

CAPUTO-INSPIRED ADAPTIVE GENETIC ALGORITHM AND TAGUCHI DESIGN OF EXPERIMENT TO SOLVE CAPACITATED AMBULANCE ROUTING PROBLEM

*Marissa Marcella Daulip Lakkui, Jackel Vui Lung Chew**

*Faculty of Computing and Informatics
Universiti Malaysia Sabah Labuan International Campus
Labuan F.T., 87000, Malaysia
marsmarcel18@gmail.com, jackelchew93@ums.edu.my*

Received: 22 May 2026; Accepted: 1 October 2026

Abstract. This study proposes an adaptive genetic algorithm based on a nabla Caputo fractional difference with the Taguchi design of experiment for solving the capacitated ambulance routing problem. The optimization problem considered aims to minimize the total ambulance routing cost while satisfying ambulance availability and capacity. In this study, a nabla Caputo fractional difference-based adaptive mutation mechanism is designed and incorporated into the genetic algorithm evolutionary process. This so-called memory-aware mechanism captures historical fitness behavior and is used as an adaptive control signal for dynamically adjusting the mutation rate during optimization. To obtain appropriate parameter settings systematically, the Taguchi design of experiment is employed in two stages. The first stage tunes the genetic algorithm parameters, such as the maximum number of generations, the population size, and the fixed mutation rate. While the second stage tunes the nabla Caputo fractional-memory parameters, such as the fractional order, the memory window, the scaling coefficient, and the maximum adaptive mutation rate. The proposed algorithm is interpreted through the lens of nonlinear dynamical systems. For the performance evaluation, this study uses Augerat benchmark instances. The results show that the proposed algorithm is competitive against established hybrid metaheuristics. These findings suggest that integrating the nabla Caputo fractional-memory dynamics with statistically guided parameter tuning offers a promising solution for the capacitated ambulance routing problem.

MSC 2010: 90C27, 68W40

Keywords: *adaptive genetic algorithm, nabla Caputo fractional difference, Taguchi design of experiment, capacitated ambulance routing problem, metaheuristic*

1. Introduction

Emergency medical services are an essential component of modern healthcare systems because they provide timely pre-hospital care and rapid transport of patients to the nearest hospital [1]. In emergencies, the efficiency of ambulance operations

* Corresponding author

is closely linked to patient outcomes, response effectiveness, and the overall quality of service delivery [2]. As the number of emergency demand locations increases, the need for efficient ambulance routing becomes increasingly important.

The ambulance routing problem is a specialized variant of the classical vehicle routing problem (VRP) [3], in which ambulances must be assigned and routed to serve geographically dispersed demand locations. In practice, ambulance routing is more difficult than conventional routing because it involves limited ambulance availability and operational constraints. One important extension of this problem is the Capacitated Ambulance Routing Problem (CARP), where each ambulance may serve multiple demand locations provided that the total demand does not exceed ambulance capacity. The primary objective is to minimize the total ambulance routing cost while ensuring that each demand location is visited exactly once and all operational constraints are satisfied. Due to the combinatorial nature of route construction, CARP is difficult to solve optimally, especially when the number of demand locations increases.

Metaheuristics is a commonly used approach for solving CARP. Marinakis et al. [4] studied the use of particle swarm optimization for the vehicle routing problem and demonstrated its effectiveness in exploring difficult combinatorial search spaces. Similarly, Tlili et al. [5] introduced a tabu search-based hyper-heuristic for heterogeneous ambulance routing. In addition, various metaheuristic techniques have been reported in the literature, where evolutionary mechanisms have been associated with improved search performance in solving routing problems [6]. More specifically, the studies by Tlili et al. [7] and Zidi et al. [8] confirmed that hybrid metaheuristic approaches can effectively accommodate the operational constraints of ambulance routing while maintaining competitive solution quality. Although these approaches have strong potential, their overall performance is dependent on the choice of parameter settings. Consequently, systematic parameter tuning remains an issue in the design of metaheuristic algorithms for CARP.

One of the ways to overcome the metaheuristics parameter tuning issue is by using the Taguchi design of experiment (TDOE). This design of experiment employs orthogonal arrays and signal-to-noise analysis to identify robust parameter settings. As a result, the parameter tuning process has become more systematic than the trial-and-error practice [9–14]. The combination of GA and TDOE has also been investigated in other optimization domains. Farahbakhsh and Kheirkhah [15] developed GA-TDOE for a multi-period inventory routing problem, while Cetin et al. [16] applied TDOE to select GA parameters for storm-sewer network optimization. These studies further demonstrate the usefulness of TDOE for improving the robustness of GA parameter selection. Nevertheless, although TDOE improves metaheuristic parameter robustness, the underlying search dynamics of a metaheuristic algorithm remain fundamentally unchanged.

Among various metaheuristic algorithms available in the literature, this study focuses on the Genetic Algorithm (GA) to solve the CARP. GA is selected due to its ability to produce near-optimal solutions within a reasonable computational time.

In addition, previous studies by Tlili et al. [7] and Zidi et al. [8] used GA as the baseline algorithm for evaluating their hybrid metaheuristic approaches for solving CARP, which demonstrate its reliability in solving routing optimization problems. However, GA often suffers from premature convergence, loss of population diversity, and stagnation in local optima, particularly when solving routing optimization problems with large search spaces [17]. These limitations reduce the exploration capability of GA and affect the quality of the final solution. Therefore, to handle these limitations, this study proposes an adaptive GA based on a nabla Caputo fractional difference. This adaptive GA can be known as CaputoGA [18].

Motivated by the findings in Chew [18] and inspired by the systematic parameter tuning technique by Taguchi [9], this study proposes a hybrid optimization framework called CaputoGA-TDOE for solving the CARP. Accordingly, the contribution of this study is not the use of GA-TDOE alone, but the integration of a nabla Caputo-inspired memory-aware adaptive mutation mechanism with a two-stage TDOE procedure for CARP. The rest of this paper is organized as follows. Section 2 describes the CARP model considered in this study. Section 3 details the proposed CaputoGA-TDOE framework. Section 4 presents the interpretations of the CaputoGA-TDOE algorithm through the lens of nonlinear dynamical systems. Section 5 discusses simulation results and comparative analysis against the established hybrid metaheuristics. Finally, Section 6 concludes the paper and outlines our future research.

2. Capacitated ambulance routing problem model

This study considers the mathematical model of CARP by Chew [18], which can be described as follows. Let the objective function be

$$\min \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij}, \quad (1)$$

where c_{ij} represents the cost measured based on the distance that the ambulance traveled between i and j . The solutions of equation (1) are assumed to exist subject to the following constraints,

$$\sum_{\substack{j=0 \\ j \neq i}}^n x_{ij} = 1, \quad \forall i \in \{1, 2, \dots, n\}, \quad (2)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} = 1, \quad \forall j \in \{1, 2, \dots, n\}, \quad (3)$$

where n is the total number of demand locations, and

$$\sum_{j=1}^n x_{0j} \leq K, \quad \sum_{i=1}^n x_{i0} \leq K, \quad (4)$$

where K denotes the number of available ambulances.

To ensure route feasibility and eliminate subtours, this model employs

$$u_i - u_j + Qx_{ij} \leq Q - d_j, \quad \forall i, j \in \{1, 2, \dots, n\}, i \neq j, \quad (5)$$

$$d_i \leq u_i \leq Q, \quad \forall i \in \{1, 2, \dots, n\}, \quad (6)$$

where u_i represents the cumulative demand served up to and including demand location i , d_i denotes the demand at i , and Q represents the ambulance capacity.

For the decision variable x_{ij} that determines whether the ambulance travels from i to j , or not, this study uses the binary values. However, these values are indirectly represented through permutations of demand locations in the GA encoding. For practical implementation, each chromosome represents a permutation of demand locations rather than the binary variables directly. A deterministic split-decoding procedure subsequently partitions each permutation into ambulance routes. Each decoded route is accepted only when its accumulated demand does not exceed Q , while the total number of generated routes must not exceed K . Therefore, the permutation representation is an indirect encoding of the binary routing decisions in equations (1)-(6), and the decoder maps each chromosome to a feasible CARP solution satisfying the same mathematical constraints. The GA therefore searches over permutations of demand locations while the decoding procedure enforces the feasibility conditions defined by equations (2)-(6).

3. Hybrid optimization framework for ambulance routing

The proposed CaputoGA-TDOE framework combines GA as the principal search for the solution, a nabla Caputo fractional difference-based mechanism as a memory-aware mechanism that adaptively controls the mutation rate, and TDOE as a systematic parameter tuning mechanism. The following subsections describe these three components in detail.

3.1. Genetic algorithm design

To solve the CARP model presented in Section 2 using GA, let us consider a chromosome represented as a permutation of demand locations. Such a chromosome can be generally expressed as $C_p = [g_1, g_2, \dots, g_n]$, $p = 1, 2, \dots, p_{total}$, where each gene g_i corresponds to one demand location, and each demand location appears exactly once in the chromosome.

The initial population is generated randomly by creating multiple chromosomes C_p . This provides a diverse group of candidate chromosomes and allows the early search process to explore multiple regions of the solution space. To evaluate the fitness of C_p , which is computed from the total routing cost, a split procedure is applied to ensure feasible routes. Starting from the first gene in the chromosome, the next genes are assigned sequentially to the current sequence while satisfying the imposed constraints. When the next gene violates the constraint, the current sequence is closed, and a new sequence is started. This process continues until all demand locations have been assigned in C_p . Specifically, the decoder checks the cumulative demand against Q and ensures that the number of constructed routes does not exceed K . A chromosome is therefore evaluated through its decoded feasible routes rather than directly through its permutation alone.

From the initial population, parent selection is carried out using tournament selection with a tournament size of three. For each tournament, three chromosomes are randomly sampled from the current population, and the chromosome with the lowest routing cost is selected. This process is repeated until two parent chromosomes are obtained.

For crossover, this study uses the order crossover (OX) technique based on the recommendation by Katoch et al. [17]. Two crossover points are selected, and the subsequence between them is copied from the first parent into the offspring. The remaining genes are then filled using the order of appearance in the second parent, excluding duplicates. Meanwhile, for mutation, this study uses the inversion mutation technique, which is also recommended by Katoch et al. [17]. Two positions are chosen randomly within the chromosome, and the subsequence between those positions is reversed.

To preserve solution quality across generations, an elitist replacement strategy is adopted. The best individual from the current population is copied directly into the next generation, while the remaining members of the new population are filled with newly generated offspring. This prevents high-quality solutions from being lost due to stochastic genetic operations [17].

3.2. Caputo-inspired adaptive mutation

In the conventional GA, mutation is commonly controlled by a fixed mutation rate. In the proposed CaputoGA, the mutation rate can be adjusted dynamically using a fractional-memory signal control [18]. Since GA evolves over discrete generations, the historical fitness behavior can be represented as a discrete sequence. Therefore, this study employs a nabla Caputo fractional difference on the generation-indexed fitness sequence as a fractional-memory operator that is consistent with the discrete nature of GA.

Let f_t denote the best fitness value at generation t , where $t = 0, 1, 2, \dots$, and let $t = 0$ denote the initial generation. For $0 < \alpha < 1$ and $t \geq 1$, the left nabla Caputo fractional difference can be expressed as [19, 20],

$${}_0^C \nabla_t^\alpha f(t) = G_0 \nabla_t^{-(1-\alpha)} \nabla f(t), \quad (7)$$

where the backward difference is $\nabla f(t) = f(t) - f(t-1)$ and $G_0 \nabla_t^{-(1-\alpha)}$ denotes the nabla fractional sum of order $1-\alpha$. Equation (7) can be expressed as a weighted aggregation of previous fitness changes,

$${}_0^C \nabla_t^\alpha f(t) = \sum_{i=0}^{t-1} \omega_i^{(\alpha)} [f(t-i) - f(t-i-1)], \quad (8)$$

where the fractional weight is defined by

$$\omega_i^{(\alpha)} = \frac{\Gamma(i+1-\alpha)}{\Gamma(1-\alpha)\Gamma(i+1)}, \quad i = 0, 1, 2, \dots \quad (9)$$

and can be computed recursively as

$$\omega_0^{(\alpha)} = 1, \quad \omega_i^{(\alpha)} = \left(\frac{i-\alpha}{i} \right) \omega_{i-1}^{(\alpha)}, \quad i \geq 1. \quad (10)$$

Therefore, equation (8) provides a discrete fractional-memory measure of the historical changes in the best fitness. The current value depends not only on the most recent fitness change but also on previous changes, with their contributions determined by the fractional-order weights. Unlike a continuous Caputo fractional derivative, equation (8) operates directly on the generation-indexed fitness sequence and does not require continuous interpolation.

For practical implementation, only the most recent W fitness values are retained. This follows the short-memory principle, in which the full historical fractional sum is truncated to a finite memory window [19]. Since W fitness values yield at most $W-1$ backward fitness differences, let $m_t = \min(t, W-1)$. The windowed nabla Caputo fractional-memory term used in the proposed CaputoGA is therefore

$${}_0^C \nabla_{t,W}^\alpha f(t) = \sum_{i=0}^{m_t-1} \omega_i^{(\alpha)} [f(t-i) - f(t-i-1)]. \quad (11)$$

Thus, only fitness changes available within the current memory window contribute to the fractional-memory signal. The fractional order α controls the weighting of historical fitness changes, whereas W determines the amount of fitness history retained in the calculation. This finite-window formulation reduces the computational requirement while preserving the memory-dependent characteristic of the nabla Caputo fractional difference.

The fractional-memory information obtained from equation (11) is incorporated into the GA evolutionary process by replacing the fixed mutation rate with a bounded adaptive mutation rate defined as

$$\mu_t = \min \left(\mu_{max}, \mu_0 \left(1 + \frac{1}{g_t + \epsilon} \right) \right), \quad (12)$$

where μ_0 denotes the fixed mutation rate, μ_{max} denotes the maximum adaptive mutation rate, ε is a small positive constant introduced to prevent division by zero, and the magnitude of the fractional-memory signal (g_t) is defined as

$$g_t = \lambda \cdot \left| {}_0^C \nabla_{t,W}^\alpha f(t) \right|, \quad (13)$$

where λ is the scaling coefficient that controls the sensitivity of the mutation-rate adaptation to the finite-memory nabla Caputo fractional-memory signal.

Equation (11) measures the memory-dependent variation in the recorded best-fitness sequence, while equation (13) scales the magnitude of this variation to obtain g_t . The resulting g_t is subsequently incorporated into equation (12) to determine the adaptive mutation rate μ_t .

3.3. Taguchi design of experiment

In the proposed CaputoGA, there are seven parameters that need to be identified and optimized. The parameters are the maximum number of generations (t_{max}), the population size (p_{total}), the fixed mutation rate (μ_0), the fractional order (α), the amount of fitness values (W), the scaling coefficient (λ), and the maximum adaptive mutation rate μ_{max} . Rather than adjusting parameters manually, this study employs the TDOE technique. This is particularly useful in metaheuristic design, where multiple parameters may interact and where exhaustive trial-and-error tuning is both costly and inconsistent.

The employed TDOE consists of two stages. In Stage 1, an L9 orthogonal array is used to tune the principal GA parameters t_{max} , p_{total} , and μ_0 . Each factor has three candidate levels, and each row of the L9 array defines one GA parameter configuration. The GA is executed for every L9 configuration over $r = 5$ repeated runs, the routing costs are recorded, and the smaller-the-better Taguchi signal-to-noise ratio is calculated. The factor levels associated with the highest signal-to-noise ratio are selected as the robust GA settings.

In Stage 2, the GA settings selected from Stage 1 are held fixed while an L16 orthogonal array is used to tune the nabla Caputo fractional-memory parameters α , W , λ , and μ_{max} . Each factor has four candidate levels, and each L16 row defines one nabla Caputo fractional-memory parameter configuration. CaputoGA is executed for every L16 configuration over the same number of repeated runs, and the smaller-the-better Taguchi signal-to-noise ratio is again used to select the robust nabla Caputo fractional-memory settings. In this way, the two-stage design separates the tuning of the conventional GA parameters from the tuning of the nabla Caputo fractional difference-based adaptive mutation mechanism.

Because the routing objective is minimization, the smaller-the-better Taguchi signal-to-noise ratio is used, which is given by

$$SNR = -10 \log_{10} \left(\frac{1}{r} \sum_{i=1}^r y_i^2 \right), \quad (14)$$

where y_i is the routing cost observed in run i , and r is the number of repeated runs. The factor levels associated with the highest SNR are selected as the most robust settings [21].

3.4. CaputoGA-TDOE algorithm

Combining Subsections 3.1, 3.2, and 3.3, the proposed CaputoGA-TDOE algorithm is outlined as shown in Algorithm 1.

Algorithm 1 CaputoGA-TDOE for Solving CARP

Phase I: GA parameter tuning using L9

- 1: Define the candidate levels of t_{max} , p_{total} , and μ_0 and construct the L9 orthogonal array.
- 2: For each L9 row, execute the conventional GA for $r = 5$ repeated runs using the GA parameter configuration.
- 3: Record the routing cost from each repeated run and calculate SNR.
- 4: Select the robust GA settings corresponding to the highest SNR.

Phase II: Caputo parameter tuning using L16

- 5: Fix the GA parameters at the settings selected in Phase I.
- 6: Define the candidate levels of α , W , λ , and μ_{max} , and construct the L16 orthogonal array.
- 7: For each L16 row, execute CaputoGA for $r = 5$ repeated runs using the selected GA settings and the corresponding nabla Caputo fractional-memory parameter configuration.
- 8: Record the routing cost from each repeated run and calculate SNR.
- 9: Select the robust nabla Caputo fractional-memory settings corresponding to the highest SNR.

Phase III: Final CaputoGA execution using the selected settings

- 10: Initialize the population using random permutations of demand locations.
 - 11: Decode each chromosome into feasible routes and evaluate its routing cost.
 - 12: Identify the initial best chromosome C_{best} and record its cost as f_{best} .
 - 13: Initialize the fitness history H using the initial best fitness value.
 - 14: **for** $t = 1$ to t_{max} **do**
 - 15: Evaluate all chromosomes and sort them based on routing cost.
 - 16: Append the current best fitness value to H and retain only the latest W fitness values.
 - 17: Compute Equation (11).
 - 18: Compute Equation (13).
 - 19: Update μ_t using Equation (12).
 - 20: Preserve the elite chromosome(s) from the current population.
 - 21: Generate offspring using tournament selection, order crossover, and inversion mutation.
 - 22: Decode and evaluate the offspring subject to the CARP feasibility constraints.
 - 23: Replace the current population with the new population.
 - 24: Update C_{best} and f_{best} if a better feasible solution is found.
 - 25: **end for**
 - 26: Return C_{best} and f_{best} .
-

For the CaputoGA-TDOE implementation, this study uses the following L9 and L16 settings for GA and nabla Caputo fractional difference-based mechanism, respectively (Table 1). These values are selected based on the preliminary run that balances the program execution time and solution quality.

Table 1. Taguchi design of experiment with L9 and L16 settings

	L9 Setting		L16 Setting
t_{max}	(400, 600, 800)	α	(0.15, 0.25, 0.35, 0.45)
p_{total}	(120, 140, 160)	W	(20, 30, 40, 50)
μ_0	(0.0025, 0.0050, 0.0075)	λ	(0.10, 0.20, 0.30, 0.40)
		μ_{max}	(0.02, 0.04, 0.06, 0.08)

4. Interpretations through the lens of nonlinear dynamical systems

This section provides interpretations of the proposed CaputoGA algorithm based on nonlinear dynamical systems. Let the conventional GA dynamics be expressed as

$$\mathcal{P}_{t+1} = \Phi(\mathcal{P}_t, \mu, \xi_t), \quad (15)$$

where $\mathcal{P}_t$ and $\mathcal{P}_{t+1}$ are the current and next-generated population respectively, μ is the fixed mutation rate, and ξ_t is the stochastic variation induced by GA operations such as selection, crossover, and mutation.

And then, let $\Omega = \{C = [g_1, g_2, \dots, g_n]\}$ denote the permutation search space of all candidate chromosomes, where each chromosome C represents a unique ordering of demand locations. At generation t , the population generated is expressed as

$$\mathcal{P}_t = \{C_{(1,t)}, C_{(2,t)}, \dots, C_{(p_{total},t)}\} \subset \Omega. \quad (16)$$

Then, let $\mathcal{D} : \Omega \rightarrow \mathcal{R}$ be the route-decoding operator that transforms $C_{(p,t)}$ into a feasible route, and let $\mathcal{J} : \mathcal{R} \rightarrow \mathbb{R}^+$ be the routing cost function. The fitness of $C_{(p,t)}$ is therefore

$$F(C_{(p,t)}) = \mathcal{J}(\mathcal{D}(C_{(p,t)})). \quad (17)$$

From equations (16) and (17), one can obtain the best fitness value given by

$$f_t = \min_{C_{(p,t)} \in \mathcal{P}_t} F(C_{(p,t)}). \quad (18)$$

The evolutionary transition from $\mathcal{P}_t$ to $\mathcal{P}_{t+1}$ can then be represented as

$$\mathcal{P}_{t+1} = \Phi(\mathcal{P}_t, \mu_t, \xi_t), \quad (19)$$

where μ_t is the adaptive mutation rate.

Following Vose's population-dynamics theory of GA [22], equation (19) can be interpreted as an iterative transformation of population states across generations. The transition operator $\Phi(\cdot)$ is nonlinear because the distribution of $\mathcal{P}_{t+1}$ depends on $\mathcal{P}_t$ through the fitness-based selection and recombination among $C_{p,t}$. Hence, equation (19) defines a discrete-time nonlinear population dynamical process.

For the proposed CaputoGA, the value of μ_t is dynamically adjusted using equation (12). Therefore, equation (19) extends the conventional GA dynamics shown

in equation (15) into a stochastic, nonlinear, and memory-dependent evolutionary dynamical system represented by equation (19).

5. Results and discussion

To evaluate the performance of the proposed CaputoGA-TDOE, this study uses benchmark instances from the Augerat dataset, specifically Set A and Set B. These instances are selected because they are used in ambulance routing optimization research [7, 8]. For comparative evaluation, two hybrid metaheuristic algorithms are considered, namely the K-Means-SA-TS algorithm by Zidi et al. [8] and the PA-PSO algorithm by Tlili et al. [7]. This study uses the results of the conventional GA reported in [7] as the baseline. Due to the limitations of reported results, the comparative evaluation focuses only on the best routing costs available in the literature. Then, for the analysis approach, this study employs descriptive statistics, the Friedman test, and the Holm-adjusted post-hoc test.

The following are the tabulated results of CaputoGA-TDOE that are compared side by side with reported results by K-Means-SA-TS, PA-PSO, and GA (Table 2), percentage gap from best-known cost (Table 3), descriptive statistics comparison (Table 4), the result of the Friedman test (Table 5), and finally, the result of the Holm-adjusted post-hoc test (Table 6).

Table 2. Total routing cost comparison

Instance	Best-Known	CaputoGA-TDOE	K-Means-SA-TS	PA-PSO	GA
A-n32-k5	784	829	869	950	957
A-n33-k5	661	688	724	765	781
A-n33-k6	742	773	798	835	798
A-n34-k5	778	836	802	920	923
A-n36-k5	799	836	863	891	1019
A-n37-k5	669	711	746	800	959
A-n37-k6	949	1047	970	1135	1115
A-n38-k5	730	863	765	892	974
A-n39-k5	822	890	931	999	1095
A-n39-k6	831	910	1046	972	257
A-n45-k6	944	1243	991	1171	1452
A-n45-k7	1146	1247	1291	1385	1397
B-n31-k5	672	687	686	696	736
B-n34-k5	788	832	809	936	865
B-n35-k5	955	976	1002	1077	1263
B-n38-k6	805	840	941	888	925
B-n39-k5	549	571	612	718	744
B-n41-k6	829	889	834	882	1105
B-n43-k6	742	762	755	874	1047
B-n44-k7	909	1016	965	1037	1126

As shown in Table 2, CaputoGA-TDOE demonstrated competitive performance across the 20 selected benchmark instances. CaputoGA-TDOE produced lower

routing costs in 11 out of 20 instances, while K-Means-SA-TS obtained lower costs in the remaining nine instances. CaputoGA-TDOE also outperformed PA-PSO in 18 out of 20 instances and GA in 19 out of 20 instances. The only exception in the comparison with GA was A-n39-k6, for which GA reported a routing cost of 257. This value should be verified because it is substantially lower than the best-known cost of 831 reported in [7], suggesting a possible inconsistency in the reported result or problem constraints. Among the instances where K-Means-SA-TS performed better than CaputoGA-TDOE were A-n34-k5, A-n37-k6, A-n38-k5, A-n45-k6, B-n31-k5, B-n34-k5, B-n41-k6, B-n43-k6, and B-n44-k7. Overall, the results indicate that the proposed CaputoGA-TDOE improves the search performance of the conventional GA and is competitive with the established hybrid metaheuristics considered in this study. Nevertheless, its performance varies across benchmark instances, indicating that no single method consistently provides the lowest routing cost for all problem instances.

Table 3. Percentage gap from the best-known total routing cost

Instance	Best-Known	CaputoGA-TDOE	K-Means-SA-TS	PA-PSO	GA
A-n32-k5	784	5.79 %	10.84 %	21.17 %	22.07 %
A-n33-k5	661	4.20 %	9.53 %	15.73 %	18.15 %
A-n33-k6	742	4.30 %	7.55 %	12.53 %	7.55 %
A-n34-k5	778	7.48 %	3.08 %	18.25 %	18.64 %
A-n36-k5	799	4.70 %	8.01 %	11.51 %	27.53 %
A-n37-k5	669	6.36 %	11.51 %	19.58 %	43.35 %
A-n37-k6	949	10.42 %	2.21 %	19.60 %	17.49 %
A-n38-k5	730	18.35 %	4.79 %	22.19 %	33.42 %
A-n39-k5	822	8.33 %	13.26 %	21.53 %	33.21 %
A-n39-k6	831	9.62 %	25.87 %	16.97 %	−69.07 %
A-n45-k6	944	31.72 %	4.98 %	24.05 %	53.81 %
A-n45-k7	1146	8.83 %	12.65 %	20.86 %	21.90 %
B-n31-k5	672	2.31 %	2.08 %	3.57 %	9.52 %
B-n34-k5	788	5.59 %	2.66 %	18.78 %	9.77 %
B-n35-k5	955	2.29 %	4.92 %	12.77 %	32.25 %
B-n38-k6	805	4.46 %	16.89 %	10.31 %	14.91 %
B-n39-k5	549	4.02 %	11.48 %	30.78 %	35.52 %
B-n41-k6	829	7.28 %	0.60 %	6.39 %	33.29 %
B-n43-k6	742	2.70 %	1.75 %	17.79 %	41.11 %
B-n44-k7	909	11.79 %	6.16 %	14.08 %	23.87 %

Table 3 presents the percentage gap between the total routing cost obtained by each algorithm and the corresponding best-known cost. A smaller percentage gap indicates better solution quality with respect to the best-known cost. Overall, CaputoGA-TDOE and K-Means-SA-TS produced the smallest percentage gaps across most of the selected benchmark instances, while PA-PSO and GA generally showed larger deviations from the best-known costs. For the selected instances and the literature-reported comparator values used in this study, CaputoGA-TDOE and K-Means-SA-TS therefore demonstrate the most competitive performance among the compared algorithms. The unusual A-n39-k6 result is due to the percentage-gap

calculation using the reported GA cost of 257, which is substantially lower than the corresponding best-known cost [7] and is therefore interpreted cautiously.

Table 4. Descriptive statistics of the percentage gap

Method	Mean Gap	Gap Deviation
CaputoGA-TDOE	8.03 %	6.76 %
K-Means-SA-TS	8.04 %	6.20 %
PA-PSO	16.92 %	6.30 %
GA	21.42 %	24.53 %

The descriptive statistics in Table 4 further support the obtained results. CaputoGA-TDOE records the lowest mean percentage gap of 8.03 %, closely followed by K-Means-SA-TS at 8.04 %, indicating that both methods achieve comparable solution quality across the selected benchmark instances. However, K-Means-SA-TS exhibits a slightly lower gap deviation of 6.20 % compared with 6.76 % for CaputoGA-TDOE, suggesting marginally more consistent performance across the instances. These descriptive results apply to the 20 selected benchmark instances and the comparator values available from the cited literature.

Table 5. Friedman test result

Statistic	p -value	Kendall's W
34.39	1.64×10^{-7}	0.57

To determine whether these observed percentage gap differences are statistically meaningful, a Friedman test is conducted, as reported in Table 5. The test result is statistically significant, with a Friedman statistic of 34.39 and a p -value of 1.64×10^{-7} , which is far smaller than the significance level of 0.05. Therefore, it can be said that the percentage gap differences among the four compared algorithms are not attributable to random variation alone. In addition, the calculated Kendall's W indicates a significant effect size. In other words, there is a strong degree of separation in the rank-based performance of the algorithms across instances. This inference is limited to the selected benchmark set and the available results used in the comparison.

Table 6. Holm-adjusted post-hoc test result

Method 1	Method 2	Holm-adjusted p -value
CaputoGA-TDOE	PA-PSO	0.0004
K-Means-SA-TS	PA-PSO	0.0024
CaputoGA-TDOE	GA	0.0028
K-Means-SA-TS	GA	0.0076
PA-PSO	GA	0.0344
CaputoGA-TDOE	K-Means-SA-TS	0.7562

Following the significant Friedman test, a Holm-adjusted post-hoc test is conducted to identify which algorithm pairs differ significantly, as shown in Table 6. The results show that CaputoGA-TDOE significantly outperforms PA-PSO and GA at the 0.05 significance level. Likewise, K-Means-SA-TS significantly outperforms

both PA-PSO and GA. However, the comparison between CaputoGA-TDOE and K-Means-SA-TS is not statistically significant, with a Holm-adjusted p-value of 0.7562. This finding is consistent with the descriptive results, where the two methods exhibit very similar mean percentage gaps. Because the comparator values were obtained from the corresponding publications rather than from a common reimplementation, these statistical comparisons should be interpreted as an analysis of the available benchmark results rather than a fully controlled head-to-head experimental study.

6. Conclusions

This study presented the CaputoGA-TDOE algorithm and provided interpretations through the lens of nonlinear dynamical systems. The experiment using the selected benchmark instances shows that CaputoGA-TDOE is a highly competitive metaheuristic approach for solving CARP. This study found that embedding the nabla Caputo fractional difference into the GA evolutionary process can improve its solution quality via the adaptive mutation mechanism. By overcoming the parameter tuning challenge, the effectiveness of the TDOE technique is demonstrated. The present results provide initial empirical support for the argument that the nabla Caputo fractional-memory dynamics with statistically guided parameter tuning can be beneficial in evolutionary optimization for CARP. The cross-method evaluation is based partly on results reported in previous studies and therefore does not constitute a fully controlled reimplementation of all competing algorithms under identical initial populations and computational conditions. Future work will consider such a controlled comparative implementation in addition to investigating other variants of the Caputo fractional difference and the theory of memory-aware mechanisms in evolutionary optimization.

Acknowledgments

This research is funded by Malaysian Ministry of Higher Education under the research grant FRGS-EC/1/2024/STG06/UMS/02/5.

References

- [1] Riyapan, S., Sanyanuban, P., Chantanakomes, J., Roongsaenthong, P., Somboonkul, B., Rangabpai, W., et al. (2024). Enhancing survival outcomes in developing emergency medical service system: Continuous quality improvement for out-of-hospital cardiac arrest. *Resuscitation Plus*, 19, 100683.

- [2] Huabangyang, T., Klaiaungthong, R., Jansanga, D., Aintharasongkho, A., Hanlakorn, T., Sakcharoen, R., et al. (2021). Survival rates and factors related to the survival of traffic accident patients transported by emergency medical services. *Open Access Emergency Medicine*, 13, 575-586.
- [3] Tassone, J., & Choudhury, S. (2020). A comprehensive survey on the ambulance routing and location problems. *arXiv preprint arXiv:2001.05288*.
- [4] Marinakis, Y., Marinaki, M., & Migdalas, A. (2025). Particle swarm optimization for the vehicle routing problem: A survey and a comparative analysis. In R. Martí, P.M. Pardalos, & M.G.C. Resende (Eds.), *Handbook of Heuristics* (pp. 1563-1596). Springer.
- [5] Tlili, T., Ben Nasser, S., Chicano, F., & Krichen, S. (2024). Tabu search-based hyper-heuristic for solving the heterogeneous ambulance routing problem with time windows. *International Journal of Intelligent Transportation Systems Research*, 22(2), 446-461.
- [6] Rahman, M.A., Sokkalingam, R., Othman, M., Biswas, K., Abdullah, L., & Abdul Kadir, E. (2021). Nature-inspired meta heuristic techniques for combinatorial optimization problems: Overview and recent advances. *Mathematics*, 9(20), 2633.
- [7] Tlili, T., Harzi, M., & Krichen, S. (2017). Swarm-based approach for solving the ambulance routing problem. *Procedia Computer Science*, 112, 350-357.
- [8] Zidi, I., Al-Omani, M., & Aldhafeeri, K. (2019). A new approach based on the hybridization of simulated annealing algorithm and tabu search to solve the static ambulance routing problem. *Procedia Computer Science*, 159, 1216-1228.
- [9] Taguchi, G. (1986). *Introduction to Quality Engineering: Designing Quality into Products and Processes*. Asian Productivity Organization.
- [10] Mosayebi, M., & Sodhi, M. (2020). Tuning genetic algorithm parameters using design of experiments. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1937-1944).
- [11] Uray, E., Carbas, S., Geem, Z.W., & Kim, S. (2022). Parameters optimization of Taguchi method integrated hybrid harmony search algorithm for engineering design problems. *Mathematics*, 10(3), 327.
- [12] Fan, F., Liu, G., Geng, J., Zhao, H., & Liu, G. (2022). Optimization of remote sensing image segmentation by a customized parallel sine cosine algorithm based on the Taguchi method. *Remote Sensing*, 14(19), 4875.
- [13] Muthana, S.A., & Ku-Mahamud, K.R. (2023). Taguchi-Grey relational analysis method for parameter tuning of multi-objective Pareto ant colony system algorithm. *Journal of Information & Communication Technology*, 22(2), 149-174.
- [14] Al Rafei, T., Yousfi Steiner, N., & Chrenko, D. (2023). Genetic algorithm and Taguchi method: An approach for better Li-ion cell model parameter identification. *Batteries*, 9(2), 72.
- [15] Farahbakhsh, A., & Kheirikhah, A.S. (2023). A new efficient genetic algorithm-Taguchi-based approach for multi-period inventory routing problem. *International Journal of Research in Industrial Engineering*, 12(4), 397-413.
- [16] Cetin, T., Ilgin, M.A., & Yurdusev, M.A. (2024). Selection of genetic algorithm parameters for optimization of storm-sewer networks using Taguchi method. *Karaelmas Science and Engineering Journal*, 14(1), 1-10.
- [17] Katoch, S., Chauhan, S., & Kumar, V. (2021). A review on genetic algorithm: Past, present, and future. *Multimedia Tools and Applications*, 80, 8091-8126.
- [18] Chew, J.V.L. (2025). Hybrid genetic algorithm with Caputo fractional derivative for ambulance routing problems. In T.T. Quan, C. Sombatheera, H.A. Pham, & N.T. Tran (Eds.), *Multi-disciplinary trends in artificial intelligence: 18th International Conference, MIWAI 2025* (pp. 317-327). Springer.

- [19] Fernandez, E., Huilcapi, V., Birs, I., & Cajo, R. (2025). The role of fractional calculus in modern optimization: A survey of algorithms, applications, and open challenges. *Mathematics*, 13(19), 3172.
- [20] Abdeljawad, T. (2013). On delta and nabla Caputo fractional differences and dual identities. *Discrete Dynamics in Nature and Society*, 2013, 406910.
- [21] Rashid, K.M.J. (2023). Optimize the Taguchi method, the signal-to-noise ratio, and the sensitivity. *International Journal of Statistics and Applied Mathematics*, 8(6), 64-70.
- [22] Vose, M.D. (1999). *The Simple Genetic Algorithm: Foundations and Theory*. MIT Press.