	32	2	2	2258*	2355	2355	3.82	-0.44	2284	1.16	1.16	-2.99
26	32	2	2	2258*	2355	2355	3.82	-0.44	2284	1.16	1.16	-2.99
27	32	2	2	2078*	2346	2346	6.83	-5.37	2089	0.53	0.53	-10.94
28	35	2	2	2176*	2287	2287	6.60	1.43	2240	2.93	2.93	-2.06
29	36	2	2	2215*	2544	2544	1.60	-11.52	2207	-0.37	-0.37	-13.24
30	36	2	2	2302*	2469	2469	8.68	1.35	2397	4.13	4.13	-2.90
31	39	2	2	2139*	2278	2278	10.10	3.41	2227	4.11	4.11	-2.22
Average							4.26	-0.07	1.00	1.00	-4.45	

Furthermore, the comparison of convergence behavior between PSO and DE is visually demonstrated using the four largest instances in this study. As Figure 4 indicates PSO looked for workable solutions fast in the early iterations, but as the search went on, PSO regularly ran into difficulty eluding locally optimal options. Conversely, with an identical quantity of function evaluations, DE revealed an ongoing improvement in producing superior solutions in compared to PSO.

The computed outcomes unequivocally demonstrate the variations in transportation costs attained by Differential Evolution (DE) and Particle Swarm Optimisation (PSO). For minor cases, both algorithms were successful in finding the best answers; however, DE constantly showed a standard deviation (SD) of zero, demonstrating its consistency and dependability throughout several runs. PSO, on the other hand, displayed larger SD values, indicating more variability and inconsistent results. This implies that DE performs better than PSO in terms of stability, which is important for practical applications that need for ndable cost reduction. Furthermore, as the search went on, PSO had trouble breaking free from local optima, but DE consistently produced better results. These results highlight DE's benefit in providing reliable and efficient solutions for vehicle routing issues, highlighting its applicability in real-world situations.

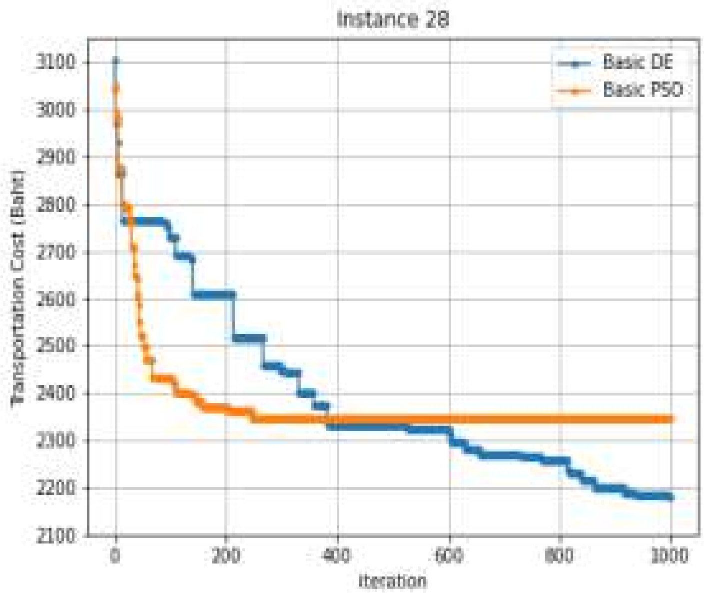

Figure 4. Comparing of behavior between PSO and DE

5 Conclusion

We can conclude that an optimization method is vital for the delivery and transport sector. A proficient route optimization algorithm has the potential to enhance an organization's or company's productivity while producing significant monitor advantages. Our algorithm, created specifically for this project, solely handles static orders that is, orders that are received and optimized. This method also partially resolves the issue. However, taking care of the dynamic orders might lead to far superior solutions. When we refer to dynamic orders, we mean orders that are received at the exact moment that the existing orders are being handled; that is, any orders that are received during this period should also be taken into consideration when planning processing. This would aid in proposing superior choices. Predicting future orders and scheduling current orders appropriately, as well as accounting for real-time road conditions and other factors that can impact the overall project completion time, would be further substantial enhancements. Ultimately, there is space for growth when it comes to this subject. To improve the success of the solutions, more research in this area may be conducted in the future. A better approach would be to use machine learning prediction algorithms to project how many orders each restaurant would receive in the future. In this manner, if we anticipate a restaurant to be busy with orders in the upcoming hour, we can allocate the present orders to riders who are farther away, bringing the number of riders closer to the restaurant in the next hour. Taking into account current road conditions, such as congested or damaged roads, traffic bottlenecks, road closures, construction, etc., could be another area for development. Real-time road conditions would allow us to determine how long each potential route would take, and from there we might recommend the shortest routes in terms of time. For example, a longer but cleaner route could complete the task faster than a shorter but less efficient one.

To enable more efficient automobile Navigation by responding to current conditions and demand predictions, future work could concentrate on combining machine learning approaches to predict dynamic orders and real-time road conditions.

Author Contributions

Syed Muhammad Daniyal: Conceptualization, Methodology, Software **Mohsin Mubeen Abbasi**: Data curation, Writing- Original draft preparation. **Dilbar Hussain**: Visualization, Investigation. **Usama Amjad**: Software, Validation. **Abdul Basit Abro**: Supervision. **Mehwish Naeem**: Writing- Reviewing and Editing.

Compliance with Ethical Standards

It is declared that all authors don't have any conflict of interest. It is also declared that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

References

[1] A. Ait-Saadi, Y. Meraihi, A. Soukane, A. Ramdane-Cherif, and A. B. Gabis, "A novel hybrid chaotic aquila optimization algorithm with simulated annealing for unmanned aerial vehicles path planning," *Computers and Electrical Engineering*, vol. 104, p. 108461, 2022.

[2] J. Zhao, M. Mao, X. Zhao, and J. Zou, "A hybrid of deep reinforcement learning and local search for the vehicle routing problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7208–7218, 2020.

[3] P. Sun, L. P. Veelenturf, M. Hewitt, and T. Van Woensel, "Adaptive large neighborhood search for the time-dependent profitable pickup and delivery problem with time windows," *Transportation Research Part E: Logistics and Transportation Review*, vol. 138, p. 101942, 2020.

[4] Z. H. Ahmed and M. Yousefikhoshbakht, "An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows," *Alexandria Engineering Journal*, vol. 64, pp. 349–363, 2023.

[5] M. Ibrahim, F. Nurhakiki, D. Utama, and A. Rizaki, "Optimised genetic algorithm crossover and mutation stage for vehicle routing problem pick-up and delivery with time windows," in *IOP Conference Series: Materials Science and Engineering*, vol. 1071, no. 1. IOP Publishing, 2021, p. 012025.

[6] W. Zhang, D. Yang, G. Zhang, and M. Gen, "Hybrid multiobjective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for vrptw," *Expert Systems with Applications*, vol. 145, p. 113151, 2020.

[7] T. Stadtler, S. Nita, and J. Dünnweber, "A parallel hybrid genetic search for the capacitated vrp with pickup and delivery," in *34th International Workshop, LCPC 2021, Newark, DE, USA, October 13–14, 2021*, 2023.

[8] P. Li, C. Zhi, and W. Li, "An algorithm to solve heterogeneous vehicle routing problem with second trip," *IEEE Access*, vol. 9, pp. 12 241–12 255, 2021.

[9] F. R. Dorcheh and M. Rahbari, "Greenhouse gas emissions optimization for distribution and vehicle routing problem in a poultry meat supply chain in two phases: a case study in iran," *Process Integration and Optimization for Sustainability*, vol. 7, no. 5, pp. 1289–1317, 2023.

[10] R. Prakash and S. Pushkar, "Green vehicle routing problem: Metaheuristic solution with time window," *Expert Systems*, vol. 41, no. 6, p. e13007, 2024.

[11] D. A. Albalawneh and M. Mohamed, "A new federated genetic algorithm-based optimization technique for multi-criteria vehicle route planning using arcgis network analyst," *International Journal of Pervasive Computing and Communications*, vol. 20, no. 2, pp. 206–227, 2024.

[12] R. M. A. BARACHO, M. J. M. VIDIGAL, M. F. PORTO, and B. COUTO, "Interdisciplinarity in smart systems applied to rural school transport in brazil," *Journal of Systemics, Cybernetics and Informatics*, vol. 22, no. 4, pp. 52–59, 2024.

[13] A. Mohammadi, B. Kiani, H. Mahmoudzadeh, and R. Bergquist, "Pedestrian road traffic accidents in metropolitan areas: Gis-based prediction modelling of cases in mashhad, iran," *Sustainability*, vol. 15, no. 13, p. 10576, 2023.

[14] A. Amini and M. Haughton, "A mathematical optimization model for cluster-based single-depot location-routing e-commerce logistics problems," *Supply Chain Analytics*, vol. 3, p. 100019, 2023.

[15] A. Bokhare and P. Metkewar, "Identifying best goalkeepers problem is a np-hard?" in *World Conference on Information Systems for Business Management*. Springer, 2023, pp. 415–421.

[16] S. Liu, J. G. Cavalcanti Costa, Y. Mei, and M. Zhang, "Gpgls: Genetic programming guided local search for large-scale vehicle routing problems," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2024, pp. 36–51.

[17] Y. Ma, Z. Cao, and Y. M. Chee, "Learning to search feasible and infeasible regions of routing problems with flexible neural k-opt," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[18] W. Zhang, D. Yang, G. Zhang, and M. Gen, "Hybrid multiobjective evolutionary algorithm with fast sampling strategy-based global search and route sequence difference-based local search for vrptw," *Expert Systems with Applications*, vol. 145, p. 113151, 2020.

[19] J. G. C. Costa, Y. Mei, and M. Zhang, "An evolutionary hyper-heuristic approach to the large scale vehicle routing problem," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 2109–2116.

[20] A. N. Jasim and L. C. Fourati, "Guided genetic algorithm for solving capacitated vehicle routing problem with unmanned-aerial-vehicles," *IEEE Access*, 2024.

[21] J. G. C. Costa, Y. Mei, and M. Zhang, "An evolutionary hyper-heuristic approach to the large scale vehicle routing problem," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 2109–2116.

[22] M. Sánchez, J. M. Cruz-Duarte, J. carlos Ortíz-Bayliss, H. Ceballos, H. Terashima-Marin, and I. Amaya, "A systematic review of hyper-heuristics on combinatorial optimization problems," *IEEE Access*, vol. 8, pp. 128 068–128 095, 2020.

[23] A. A. Laghari, A. K. Jumani, R. A. Laghari, H. Li, S. Karim, and A. A. Khan, "Cognitive robotics."

[24] J. G. C. Costa, Y. Mei, and M. Zhang, "An evolutionary hyper-heuristic approach to the large scale vehicle routing problem," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 2109–2116.

[25] A. A. Abro, A. A. Khan, M. S. H. Talpur, I. Kayijuka, and E. Yaşar, "Machine learning classifiers: a brief primer," *University of Sindh Journal of Information and Communication Technology*, vol. 5, no. 2, pp. 63–68, 2021.

[26] Z. Li and V. Tam, "Adaguide: An adaptive and guided differential evolution for continuous optimization problems," *Applied Intelligence*, pp. 1–79, 2024.

[27] J. G. C. Costa, Y. Mei, and M. Zhang, "Learning penalisation criterion of guided local search for large scale vehicle routing problem," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2021, pp. 1–8.

[28] A. A. Laghari, A. K. Jumani, R. A. Laghari, H. Li, S. Karim, and A. A. Khan, "Unmanned aerial vehicles advances in object detection and communication security review," *Cognitive Robotics*, 2024.

[29] A. A. Abro, “Identifying the machine learning techniques for classification of target datasets,” *Computer Science-Artificial Intelligence*, 2020.

[30] T. Li, A. Rezaeipanah, and E. M. T. El Din, “An ensemble agglomerative hierarchical clustering algorithm based on clusters clustering technique and the novel similarity measurement,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 3828–3842, 2022.

[31] M. Z. Ghawy, G. A. Amran, H. AlSalman, E. Ghaleb, J. Khan, A. A. Al-Bakhrani, A. M. Alziadi, A. Ali, and S. S. Ullah, “An effective wireless sensor network routing protocol based on particle swarm optimization algorithm,” *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 8455065, 2022.

[32] B. T. Tezel and A. Mert, “A cooperative system for metaheuristic algorithms,” *Expert Systems with Applications*, vol. 165, p. 113976, 2021.