mathematics

MDPI

Article

A Memetic Genetic Algorithm Framework for Public Transit Route Optimization: The Genetic Scissors Method

Kadir Akgol * and Mucahit Kayis

Department of Civil Engineering, Faculty of Engineering, Giresun University, Giresun 28200, Turkey; mucahitkayis@outlook.com

* Correspondence: kadir.akgol@giresun.edu.tr

Abstract

This study proposes a memetic genetic algorithm framework, termed Genetic Scissors, for improving the spatial directness of public transit routes based on the rationality measure. The framework is formulated on an extended problem structure and incorporates a same-location constraint that prevents stops sharing identical geographic coordinates from being selected consecutively within the same route, together with an enhanced repair procedure that preserves the positional integrity of route start and end stops. It combines three components: a two-phase fitness structure that resolves the scale incompatibility between absolute distance and normalized rationality, six original intensification operators acting at the node, segment, block, and in-route evolutionary search levels, and a regression-based adaptive parameterization scheme responsive to problem size. Experimental analyses on 213 normal clusters derived from the Istanbul public transit network, together with evaluations on a 109-cluster fair-comparison subsample, show that the proposed method improves the existing routes in 81.7% of cases, with a mean rationality improvement of 2.41%, and yields a highly significant Wilcoxon signed-rank effect ($p < 0.001$, $r = 0.750$). In direct comparison, Genetic Scissors outperforms the reference method in 84 of 109 clusters ($p < 0.001$, $r = 0.714$).

Keywords: route optimization; memetic genetic algorithm; rationality; two-phase fitness; public transit; ablation analysis

MSC: 90C27; 90B06

1. Introduction

Public transit route design is one of the principal decision domains that shapes not only the physical structure of the transportation network but also service accessibility, operational efficiency, and user experience [1]. The sequence of stops along a route and the assignment of these stops to routes determine the spatial organization of the network and play a decisive role in the directness of passenger movements [2,3]. In practice, however, transit routes are rarely formed under ideal optimization conditions. Instead, they typically emerge through the cumulative effects of historical development, fragmented responses to growing demand, and operational constraints. As a result, the resulting route structures cannot always be assumed to represent optimal configurations. For this reason, the evaluation and improvement of existing routes through quantitative criteria has become an important line of research in transportation planning [4].

Check for updates

Academic Editor: Hongguang Ma

Received: 27 April 2026
Revised: 7 May 2026
Accepted: 28 May 2026
Published: 30 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Most studies on route optimization assess solution quality through single criteria such as total distance, total travel time, or operating cost [5]. Although this approach is functional for many applications, it is insufficient for public transit systems. A route may be short without being direct. In particular, in multi-route networks, when the spatial distribution of stops and inter-route interactions are considered jointly, geometrically indirect route structures may arise even if aggregate distance measures appear acceptable [6]. In this context, rationality, defined as the ratio of traveled network distance to Euclidean distance, has emerged as an important indicator for evaluating the spatial directness of public transit routes [7]. Nevertheless, the literature still tends to treat this measure as a secondary performance indicator, and studies that place it at the center of the optimization process remain limited.

As the numbers of stops and routes increase, public transit route optimization generates a rapidly expanding combinatorial search space and therefore bears conceptual similarities to difficult optimization problems such as the Traveling Salesman Problem and the Vehicle Routing Problem [8]. For this reason, metaheuristic methods, and particularly genetic algorithm-based approaches, have become widely used practical solution tools in this area. In the TSP literature, Applegate, et al. [9] provided a comprehensive computational treatment of the problem and demonstrated the necessity of heuristic search, whereas Khan and Maiti [10] showed that a swap-sequence-based artificial bee colony algorithm could achieve strong performance on TSP instances. In the VRP literature, the seminal study of Baker and Ayechew [11] demonstrated that a genetic algorithm could produce competitive results against tabu search and simulated annealing on the classical VRP, while Prins [12] proposed a simple yet effective evolutionary algorithm that achieved new best-known solutions on numerous benchmark instances. In the public transit context, the problem has been extensively studied under the framework of the transit route network design problem. According to the systematic review by Iliopoulou, et al. [13], the genetic algorithm is among the most frequently used metaheuristics in this problem class. Bielli, et al. [14] applied GA-based optimization to bus network design in a structured manner; Pattnaik, et al. [15] proposed a two-stage procedure in which candidate routes are first generated and then selected through a GA; and Arbex and da Cunha [16] developed an alternating-objective GA that makes the management of multiple optimization targets more tractable. Taken together, these studies show that changing the optimization target during the search process has become a well-established idea in the literature. Within this body of work, Akgol, Demir, Aydogdu and Murat [7] constitute an important reference point by placing rationality at the core of the optimization process through a GA structure equipped with flow-direction-oriented operators and by showing that existing Istanbul bus routes can be improved accordingly.

In evolutionary computation, the enhancement of global search through local improvement has long constituted a productive line of research. Memetic algorithms, originally proposed by Moscato [17], combine population-based global exploration with local improvement strategies and can therefore produce higher-quality solutions than purely evolutionary approaches [18,19]. Nguyen, et al. [20] emphasized that the balance between exploration and exploitation is a decisive factor in memetic algorithms, while Radcliffe and Surry [21] showed that solutions generated after crossover can be transformed into higher-quality structures through local improvement procedures. Subsequent studies on memetic GAs include multi-parent crossover for TSP [22], hill climbing within GA cycles [23], and Lamarckian/Baldwinian integration of local search [24]. Nested GA architectures offer a means of optimizing different decision layers as interconnected sub-processes [25–28]. For example, Lightner-Laws, Agrawal, Lightner and Wagner [25] treated driver assignment in an outer loop and routing in an inner loop for the multi-depot vehicle routing problem,

whereas Kouretas and Kepaptsoglou [26] linked modal assignments in an outer loop to routing decisions in an inner loop within a GIS-supported framework. Decomposition-based approaches improve performance in large-scale problems by partitioning the search space into manageable subproblems. Santini, et al. [29] systematically examined alternative decomposition strategies for vehicle routing heuristics and showed the advantages of route-based decomposition for large problems, while Debels and Vanhoucke [30], Zhang, et al. [31], and Kerscher and Minner [32] demonstrated the contribution of decomposition-based structures in several problem classes. Multi-level GA structures, in turn, treat the evolutionary process hierarchically at the individual and group levels [33–37], and studies such as Akbari and Ziarati [33] and Grudniewski and Sobey [34] have shown their effectiveness in avoiding local optima. More recently, approaches inspired by biological processes have also been used to reinterpret evolutionary operators. Sohrabi, et al. [38] proposed a genetics-inspired algorithm to alleviate premature convergence, whereas Meniz [39] introduced a TOPSIS-based dynamic decision mechanism for selection operators.

Furthermore, recent studies have demonstrated the growing applicability of evolutionary and memetic algorithms across various complex optimization tasks. For instance, the integration of genetic algorithms with local search methods has proven highly effective in fundamental path optimization problems [40], while advanced memetic architectures have been successfully deployed for trajectory optimization in automated operations [41]. Within the specific domain of public transportation, there is a strong trend toward utilizing data-driven frameworks and evolutionary approaches to address sustainable route selection [42], customized bus service design [43], and the integrated optimization of feeder transit routes [44]. These recent advancements underscore the necessity of developing specialized memetic structures, such as the proposed Genetic Scissors framework, to effectively handle domain-specific constraints in large-scale transit networks.

In the broader context of evolutionary routing, recent state-of-the-art studies have addressed operational complexities and uncertainties in vehicle routing problems. Game-theoretic evolutionary algorithms have been proposed for the Vehicle Routing Problem with Time Windows (VRPTW) under uncertain travel times [45], data-driven evolutionary frameworks have been developed for dynamic VRPTW under limited computational budgets [46], and surrogate-assisted bi-level evolutionary algorithms have been applied to multi-depot VRP with uncertain demand [47]. Although these studies represent important advances in operational routing, the focus of the present study structurally diverges from classical VRPs. Rather than addressing dynamic operational constraints such as time windows, fleet capacities, or demand uncertainty, the present study targets the spatial directness (rationality) of existing multi-route transit networks under rigid topological constraints, specifically the same-location penalty and strict positional integrity of route boundaries. The proposed Genetic Scissors framework is therefore positioned not as a generic routing solver, but as a specialized memetic architecture designed for the multi-scale geometric interventions required by this extended problem.

Despite these advances, two major limitations remain apparent in the existing literature. First, when the search process is based on a single-stage objective function, improvement in one performance criterion may come at the expense of other performance dimensions. Second, in memetic or hybrid approaches, the questions of which candidate solutions should be improved locally, with what intensity, and under which computational budget are rarely addressed in a systematic manner. In addition, the flow-direction operators proposed by Akgol, Demir, Aydogdu and Murat [7] are not designed to handle the extended problem structure considered in the present study. This extended structure introduces two additional challenges. The first is that, in real network data, stops belonging to different routes may share identical coordinates, making it necessary to prevent such

nodes from being selected consecutively within the same route. The second is that the start and end stops of each route must be preserved not merely at the pairing level, but with full positional integrity. These two constraints make the problem both more realistic and less directly compatible with existing methods.

The main objective of this study is to propose a memetic optimization framework that addresses these three issues in an integrated manner: an objective structure that does not rely on a single criterion, a selective and controlled local improvement mechanism, and a constraint design compatible with the extended problem formulation. The proposed Genetic Scissors (GS) framework combines six original intensification operators applied to selected elite individuals with a compact GA component that performs an independent evolutionary search over selected route segments. In this structure, the inner evolutionary process is not a compulsory computational layer executed for every candidate solution, as in classical nested GAs, but a targeted intervention applied only to selected individuals with improvement potential. In this way, the memetic nature of combining global search with local refinement is retained, while computational cost is controlled without being spread across the entire population.

The contributions of the paper are threefold. First, a two-phase fitness function transitions between the absolute distance measure (pt) and the normalized rationality measure (rat) through a problem-size-sensitive meta-parameter, addressing the scale incompatibility that is left unresolved by conventional alternating-objective schemes [16]. Second, the Genetic Scissors framework introduces six original intensification operators (GS-SR, GS-FR, GS-CX, GS-BT, GS-BI, GS-NLS) and an inner evolutionary search component (GS-IRS), all designed for the extended constraint structure and unified within a single memetic architecture. Third, a regression-based adaptive parameterization scheme determines the upper-level control parameters in a problem-size-sensitive manner through regression analysis on representative clusters.

The proposed framework was systematically evaluated on 109 clusters derived from the Istanbul public transit network. The experimental results show that Genetic Scissors improved rationality in 81.7% of the clusters, with an average improvement of 2.41%. The Wilcoxon signed-rank test indicated that this improvement was highly significant ($p < 0.001$, $r = 0.750$). In direct comparison with the reference method, the proposed approach outperformed the baseline method in 84 of the 109 clusters, and this superiority was again found to be statistically significant ($p < 0.001$, $r = 0.714$). In addition, the scale-invariant performance analysis with respect to stop count and route count showed that Genetic Scissors remains stable across problem scales, successfully avoiding the performance degradation observed in the reference method, particularly in large and multi-route networks.

The remainder of the paper is organized as follows. The next section presents the formal mathematical formulation of the problem and establishes the extended constraint structure. The following section introduces the Genetic Scissors framework, including the mathematical definitions of the six intensification operators and the corresponding pseudocode blocks. The adaptive parameterization mechanism based on regression analysis is then presented, followed by the section describing the experimental design and the statistical testing framework. The subsequent results section reports the ablation analysis, the comparative validation on 109 clusters, and the visual evaluation of route structures. The final sections discuss the limitations of the proposed method and conclude the paper by summarizing the main findings and outlining directions for future research.

2. Problem Formulation

This section presents the formal mathematical formulation of the public transit route optimization problem considered in this study. The dataset and the structural setup of the problem are introduced first, followed by the coordinate transformation and the distance matrices. The objective functions are then defined, after which the extended constraint system is formulated through set-based notation. The section concludes with the formal statement of the optimization problem.

2.1. Dataset and Problem Setup

The study is based on the dataset used by Akgol, Demir, Aydogdu and Murat [7]. This dataset contains the ordered geographic coordinates of the existing stops of 732 bus routes in Istanbul and is partitioned into 250 spatial clusters. Each cluster represents a multi-route subnetwork composed of between one and ten routes.

For a given cluster $c \in C$, let the cluster composed of k routes be defined as

$$c = \{H_1, H_2, \ldots, H_k\},\ |H_i| = n_i,\ N = \sum_{i=1}^{k} n_i \tag{1}$$

where H_i denotes the i-th route in the cluster, n_i denotes the number of stops on that route, and N denotes the total number of stops in the cluster. The problem consists of reassigning all intermediate stops within the cluster to the routes and determining their within-route ordering, while keeping the positions of the start and end stops of each route fixed.

2.2. Coordinate Transformation and Distance Matrices

2.2.1. Global Numbering and Coordinate Transformation

All stops in the cluster are assigned global identities from 1 to N, independently of route membership. The set of global indices associated with route i is defined as

$$G(i) = \{s(i), s(i)+1, \ldots, s(i)+n_i-1\} \tag{2}$$

where $s(1) = 1$, and for $i > 1$, $s(i) = 1 + \sum_{m=1}^{i-1} n_m$.

To enable metric distance calculations from raw geographic coordinates, a local planar transformation based on the Mercator approximation is applied. The arithmetic mean of all stop coordinates in the cluster, (λ_0, φ_0), is taken as the reference point, and for each stop the transformed coordinates are computed as

$$x_i = (\lambda_i - \lambda_0)\cos(\varphi_0) \cdot R \tag{3}$$

$$y_i = (\varphi_i - \varphi_0) \cdot R \tag{4}$$

where (λ_i, φ_i) denotes the longitude and latitude of stop i in radians, $R = 6,371,000$ m is the Earth's radius, and (x_i, y_i) denotes the corresponding metric coordinates in the local plane. Because the spatial extent of each cluster is limited, the distortion introduced by this transformation remains negligible.

2.2.2. Distance Matrices

Two distance matrices are defined for the problem. The Euclidean distance matrix, $EM \in \mathbb{R}^{N\times N}$, represents the direct geometric distance between pairs of stops, whereas the route distance matrix, $DSM \in \mathbb{R}^{N\times N}$, represents the cost of successive movements on the network. The raw distance between stops i and j is defined as

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{5}$$

where d_{ij} denotes the Euclidean distance between stops i and j, and $(x_i - y_i)$ and $(x_j - y_j)$ denote their metric coordinates in the local plane, as defined in Equations (3) and (4).

In real-world data, the same physical location may be represented multiple times, or different records may correspond to coordinates that are extremely close to one another. Such cases can produce artificially short distances and may therefore mislead the optimization process. To prevent this effect, a penalty is imposed on the route distance matrix as follows:

$$\mathrm{DSM}_{ij} = \begin{cases} M, & d_{ij} < 1 \text{ m} \\ d_{ij}, & \text{otherwise} \end{cases} \tag{6}$$

Here, $M = 10^6$ is a large penalty constant. Unlike the penalized route distance matrix defined in Equation (6), the Euclidean distance matrix EM is preserved as the raw geometric reference, because the direct distance appearing in the denominator of the rationality measure must retain its geometric meaning

To represent the relative importance of stop pairs, an $N \times N$ trip-weight matrix TM is defined. In the absence of observed demand data, equal weight is assigned to all stop pairs:

$$\mathrm{TM}_{ij} = 1, \ \forall i, j \in \{1, 2, \ldots, N\} \tag{7}$$

This assumption ensures methodological simplicity and comparability across clusters. In applications where observed demand data are available, the same matrix structure can be retained while populating TM according to the corresponding demand levels.

2.3. Objective Functions

Two objective functions are defined for the problem. The first is the total path length, denoted by $pt(g)$, which gives the sum of the route distances between successive stop pairs along all routes in a solution:

$$pt(g) = \sum_{i=1}^{k} \sum_{j=1}^{n_i - 1} \mathrm{DSM}_{g(i,j),g(i,j+1)} \tag{8}$$

where $g(i, j)$ denotes the global identity of the stop located at position j on route i in solution g, k is the number of routes in the cluster, and n_i is the number of stops assigned to route i.

The second objective is rationality, denoted by $rat(g)$, which is defined as the weighted average of the ratio between route distance and Euclidean distance, following Akgol, Demir, Aydogdu and Murat [7]:

$$rat(g) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{TM}_{ij} \cdot \frac{\mathrm{DSM}_{ij}}{\mathrm{EM}_{ij}}}{\sum_{i=1}^{N} \sum_{j=1}^{N} \mathrm{TM}_{ij}} \tag{9}$$

where TM_{ij} denotes the weight assigned to the stop pair (i, j), DSM_{ij} denotes the route-based distance between stops i and j, and EM_{ij} denotes the corresponding Euclidean distance. As $rat(g)$ approaches 1, the route structure becomes geometrically more direct, whereas larger values indicate more indirect movement patterns between stop pairs. In the literature, this measure is also referred to as **circuity** or the **directness index**.

There is an important scale difference between the two objective functions. While $pt(g)$ is an absolute distance measure, $rat(g)$ is a dimensionless ratio. This difference makes a standard weighted-sum combination problematic and motivates the two-phase fitness structure introduced in the next section.

2.4. Extended Constraint System

The problem considered in this study extends the reference formulation by introducing two additional constraints. Let $g = (g_1, g_2, \ldots, g_N)$ denote the solution vector. The feasible solution space $\mathcal{F}$ is defined as the intersection of the following four constraint sets.

The first is the permutation constraint. The solution must be a permutation defined on the set $\{1, 2, \ldots, N\}$:

$$\mathcal{F}_1 = \left\{ g \in \mathbb{Z}^N : \{g_1, \ldots, g_N\} = \{1, 2, \ldots, N\} \right\} \tag{10}$$

Equation (10) states that each stop must appear exactly once in the solution vector and that no stop may be omitted or repeated.

The second is the start-end positional constraint. For each route i, the start stop S_i and the end stop E_i must remain fixed at the corresponding block positions of the chromosome:

$$\mathcal{F}_2 = \left\{ g \in \mathbb{Z}^N : g_{s(i)} = S_i \text{ and } g_{s(i)+n_i-1} = E_i,\ \forall i \in \{1, \ldots, k\} \right\} \tag{11}$$

Here, S_i denotes the starting index of route i in the chromosome, as defined after Equation (2), and n_i denotes the number of stops assigned to route i. Unlike the reference formulation, in which this condition is enforced only at the pairing level, the present study strengthens it into a full positional integrity constraint. Accordingly, the start and end stops must not only belong to the correct route but must also occupy the exact boundary positions of the corresponding route block.

The third is the same-location constraint. In real transit data, stops belonging to different routes may share identical geographic coordinates. If such a pair is selected consecutively within the same route, it introduces an artificial advantage into the optimization process even though it does not represent an actual geometric movement. Let P denote the set of stop pairs located at the same coordinates:

$$\mathcal{P} = \{(u, v) : (x_u, y_u) = (x_v, y_v),\ u \neq v\} \tag{12}$$

where (x_u, y_u) and (x_v, y_v) denote the local planar coordinates of stops u and v, respectively, as defined in Equations (3) and (4). Under this definition, the same-location constraint is written as

$$\mathcal{F}_3 = \left\{ g \in \mathbb{Z}^N : (g(i, j), g(i, j+1)) \notin \mathcal{P},\ \forall i, \forall j \in \{1, \ldots, n_i - 1\} \right\} \tag{13}$$

In Equation (13), $g(i, j)$ denotes the global identity of the stop placed at position j on route i. The constraint therefore requires that no pair of consecutive stops on any route may belong to the set P.

The fourth is the route integrity constraint. Each stop must be assigned to one and only one route. This condition is already implied indirectly by the permutation structure in Equation (10), but it is stated explicitly here because the chromosome representation preserves the block structure of individual routes:

$$\mathcal{F}_4 = \left\{ g \in \mathbb{Z}^N : G(i) \cap G(j) = \varnothing\ \forall i \neq j,\ \bigcup_i G(i) = \{1, \ldots, N\} \right\} \tag{14}$$

Here, $G(i)$ denotes the set of global indices assigned to route i, as defined in Equation (2). Equation (14) states that the route blocks are pairwise disjoint and that their union covers the entire set of stops.

The feasible solution space is therefore given by the intersection of the four constraint sets:

$$\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \cap \mathcal{F}_3 \cap \mathcal{F}_4 \tag{15}$$

In particular, the start-end positional constraint in Equation (11) and the same-location constraint in Equation (13) together define the extended problem structure that prevents the direct application of the reference method proposed by Akgol, Demir, Aydogdu and Murat [7]. These two constraints also provide the main motivation for the operator design introduced in the next section.

2.5. Problem Statement

Under the structure defined above, the optimization problem considered in this study can be written as

$$\mathcal{F} = \mathcal{F}_1 \cap \mathcal{F}_2 \cap \mathcal{F}_3 \cap \mathcal{F}_4 \tag{16}$$

where g_c denotes the current generation counter, and $J(g, g_c)$ denotes the active fitness function. Depending on the current phase of the search, J is evaluated using either the total path length measure pt or the rationality measure rat. The mathematical definition of this phase-dependent structure is given in the next section.

3. The Genetic Scissors Framework

This section presents the main methodological contribution of the paper. It first introduces the two-phase fitness function and the main memetic genetic algorithm flow. It then explains the motivation of the Genetic Scissors framework and provides the formal definitions of the six intensification operators. Finally, it describes the in-route evolutionary search component, GS-IRS, together with the operator selection mechanism.

3.1. Two-Phase Fitness Function

As discussed earlier, the scale difference between $pt(g)$ and $rat(g)$ makes a standard weighted-sum combination unsuitable. For this reason, the fitness function is defined in a two-phase form that depends on the generation counter:

$$J(g, g_c) = \begin{cases} pt(g), & g_c \leq \theta_r \cdot n_g \\ rat(g), & g_c > \theta_r \cdot n_g \end{cases} \tag{17}$$

Here, g_c denotes the current generation counter, n_g denotes the total number of generations, and $\theta_r \in (0, 1)$ denotes the phase-switch ratio, corresponding to the parameter *ratiomode*. In this study, the value of θ_r is determined through the regression-based procedure described later.

In the first phase of the search, referred to as the distance phase, selection pressure is applied through total path length. The purpose of this phase is to construct geometrically balanced and feasible route structures. In the second phase, referred to as the rationality phase, the active objective switches to rationality minimization, and the search is directed toward more direct route geometries. Although both pt and rat are computed and monitored throughout the search process, selection pressure is imposed only through the objective associated with the active phase. The global best solution g^* is updated according to the active fitness function as follows:

$$g^* \leftarrow g \text{ if } J(g, g_c) < J(g^*, g_c) \tag{18}$$

This design differs from the alternating-objective approach of Arbex and da Cunha [16] in two respects. First, the phase-switch point is not a fixed rule but a problem-size-sensitive

meta-parameter. Second, the transition does not occur between objectives defined on the same scale, but between two fundamentally different objective types, namely an absolute distance measure and a normalized geometric ratio.

The specific sequence of this two-phase transition (pt → rat) is not arbitrary but a structural necessity. Total path length (pt) is an absolute additive metric whose gradient rapidly organizes randomly generated initial populations into geometrically coherent macro-structures by eliminating extreme detours. Rationality (rat), in contrast, is a fractional, dimensionless ratio. Optimizing a fractional ratio directly on a chaotic random initial topology generates a deceptive search landscape, since the same low rationality value can be attained by topologically distinct configurations and the algorithm easily becomes trapped in local optima. Preliminary experiments testing eleven different objective configurations on ten random clusters, varying the phase weight from 100% rationality to 100% path length, systematically confirmed that configurations dominated by early-phase rationality produced inferior final solutions, while the adopted pt-first sequence yielded the most stable and best-performing convergence. The two-phase order therefore reflects a structural builder–refiner relationship: pt establishes coherent route geometry in the early phase, and rat performs fine-grained spatial directness refinement on the resulting topology.

3.2. Main Algorithm and Chromosome Representation

A solution is represented as a feasible permutation vector under the previously defined constraint set $\mathcal{F}$. Specifically, a solution $g \in \mathbb{Z}^N$ is encoded in a route-block structure as

$$g = \left[\underbrace{S_1, d_{1,2}, \ldots, d_{1,n_1-1}, E_1}_{H_1 \text{ block}},\ \underbrace{S_2, \ldots, E_2}_{H_2 \text{ block}},\ \ldots,\ \underbrace{S_k, \ldots, E_k}_{H_k \text{ block}} \right] \tag{19}$$

where S_i and E_i denote the start and end stops of route i, respectively, and $d_{i,j}$ denotes an intermediate stop assigned to route i. The repair operator $\mathcal{R} : \mathbb{Z}^N \to \mathcal{F}$ restores feasibility after operator applications by relocating the start and end stops of each route to the block boundaries while preserving, as much as possible, the relative ordering of the remaining stops.

The overall flow of the proposed memetic genetic algorithm is summarized in Algorithm 1. In addition to the standard GA components, namely roulette-wheel selection, prefix-preserving crossover, three mutation operators, and elitism, the framework incorporates a problem-specific intensification layer built on the proposed Genetic Scissors operators. The reference AkAy operators are retained only for comparative evaluation against the baseline method of Akgol, Demir, Aydogdu and Murat [7], whereas the Genetic Scissors framework constitutes the main methodological contribution of the present study.

Algorithm 1. Two-phase memetic genetic algorithm with the Genetic Scissors framework

Input: cluster C, number of routes k, number of stops N, distance matrix DSM, Euclidean matrix EM, trip-weight matrix TM, same-location matrix S, number of generations n_g, population size ps, phase-switch ratio θ_r, mutation probabilities p_m, p_m2, p_mf, GS application probability p_mpr, GS-IRS warm-up threshold θ_mg, elite-selection probability prob_best, early-stopping patience values for distance and rationality phases
Output: best solution g*

Algorithm 1. *Cont.*

1. Build the optimization problem for cluster C
2. Generate an initial population P of size ps using random permutations
3. Repair all individuals in P to satisfy route-boundary constraints
4. Set g* as the best individual in P
5. Initialize phase-specific best values and early-stopping counters
6. For g_c = 1 to n_g do
7. If g_c ≤ θ_r · n_g and no early phase switch occurred, then
8. ActivePhase ← distance phase
9. Construct DSM_active using the current scaling level
10. Else
11. ActivePhase ← rationality phase
12. DSM_active ← DSM
13. End If
14. Evaluate all individuals in P using DSM_active, EM, TM, and S
15. Compute both distance-based and rationality-based objective values
16. Set the active objective J according to ActivePhase
17. If the current population contains a better individual than g*, then
18. Update g*
19. End If
20. Check early-stopping conditions
21. If the distance phase stagnates, switch to the rationality phase
22. If the rationality phase stagnates, terminate the search
23. Select ps parent individuals by roulette-wheel selection
24. Apply prefix-preserving crossover to generate offspring
25. Apply mutation operators with probabilities p_m, p_m2, and p_mf
26. Repair all offspring to restore feasibility
27. Apply Genetic Scissors intensification to the offspring:
28. GS-SR is applied probabilistically at every generation
29. GS-FR is applied every 25 generations
30. GS-CX is applied every 10 generations
31. GS-BT is applied every 15 generations
32. GS-BI is applied every 20 generations
33. GS-NLS is applied every 30 generations only in the distance phase
34. Insert the elite solution into the new population
35. If g_c > θ_mg · n_g, then
36. Select a target individual
37. With probability prob_best, choose the current best solution
38. Otherwise, choose a random individual from the population
39. Select one route within the target individual
40. Apply GS-IRS to the selected route
41. Repair the updated individual and return it to the population
42. End If
43. End For
44. Return g*

Algorithm 1 summarizes the proposed Genetic Scissors-based memetic genetic algorithm. The search starts with a population of feasible route assignments and evolves under a two-phase objective structure. In the first phase, the algorithm emphasizes distance minimization using a progressively relaxed active distance matrix. In the second phase,

the search directly targets rationality minimization. At the population level, selection, crossover, mutation, and repair are followed by the proposed Genetic Scissors operators, which intensify the search at different structural scales. After a warm-up period, the GS-IRS component performs an additional route-level memetic search on a selected individual. Elitism and phase-specific early-stopping rules are used to preserve the best solution and improve computational efficiency.

3.3. Genetic Scissors: Architectural Motivation

The Genetic Scissors (GS) framework emerged from the observation that, under the previously defined extended problem structure, uniform local search mechanisms are not sufficiently effective in exploring the solution space. Although the AkAy operators of the reference method in Akgol, Demir, Aydogdu and Murat [7] provide an effective intensification mechanism based on random inter-route segment exchanges, they are not sufficient to address the multi-scale intervention types required by the extended structure.

The name of the framework is an algorithmic metaphor inspired by the notion of targeted DNA cutting in biology. In particular, the idea of selective cutting and rearrangement of target sequences in the CRISPR/Cas9 mechanism described by Jinek, et al. [48] provides a conceptual analogy for the selective decomposition and reorganization of specific parts of chosen individuals. The GS framework does not attempt to model a biological system literally. Rather, it uses this analogy as a semantic basis for selective, component-level, and reversible restructuring.

The GS framework is based on a two-level intervention logic. At the first level, six rule-based operators, namely GS-SR, GS-FR, GS-CX, GS-BT, GS-BI, and GS-NLS, generate rapid and targeted structural modifications on selected parts of a solution. At the second level, the GS-IRS component performs an independent evolutionary search on a selected route or route window. Taken together, these two levels define a multi-layered memetic intensification architecture that differs from schemes based on a single local search mechanism.

3.4. Six Intensification Operators

This subsection introduces the six intensification operators of the GS framework using formal mathematical notation. Each operator is defined as a mapping $\mathcal{O} : \mathcal{F} \to \mathcal{F}$. The resulting solution is accepted if it is preferred to the current solution under the active fitness function. After each operator application, feasibility is restored through the repair function $\mathcal{R}$.

3.4.1. GS-SR: Stochastic Relocation

GS-SR is a fast and low-cost relocation operator defined at the single-node level. First, a node $v \in g$ with a potentially high contribution to rationality deterioration is selected through a mixed stochastic selection mechanism:

$$v \sim \pi_\sigma(g) \tag{20}$$

where π_σ denotes a probability distribution that performs random selection among different reference strategies. The selected node v is then removed from its current position, and only two reinsertion options are considered: immediately to the left or immediately to the right of its nearest neighboring node. The two candidate solutions are defined as

$$g'_{left} = \text{Insert}\Big(g \backslash \{v\}, pos_{left}(u^*)\Big),\ g'_{right} = \text{Insert}\Big(g \backslash \{v\}, pos_{right}(u^*)\Big) \tag{21}$$

where $u^* = \text{argmin}_{u \neq v} d_{v,u}$. Here, $d_{v,u}$ denotes the distance between nodes v and u, and $pos_{left}(\cdot)$ and $pos_{right}(\cdot)$ denote the insertion positions immediately to the left and right of the selected reference node, respectively. Among the two candidates, the better one is selected and accepted only if it improves upon the current solution:

$$g' = \begin{cases} \arg \min\limits_{g'' \in \{g'_{left}, g'_{right}\}} J(g'', g_c), & \text{if improvement is obtained} \\ g, & \text{otherwise} \end{cases} \tag{22}$$

3.4.2. GS-FR: Exhaustive Relocation

GS-FR is a more comprehensive variant of GS-SR. The node selection step is identical, but the selected node v is tested at all feasible positions in the chromosome in order to determine the best insertion point:

$$g' = \arg \min_{q \in \{1,\ldots,N\}} J(\text{Insert}(g \backslash \{v\}, q), g_c) \tag{23}$$

where q denotes a candidate insertion position in the chromosome and N is the total number of nodes in the cluster. Although $\mathcal{O}(N)$ more expensive than GS-SR, GS-FR can capture improvements missed by limited candidate evaluations.

3.4.3. GS-CX: Cross Exchange

GS-CX is an inter-route operator based on segment exchange between two routes. For two routes H_a and H_b, two node pairs, $(u_1, v_1) \in H_a \times H_b$ and $(u_2, v_2) \in H_a \times H_b$, are identified on the basis of geometric proximity relations. The cross exchange then swaps the corresponding segments of the two routes:

$$g' = \text{Swap}(g; [u_1, \ldots, u_2] \in H_a, [v_1, \ldots, v_2] \in H_b) \tag{24}$$

In Equation (24), $[u_1, \ldots, u_2]$ denotes the segment selected from route H_a, and $[v_1, \ldots, v_2]$ denotes the segment selected from route H_b. The operator has two modes of operation: the first determines the exchange pair on the basis of within-route proximity relations, whereas the second determines the segments with respect to a fixed reference node. In the reference method, namely the AkAyM and AkAyM2 operators, segment boundaries are selected entirely at random. GS-CX, by contrast, introduces an exchange mechanism guided by geometric reasoning.

3.4.4. GS-BT: Block Transfer

GS-BT is a one-way structural intervention in which a route block is removed from a source route and inserted into a target route. First, the internal edge costs on the source route H_s are evaluated, and two high-cost boundaries $[i_1, i_2]$ are identified. The block $B = g[i_1 : i_2]$ defined by this interval is then extracted. Subsequently, insertion positions are evaluated over the candidate target routes H_t, with $t \neq s$:

$$q^* = \arg \min_{t \neq s, q \in \text{positions}(H_t)} J(\text{Insert}(g \backslash B, q, B), g_c) \tag{25}$$

Here, B denotes the extracted block, q denotes a candidate insertion position on the target route, and positions (H_t) denotes the set of feasible insertion positions on route H_t. To reduce computational cost, an approximate evaluation is first performed, followed by a full fitness calculation only for a limited number of promising candidate positions.

3.4.5. GS-BI: Block Interchange

GS-BI is a high-impact operator in which the entire inner blocks of two routes are exchanged reciprocally. For a randomly selected source route H_s, inner-block exchanges with every other route H_t are evaluated sequentially:

$$g' = \underset{t \neq s}{\operatorname{argmin}} J(\text{InnerBlockSwap}(g; H_s, H_t), g_c) \tag{26}$$

In Equation (26), InnerBlockSwap$(\cdot)$ denotes the operation that exchanges the full inner blocks of routes H_s and H_t, while preserving their corresponding boundary nodes. Only improving swaps are accepted; otherwise, the solution is preserved. GS-BI accesses solution regions unreachable by smaller-scale operators.

3.4.6. GS-NLS: Neighborhood-Based Local Search

GS-NLS is a more computationally expensive but also more comprehensive operator than the preceding operators, as it is based on a systematic neighborhood scan over all inner nodes of a route. For a randomly selected route H_j, each inner node $v \in H_j \backslash \{S_j, E_j\}$ is temporarily removed. The κ nearest neighbors of v, denoted by $\mathcal{N}_\kappa(v)$, are then identified according to Euclidean distance, and reinsertion is attempted at both the left and right positions of each neighbor:

$$g' = \arg \min_{v \in H_j, u \in \mathcal{N}_\kappa(v), dir \in \{\text{left, right}\}} J(\text{Insert}(g \backslash \{v\}, \text{pos}(u, \text{ dir})), g_c) \tag{27}$$

In Equation (27), $\mathcal{N}_\kappa(v)$ denotes the set of the κ nearest neighbors of node v, dir denotes the insertion direction, and $\text{pos}(u, \text{ dir})$ denotes the insertion position immediately to the left or right of node u. After all node-neighbor-direction combinations have been evaluated, the single best move is applied in one step. This structure aims not at the cumulative effect of multiple small moves, but at identifying and applying one high-impact rearrangement.

3.5. GS-IRS: In-Route Evolutionary Search

GS-IRS is the component that most clearly expresses the memetic character of the GS framework. Whereas the other six operators rely on rule-based local moves, GS-IRS performs an independent evolutionary search on a selected route or route segment. It should therefore be regarded not merely as a local search step, but as a second evolutionary optimization layer defined on a selected subproblem.

Once a target route H_j has been selected, GS-IRS stochastically operates in one of two modes. In the full-route mode, the entire inner structure of H_j is re-optimized. In the window mode, the search is restricted to a randomly selected window $[j_a, j_b]$ within H_j. The resulting subproblem is defined for the corresponding segment with the following total path objective:

$$\min_{g_{sub} \in \mathcal{F}_{sub}} pt_{local}(g_{sub}) = \sum_{i=1}^{|g_{sub}|-1} \text{DSM}_{g_{sub}[i], g_{sub}[i+1]} \tag{28}$$

where F_{sub} denotes the feasible solution space of the selected segment, g_{sub} denotes a candidate ordering within that local segment, $|g_{sub}|$ denotes the number of nodes in the segment, and pt_{local} denotes the total path length over the segment. The inner evolutionary process uses selection, crossover, mutation, and elitism. After convergence, the best local ordering is written back to the main chromosome. Algorithm 2 summarizes this mechanism.

Algorithm 2. GS-IRS procedure (In-route evolutionary search)

Input: incumbent solution g, selected target route H_j, active distance matrix DSM_active, raw distance matrix DSM_raw, same-location matrix S, local search scale k_loc, active phase indicator Phase
Output: best solution g
1. Determine the working mode
2. If rand() < 0.20, then
3. Mode ← FULL-ROUTE
4. I_work ← all node positions of H_j
5. Else
6. Mode ← WINDOW
7. Randomly select a feasible inner segment of H_j
8. Expand the selected segment by including its two boundary nodes
9. I_work ← indices of the resulting local window
10. End If
11. Extract the local route segment H_work from H_j using I_work
12. Define the local constraint structure F_work with fixed start and end nodes
13. Construct local matrices from H_work:
14. DSM_work ← submatrix of DSM_active
15. DSM_raw,work ← submatrix of DSM_raw
16. EM_work ← DSM_raw,work
17. Set local GA parameters as a function of the segment size and k_loc
18. Generate an initial local population P_loc
19. Repair each local individual to preserve local feasibility
20. Insert the original ordering of H_work into P_loc
21. Set h* as the best local individual in P_loc
22. For g_loc = 1 to n_gˆloc do
23. Evaluate all local individuals in P_loc using the local objective
24. Update h* if a better local solution is found
25. Select parent individuals by roulette-wheel selection
26. Apply prefix-preserving crossover to generate local offspring
27. Apply local mutation operators
28. Repair all offspring to preserve local feasibility
29. Insert h* into the new local population by elitism
30. End For
31. Replace the corresponding positions of H_j in g with h*
32. Apply the global repair operator to g
33. Return g

An important point is that, unlike classical nested GAs [11,25–28], GS-IRS is not a mandatory computational layer applied to every candidate solution. It is activated only after the warm-up threshold $g_c > \theta_{mg} \cdot n_g$ has been exceeded and only for selected individuals. This selective application provides controlled search depth without distributing the computational cost of the memetic architecture across the entire population.

3.6. Operator Selection and Warm-Up Threshold

The activation condition of the GS framework and the mechanism for selecting the target individual are governed by two parameters. First, the warm-up threshold is defined over the generation counter as

$$g_c > \theta_{mg} \cdot n_g \tag{29}$$

where g_c denotes the current generation counter, n_g denotes the total number of generations, and θ_{mg} denotes the warm-up ratio for activating the GS framework. In this study, $\theta_{mg} = 0.15$, meaning that the first 15% of the total generations are reserved for global exploration and that the GS framework becomes active only after this threshold has been exceeded.

Second, the individual to which intensification is applied is determined by a stochastic rule. To balance the systematic refinement of the global best individual g^* with the preservation of population diversity, the target individual is selected as follows:

$$g_{target} = \begin{cases} g^*, & r < \text{prob}_{\text{best}} \\ G_{(p,:)}, p \sim \mathcal{U}\{1, \ldots, ps\}, & r \geq \text{prob}_{\text{best}} \end{cases} \quad (30)$$

where $r \sim \mathcal{U}(0,1)$ is a uniformly distributed random variable, $\text{prob}_{\text{best}}$ denotes the probability of selecting the current best individual, ps denotes the population size, and $G_{(p,:)}$ denotes the p-th individual in the current population. In this study, $\text{prob}_{\text{best}} = 0.50$. Therefore, intensification is applied with approximately equal probability either to the global best individual or to a randomly selected individual. This balance provides a compromise between deepening the elite solution and maintaining exploratory diversity across the population.

Within the GS framework, the six intensification operators together with the GS-IRS component form a complementary operator family in terms of move type, operational scope, and computational cost. Table 1 presents a comparative summary of these operators. Since each operator acts at a different intervention scale, the GS framework establishes the exploration–exploitation balance not only at the level of the main GA, but also through the internal diversity of the intensification operators.

Table 1. Comparative summary of the operators in the Genetic Scissors framework.

Operator	Scale	Move Type	Operational Scope	Computational Cost
GS-SR	Node	Stochastic relocation	Intra-route	Low
GS-FR	Node	Exhaustive relocation	Intra-route	Moderate
GS-CX	Segment	Reciprocal exchange	Inter-route	Moderate
GS-BT	Block	Unidirectional transfer	Inter-route	Moderate–High
GS-BI	Entire inner block	Reciprocal exchange	Inter-route	High
GS-NLS	Route	Systematic neighborhood search	Intra-route	High
GS-IRS	Route/window	Inner evolutionary search	Selective route-level	Highest

This operator diversity yields a multi-level intensification architecture, distinguishing the proposed framework from memetic structures that rely on a single type of local search mechanism.

4. Regression-Based Adaptive Parameterization

The performance of the proposed algorithm is sensitive to upper-level control parameters such as the population size (ps), the number of generations (n_g), the phase-switch ratio (θ_r), and the application probability of the intensification operators (p_{mpr}). Defining these parameters through a single fixed set of values for all clusters leads to imbalanced behavior across problem scales. In small clusters, large parameter values introduce unnecessary computational cost, whereas in large clusters, small parameter values restrict the search capacity. For this reason, the parameters are determined through a two-stage procedure that is responsive to problem size. First, suitable parameter values are identified through

random search on representative clusters, after which the resulting patterns are generalized through regression analysis.

4.1. Random Search on Representative Clusters

Direct parameter search over all 213 normal clusters is not computationally practical. Therefore, a limited number of clusters were selected to represent the problem size spectrum, using size groups defined according to the number of nodes. The size grouping is defined as

$$\mathcal{B}(N) = \begin{cases} \text{Very small}, & N \leq 40 \\ \textit{Small}, & 40 < N \leq 70 \\ \text{Medium}, & 70 < N \leq 200 \\ \text{Large}, & N > 200 \end{cases} \tag{31}$$

Here, $\mathcal{B}(N)$ denotes the size class associated with a cluster containing N nodes. At least one cluster was selected from each size group. When multiple candidate clusters existed within the same group, the cluster closest to the group mean was preferred. In total, six representative clusters were identified, spanning the full range from very small to large instances.

Random search was adopted as the parameter exploration method. As shown by Bergstra and Bengio [49], when not all parameters are equally influential on solution quality, random search can be more effective than grid search under the same evaluation budget. The search space was defined through the following independent uniform distributions:

$$\begin{gathered} \Theta = \{ps\} \times \{n_g\} \times \{\theta_r\} \times \{p_{mpr}\} \\ ps \sim \mathcal{U}_{\mathbb{Z}}\{50\ 250\},\ n_g \sim \mathcal{U}_{\mathbb{Z}}\{500, 3000\},\ \theta_r \sim \mathcal{U}(0.20, 0.90),\ p_{mpr} \sim \mathcal{U}(0.10, 0.50) \end{gathered} \tag{32}$$

In Equation (32), Θ denotes the parameter search space, $\mathcal{U}_{\mathbb{Z}}$ denotes a discrete uniform distribution over integers, and $\mathcal{U}$ denotes a continuous uniform distribution. For each representative cluster, $T = 15$ independent parameter trials were performed, and each trial was evaluated over $R = 5$ independent runs. This resulted in a total of $6 \times 15 \times 5 = 450$ algorithm runs.

The performance of a parameter vector θ on cluster c was defined as the mean rationality value over the repeated runs:

$$f(\theta, c) = \frac{1}{R} \sum_{\updownarrow=1}^{R} rat_{\updownarrow}(\theta, c) \tag{33}$$

where $f(\theta, c)$ denotes the average performance of parameter vector θ on cluster c, and $rat_{\updownarrow}(\theta, c)$ denotes the rationality value obtained in the $\updownarrow$-th run under parameter vector θ. For each cluster, the best parameter combination was selected according to

$$\theta_c^* = \arg \min_{\theta \in \{\theta_1, \theta_2, \ldots, \theta_T\}} f(\theta, c) \tag{34}$$

where θ_c^* denotes the best parameter vector identified for cluster c, and $\{\theta_1, \theta_2, \ldots, \theta_T\}$ denotes the set of tested parameter vectors. This structure ensures parameter selection through a stable performance criterion based on repeated-run averages rather than on isolated noisy outcomes.

4.2. Generalization Through Regression Analysis

For each representative cluster, the relationship between the resulting optimal parameter vector θ_c^* and the corresponding number of nodes N was examined. The aim of this stage is not merely to tune the parameters for a few representative instances, but to derive analytical expressions that can be generalized across all cluster sizes. In this way, once the number of nodes is known for a new cluster, the parameter values can be computed directly without performing a separate parameter search.

Linear regression models were constructed for $ps^*(N)$ and $n_g^*(N)$:

$$ps^*(N) = \beta_0 + \beta_1 N \tag{35}$$

$$n_g^*(N) = \gamma_0 + \gamma_1 N \tag{36}$$

where $ps^*(N)$ denotes the population size predicted for a cluster with N nodes, $n_g^*(N)$ denotes the corresponding predicted number of generations, and β_0, β_1, γ_0, γ_1 are regression coefficients. The coefficients were estimated by the ordinary least squares method:

$$\beta = \left(X^\top X\right)^{-1} X^\top y \tag{37}$$

where X denotes the design matrix, y denotes the response vector, and β denotes the vector of regression coefficients. Although Equation (37) is written here in generic form, the same estimation principle was used for both regression models in Equations (35) and (36).

In the initial regression based on six representative clusters, one cluster, namely Cluster 184 with $N = 47$, was found to deviate markedly from the others with respect to the phase-switch ratio θ_r. In this cluster, the best result was obtained at θ_r = 0.81, whereas for the remaining clusters this parameter remained within the range 0.38 to 0.57. This observation was not treated as a selective data-cleaning decision, but as a methodological adjustment intended to preserve the explanatory power of the regression model. Accordingly, the relevant cluster was excluded only at the parameter generalization stage. The regression results obtained on the remaining five clusters are shown in Figure 1, while the estimated coefficients and explanatory statistics are summarized in Table 2.

Table 2. Final parameter set: regression-based and fixed values.

Parameter	Value/Equation	Selection Rule	Rationale
ps	$73.54 + 0.4391 \cdot N$	Regression ($R^2 = 0.589$)	Moderate linear trend
n_g	$972.48 + 4.3151 \cdot N$	Regression ($R^2 = 0.817$)	Strong linear trend
θ_r	0.45	Fixed ($R^2 = 0.116$)	Weak explanatory power
p_{mpr}	0.30	Fixed ($\sigma = 0.018$)	Low variance
p_m, p_{m2}	0.03	Design choice	Mutation diversity
p_{mf}	0.07	Design choice	Structure-preserving mutation
$\text{prob}_{\text{best}}$	0.50	Design choice	Target selection balance
θ_{mg}	0.15	Design choice	GS warm-up threshold
n_g^{in}	≤ 150	Empirical test	Quality–time balance

The regression results reveal two distinct parameter behavior profiles. For ps and n_g, reliable linear relationships with the number of nodes were obtained, with $R^2 = 0.589$ and $R^2 = 0.817$, respectively. These two parameters were therefore adapted to cluster size through regression-based expressions. In contrast, the low explanatory power obtained for θ_r, namely $R^2 = 0.116$, indicated that this parameter cannot be represented reliably by a linear model in terms of N. Similarly, p_{mpr} exhibited a narrow spread across clusters, with

a standard deviation of $\sigma = 0.018$. These two parameters were therefore assigned fixed values supported by the observed data.

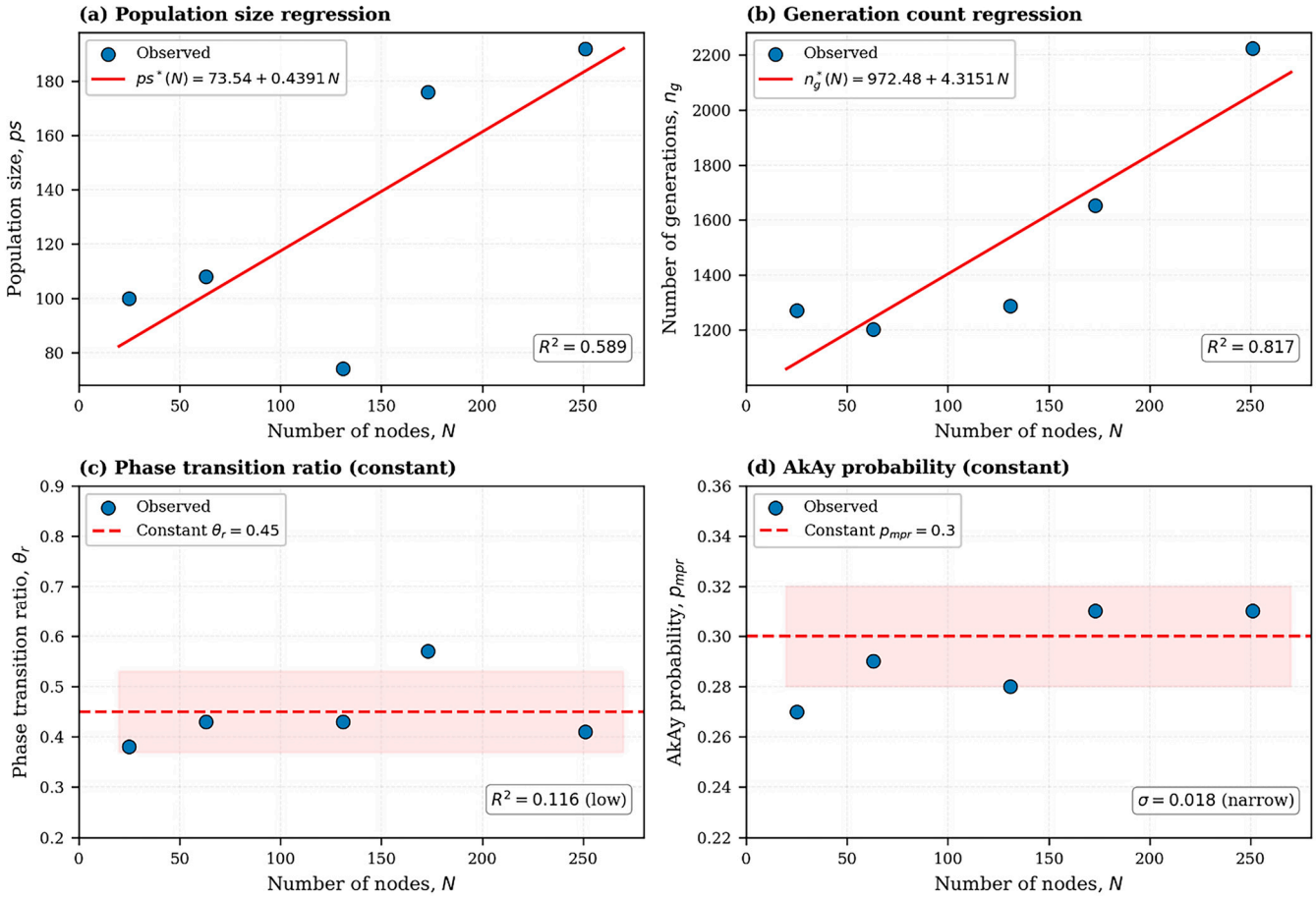


Figure 1. Relationship between the best parameter values obtained on the representative clusters and the number of nodes N: (**a**) population size ps with regression line ($R^2 = 0.589$); (**b**) number of generations n_g with regression line ($R^2 = 0.817$); (**c**) phase-switch ratio θ_r with fixed reference value; and (**d**) intensification application probability p_{mpr} with fixed reference value. The asterisk (*) denotes the optimal parameter values predicted by the regression models.

Finally, the upper bound of the inner generation count in the GS-IRS component, denoted by n_g^{in}, was evaluated through a separate comparative experiment. Four usage modes were compared, namely no GS-IRS, unrestricted inner n_g, execution once every five generations, and $n_g^{\text{in}} \leq 150$. Among these alternatives, the configuration $n_g^{\text{in}} \leq 150$ provided the most balanced profile in terms of both rationality ($rat = 2.646$) and computational time (t = 173 s). This choice was therefore treated not as a heuristic preference, but as an experimentally supported design decision. To validate the cost-effectiveness of this bounded design, the empirical runtime of GS-IRS was evaluated on a representative cluster across three configurations. When GS-IRS is omitted, the algorithm completes in 129 s but yields a relatively poor mean rationality of 2.9898, indicating limited geometric refinement. Releasing the inner-generation bound increases the runtime to 411 s without producing proportional quality gains. The bounded configuration adopted in this study ($n_g^{in} \leq 150$) increases the runtime to only 173 s, a modest overhead of approximately 44 s, while reducing the mean rationality to 2.6460. This 11.5% improvement in geometric quality at a 34% runtime increase confirms that the bounded GS-IRS design provides a favorable cost-effectiveness profile, supporting its inclusion as a selective rather than exhaustive intensification mechanism.

The practical outcome of the adaptive parameterization scheme was an adjustable structure ranging approximately from $ps \approx 86, n_g \approx 980$ for $N = 25$ to $ps \approx 184$ and $n_g \approx 2056$ for $N = 251$. This structure avoids unnecessary computational burden in small problems while preventing insufficient search capacity in large ones. As a result, the parameter configuration ceases to be a single fixed setting and becomes a mechanism that adapts automatically to problem size.

5. Experimental Design

This section outlines the experimental design used to evaluate the Genetic Scissors framework, covering the dataset and baseline reference, the cluster filtering procedure and evaluation samples, the comparison methods and execution settings, and the statistical testing framework.

5.1. Dataset and Baseline Reference

The experimental evaluation was conducted on the Istanbul public transit dataset described earlier. For each cluster, the baseline corresponds to the actual stop ordering contained in the raw data, evaluated by directly applying Equations (8) and (9). This yields two baseline profiles for the existing real route structure of each cluster before the optimization algorithm is executed: the total path length, denoted by $\text{baseline}_{pt}(c)$, and the rationality, denoted by $\text{baseline}_{rat}(c)$. This approach allows the contribution of the optimization algorithm to be assessed not against an abstract ideal solution, but against the routes currently used in the real system.

To reduce the sensitivity of stochastic evolutionary algorithms to random initialization, each cluster and each algorithmic variant were evaluated over R = 5 independent runs. The mean rationality value for a cluster is defined as

$$\overline{rat}(c) = \frac{1}{R} \sum_{\updownarrow=1}^{R} rat_{\updownarrow}(c) \tag{38}$$

where $\overline{rat}(c)$ denotes the average rationality over repeated runs for cluster c, and $rat_{\updownarrow}(c)$ denotes the rationality value obtained in the $\updownarrow$-th run. The percentage improvement with respect to the baseline is then defined as

$$\Delta rat(c) = \frac{\text{baseline}_{rat}(c) - \overline{rat}(c)}{\text{baseline}_{rat}(c)} \times 100 \tag{39}$$

Positive values of $\Delta rat(c)$ indicate improvement relative to the existing route structure, whereas negative values indicate deterioration. The dispersion across repeated runs is evaluated through the standard deviation

$$\sigma_{rat}(c) = \sqrt{\frac{1}{R-1} \sum_{\updownarrow=1}^{R} \left(rat_{\updownarrow}(c) - \overline{rat}(c) \right)^2} \tag{40}$$

where $\sigma_{rat}(c)$ denotes the standard deviation of rationality over repeated runs for cluster c. Low values of $\sigma_{rat}(c)$ indicate that the method produces consistent solutions despite different random initializations.

The choice of the genetic algorithm as the underlying evolutionary backbone of Genetic Scissors follows the systematic algorithm-selection study reported in [7], where eight optimization methods, including exact algorithms (Brute Force Search, Held-Karp dynamic programming, Branch and Bound), constructive heuristics (Nearest Neighbor, Christofides), and three metaheuristics (Artificial Bee Colony, Genetic Algorithm, Tree-Seed

Algorithm), were comparatively evaluated. In that study, exact methods proved infeasible for problem sizes representative of real transit networks, and constructive heuristics did not reach competitive solution quality. Among the metaheuristics, GA produced the best mean performance with the lowest variability, motivating its adoption as the validated baseline for the present extended framework.

5.2. Cluster Filtering and Evaluation Samples

When the baseline rationality values were examined for all 250 clusters in the dataset, a small subset was found to produce unusually high values. This behavior has two main causes. First, some routes preserve outbound and inbound movements on the same geometric structure, which creates a structural effect that can inflate rationality dramatically for stop pairs passing through the turning region. Second, data-quality issues, such as errors in stop coordinates or incorrectly defined route compositions, prevent the formation of a reliable reference in some clusters. Including such clusters directly in the optimization assessment would therefore produce misleading results. The set of normal clusters retained for analysis is defined as

$$C_{\text{normal}} = \left\{ c \in C : \text{baseline}_{rat}(c) < \theta_f \right\} \tag{41}$$

where $\theta_f = 6$ is the filtering threshold. After this filtering step, 213 clusters were found suitable for analysis. The excluded clusters were classified not only according to the magnitude of the values they produced, but also according to the structural source of those values. The exclusion decision should therefore be interpreted not as data manipulation, but as a methodological adjustment introduced to preserve comparability across evaluations.

The experimental assessment was conducted on two different samples. The general evaluation sample consists of all 213 normal clusters and is used to measure the absolute performance of each method against the baseline under its own problem definition. The fair-comparison sample is a 109-cluster subset obtained by excluding, from the 213 normal clusters, those clusters that contain stops with identical coordinates within the same route. This subset represents a common problem domain in which the extended constraint structure, in particular the same-location constraint $\mathcal{F}_3$, does not provide a one-sided advantage to the proposed method. As a result, the direct comparison with the reference method in Akgol, Demir, Aydogdu and Murat [7] is carried out on a fairer basis.

5.3. Comparison Methods and Ablation Analysis

The evaluation was conducted on three main configurations: the baseline, representing the existing route arrangements currently used in the real system; the reference method, corresponding to the two-phase GA equipped with the AkAy operators proposed by Akgol, Demir, Aydogdu and Murat [7], namely AkAy, AkAyM, and AkAyM2; and Genetic Scissors (GS), corresponding to the proposed method incorporating the six intensification operators together with the GS-IRS component.

To distinguish the individual contributions of the method components, ablation experiments were designed on nine representative clusters. In these experiments, six different algorithmic configurations were evaluated: (a) a structure based only on total path length, (b) a two-phase fitness structure without intensification, (c) the GS-IRS component alone, (d) the AkAy operators as the reference method, (e) the combination of AkAy and GS-IRS, and (f) the full GS framework, including the six operators and GS-IRS. These configurations make it possible to determine whether the performance of the proposed method arises from a single dominant operator or from the interaction of multiple components.

All experiments were conducted in the same hardware environment using the previously determined adaptive parameters. In addition to solution quality, the mean runtime

was recorded for each configuration, allowing the methods to be evaluated jointly under a quality-cost trade-off.

5.4. Statistical Tests

Since the cluster-level rationality values may not satisfy the normality assumption and may contain outliers, the nonparametric Wilcoxon signed-rank test was adopted for paired comparisons. Because this test is based on the signed ranks of paired observations, it does not require distributional assumptions and provides robust results in the presence of outliers.

The test was applied to three comparison types: the difference between the baseline and the reference method; the difference between the baseline and Genetic Scissors; and the difference between the reference method and Genetic Scissors, evaluated on the 109-cluster fair-comparison sample. For each test, the test statistic W, the p-value, and the effect size r were reported. The effect size was calculated as

$$r = \frac{|Z|}{\sqrt{n}} \tag{42}$$

where Z denotes the standardized normal transformation of the Wilcoxon test statistic, and n denotes the number of paired comparison samples. According to Cohen's interpretation, $|r| < 0.1$ indicates a negligible effect, $0.1 \leq |r| < 0.3$ a small effect, $0.3 \leq |r| < 0.5$ a medium effect, and $|r| \geq 0.5$ a large effect [50]. The significance threshold was set to $\alpha = 0.05$. However, most of the obtained p-values remained well below this threshold, as shown in the subsequent results.

This statistical framework ensured that visual and descriptive comparisons were supported by a formal test of quantitative significance and confirmed that the differences between the methods were not merely incidental, but systematically reliable.

6. Results and Discussion

This section presents the findings obtained from the experimental evaluation of the Genetic Scissors framework. It begins with the ablation analysis conducted on nine representative clusters to identify the contributions of the individual components. It then presents the comparative validation supported by the Wilcoxon test on 109 clusters. The remaining subsections examine the behavior of the method with respect to problem scale, the visual improvements in route geometry, and the methodological interpretation of the findings.

6.1. Ablation Analysis: Component Contributions

To identify the components responsible for the performance of the Genetic Scissors framework, the six algorithmic configurations introduced earlier were evaluated on nine representative clusters. The results are reported in Table 3.

The results yield four main findings. First, the ***pt*** configuration, which is based solely on total path length, produces extremely high rationality values, with a mean of 25.025, which is nearly an order of magnitude larger than the mean baseline value. This finding shows that pure distance minimization is fundamentally insufficient for preserving spatial directness in multi-route public transit networks. Second, the two-phase fitness structure, denoted $\boldsymbol{pt} \rightarrow \boldsymbol{rat}$, improves upon ***pt*** (25.025 $\rightarrow$ 20.672), but still fails to achieve satisfactory solution quality when used in isolation. This result indicates that the two-phase objective structure is necessary but not sufficient, and that it must be supported by problem-specific intensification mechanisms.

Table 3. Ablation results on the nine representative clusters (mean rationality values).

Cluster	Baseline	*pt*	*pt*→*rat*	GS-IRS	AkAy	AkAy + GS-IRS	Genetic Scissors
C4	2.551	22.550	9.687	4.612	2.534	2.418	2.438
C15	2.905	30.421	19.435	7.257	2.807	2.649	2.645
C18	2.643	78.346	97.840	36.550	3.267	2.994	2.745
C49	3.932	13.843	12.294	4.636	2.328	2.753	2.725
C62	2.390	39.828	8.287	4.971	3.366	2.658	2.388
C78	2.434	14.203	9.825	5.499	2.377	2.383	2.363
C117	2.480	5.990	5.733	4.214	2.709	2.357	2.394
C213	2.675	12.665	12.502	5.444	3.003	2.773	2.643
C250	2.835	7.379	10.443	3.971	2.076	2.308	2.360
Mean	2.761	25.025	20.672	8.573	2.718	2.588	2.522

Third, the configuration containing only the GS-IRS component, with a mean value of 8.573, provides a substantial improvement over ***pt*** and ***pt*** → ***rat***, yet remains inferior to both the baseline and the AkAy-based structures. This confirms that GS-IRS is not positioned as an independent solution mechanism, but rather as a complementary component that becomes effective within an appropriate intensification framework. Indeed, when GS-IRS is combined with the AkAy operators, the mean performance improves from 2.718 to 2.588.

In addition to these four comparative findings, the numerical performance profile of the Genetic Scissors method on the nine representative clusters is presented in Table 4. The table reports, for each cluster, the mean and minimum rationality values relative to the baseline, the standard deviation across repeated runs, and the corresponding percentage improvement.

Table 4. Performance of GS relative to the baseline on the nine representative clusters (R = 5 independent runs).

Cluster	Baseline	GS Mean	GS Std	GS Min	GS Improvement (%)
C4	2.551	2.438	0.014	2.420	4.43
C15	2.905	2.645	0.067	2.533	8.95
C18	2.643	2.745	0.108	2.634	−3.86
C49	3.932	2.725	0.053	2.669	30.70
C62	2.390	2.388	0.000	2.388	0.08
C78	2.434	2.363	0.045	2.320	2.92
C117	2.480	2.394	0.015	2.380	3.47
C213	2.675	2.643	0.124	2.526	1.20
C250	2.835	2.360	0.003	2.356	16.75
Mean	2.761	2.522	—	2.470	8.66

Table 4 shows that Genetic Scissors improves upon the baseline in eight of the nine representative clusters and yields an average rationality improvement of 8.66%. In particular, the standard deviation being equal to zero in Cluster 62 ($\sigma = 0.000$) indicates that all five independent runs converged to the same solution, demonstrating that the method exhibits highly stable behavior for this type of cluster.

The GS Min column reported in Table 4 represents the best rationality values reached by the algorithm across the five independent runs, and it is notable that these values fall substantially below the corresponding mean performance levels. From the perspective of GS Min, improvement over the baseline is obtained in all nine clusters, and the mean

improvement reaches 10.54%. This value is approximately two percentage points higher than the 8.66% improvement computed on the basis of GS Mean. The difference reflects the gap between the performance ceiling achieved in the best runs and the average behavior of the method, and suggests that higher-quality solutions may be reachable if the number of runs is increased or the diversity of initial conditions is enriched. Notably, even in Cluster 18, where GS Mean remains above the baseline, the GS Min value (2.634) falls below the baseline (2.643), indicating that at least one of the five runs improved upon the existing configuration. The C18 case is therefore better interpreted as a stability boundary than a failure (see Section 6.4). The gap between GS Min and GS Mean becomes particularly pronounced in some clusters. In Cluster 49, the difference between GS Min (2.669) and GS Mean (2.725) corresponds to 2.06%; in Cluster 15, the difference between GS Min (2.533) and GS Mean (2.645) is 4.43%; and in Cluster 213, the difference between GS Min (2.526) and GS Mean (2.643) reaches 4.63%. These results indicate that the solution spaces of these clusters are broader and that the algorithm may converge to different local attraction regions across repeated runs. This observation also provides a concrete basis for the later discussion on the relationship between computational budget and search depth, and suggests that longer generation budgets and adaptive early-stopping mechanisms may improve the ability of the method to approach its performance ceiling.

Fourth, and most importantly, the proposed Genetic Scissors structure outperforms all other configurations on the nine representative clusters by producing the lowest mean rationality value (2.522). This mean is 8.66% lower than the baseline and 2.5% lower than the AkAy + GS-IRS combination. The superiority becomes even more pronounced in specific cluster instances. In Cluster 62, Genetic Scissors yields a rationality value of 2.388, whereas AkAy alone remains at 3.366. Even in a difficult case such as Cluster 18, Genetic Scissors (2.745) substantially outperforms all other advanced configurations, including AkAy (3.267) and AkAy + GS-IRS (2.994). These findings show that Genetic Scissors is not merely a variation in the existing reference approach, but rather an integrated intensification architecture arising from the interaction of components operating at the node, segment, block, and in-route evolutionary search levels.

6.2. Comparative Validation on 109 Clusters

To examine whether the findings of the ablation analysis remain valid at a broader scale, the reference method and Genetic Scissors were evaluated on the 109-cluster fair-comparison sample. The results were examined both in terms of their performance relative to the fv over one another.

Performance relative to the baseline. The reference method (AkAy) improved upon the baseline in 59 of the 109 clusters, while it produced deterioration in the remaining 50 clusters. Its mean improvement remained at approximately 0.23%, and the Wilcoxon signed-rank test did not indicate statistical significance ($p > 0.05$). In contrast, Genetic Scissors improved upon the baseline in 89 of the 109 clusters and deteriorated in only 20 clusters. The mean improvement was calculated as 2.41%, while the median improvement was 0.92%. The Wilcoxon test indicated that this improvement was highly significant, with $W = 856$, $p < 0.001$, and $r = 0.750$. According to Cohen's criteria, this effect size falls into the large-effect category, as reported in Table 5.

Direct comparison between methods. Genetic Scissors produced a better rationality value than the reference method in 84 of the 109 clusters. The Wilcoxon test again indicated that this difference was highly significant ($p < 0.001$, $r = 0.714$), corresponding to a large effect size. This result shows that the proposed framework provides systematic superiority not only within its own extended problem definition, but also on the common problem subset where it is compared fairly with the reference method.

Table 5. Wilcoxon signed-rank test results (109 clusters).

Comparison	Improved/Worsened	Mean Δrat (%)	W	p	r	Effect
Baseline vs. AkAy	59/50	0.23	—	>0.05	—	Not significant
Baseline vs. GS	89/20	2.41	856	<0.001	0.750	Large
AkAy vs. GS	84/25	—	—	<0.001	0.714	Large

The em dash (—) indicates that the corresponding value is not applicable.

The distributional reflection of the performance difference between the methods can be seen in the boxplot shown in Figure 2. While the median of the baseline distribution is around 2.69, the AkAy distribution remains centered around a similar median with a somewhat reduced spread. By contrast, the Genetic Scissors distribution is shifted downward more clearly in terms of both the median and the lower quartile values. This visual pattern provides the distributional basis for the large effect sizes reported in Table 5 ($r = 0.750$ and $r = 0.714$).

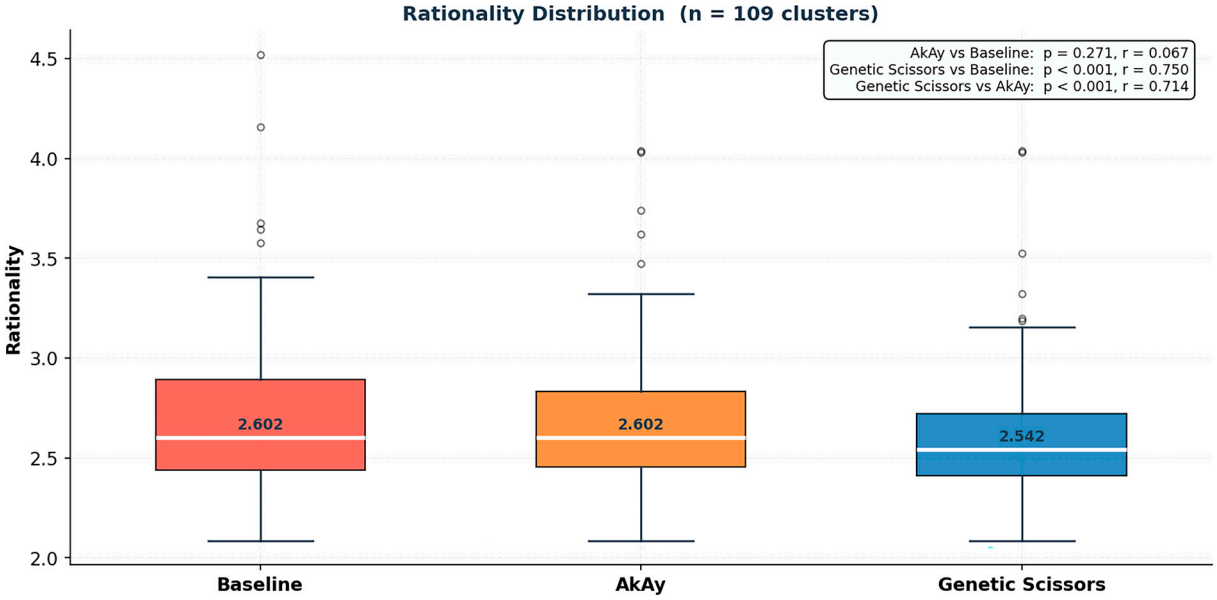


Figure 2. Boxplot comparison of rationality values for the baseline, the reference method (AkAy), and Genetic Scissors across 109 clusters.

Taken together, these findings show that the improvement effect of Genetic Scissors relative to the existing routes appears consistently in 81.7% of the clusters (89/109), while its superiority over the reference method is observed in 77.1% of the clusters (84/109). The fact that the observed p-values remain below 0.001 confirms that these differences are not incidental, but systematic.

Although the average rationality improvement of 2.41% might appear modest in purely numerical terms, its practical significance in a metropolitan transit network is substantial. First, this gain was achieved against a highly constrained, real-world baseline of operationally established routes that had already been refined through years of operational experience; improving such a baseline by 2.41% without violating positional integrity constraints is structurally non-trivial. Second, since rationality is a fundamental geometric ratio, a permanent improvement of this magnitude across 81.7% of the network clusters has the potential to contribute to non-trivial cumulative effects on operational metrics such as fuel consumption, fleet utilization, and aggregate passenger travel time, although a precise

quantification of these downstream impacts would require complementary operational modeling beyond the scope of the present geometric optimization study. Third, the large effect size (r = 0.750) confirms that this gain reflects a systematic structural enhancement rather than a marginal fluctuation.

6.3. Behavior with Respect to Problem Scale

It is important to determine whether the difference between the methods is consistent not only in terms of average performance but also across the full range of problem scales. For this purpose, performance was analyzed with respect to both the number of stops N and the number of routes k.

Effect of the number of stops. For the reference method, a significant negative correlation was observed between the number of stops and the improvement ratio, indicating that the performance of AkAy declines systematically as cluster size increases. By contrast, for Genetic Scissors, the correlation between N and Δrat was not significant ($p > 0.05$). This asymmetry shows that the proposed method exhibits a more balanced performance profile across small- and large-scale problems.

Effect of the number of routes. A similar difference is observed with respect to the number of routes. Whereas the reference method loses performance in multi-route structures ($k \geq 4$), Genetic Scissors is able to produce positive mean improvement across all route-count levels. This result suggests that the GS-CX, GS-BT, and GS-BI operators, which introduce inter-route structural interventions, play a decisive role in large and multi-route problems. In particular, the GS-BI operator enables reciprocal exchange of entire inner blocks, thereby providing access to regions of the solution space that cannot be reached by smaller-scale operators.

The relationship between the number of stops and the improvement ratio is shown visually in Figure 3. For the reference method, a clear downward trend is observed in the improvement ratio as cluster size increases, whereas for Genetic Scissors this trend disappears, and the improvement ratio exhibits a distribution that is largely independent of cluster size.

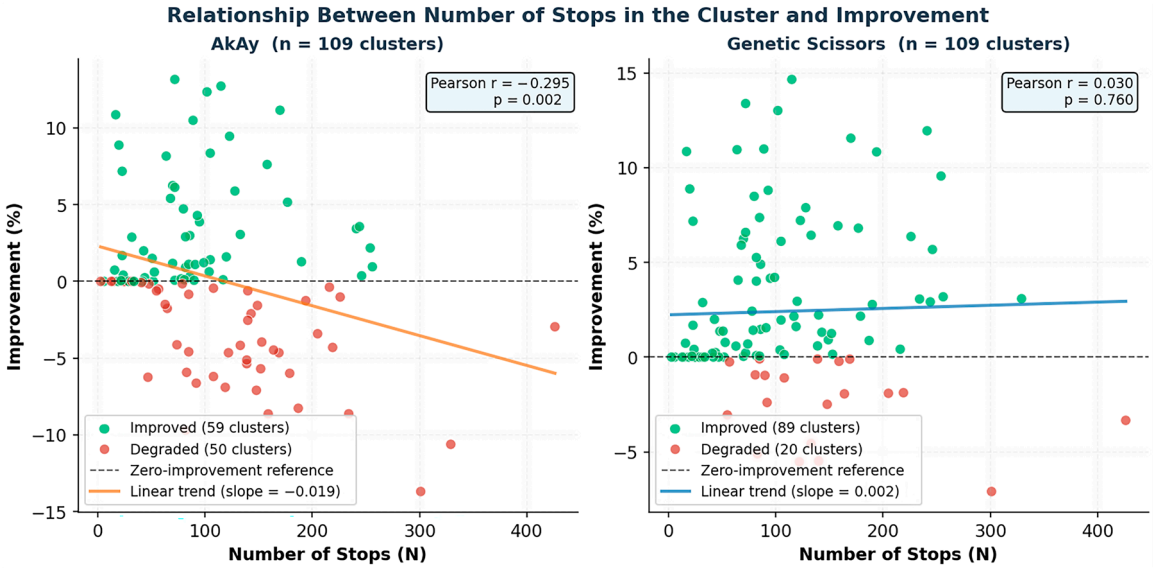


Figure 3. Relationship between the number of stops (N) and the rationality improvement relative to the baseline (Δrat): comparative distribution for the reference method and Genetic Scissors. Green and red points denote clusters with positive and negative improvement, respectively; the horizontal dashed line marks the zero-improvement reference; and the colored solid line shows the linear trend fitted to the data.

Taken together, these findings indicate that the superiority of the Genetic Scissors framework does not stem merely from containing a larger number of operators, but from the fact that these operators provide the structural intervention diversity required for large and multi-route problems. This behavior points to a structural robustness across problem scales, supporting the applicability of the proposed framework to real-world public transit networks.

The term scale-invariant here refers specifically to the structural geometric performance of the framework, that is, the absence of systematic rationality degradation as the number of stops or routes increases, rather than to formal asymptotic computational complexity. Empirical evidence on the computational cost of the framework's heaviest component (GS-IRS) is reported in Section 4.2.

6.4. Visual Improvements in Route Geometry

The numerical performance indicators demonstrate the rationality improvement achieved by the algorithm. However, the way in which this improvement is reflected in route geometry can be observed more concretely through visual comparison. For this purpose, five representative examples were selected to reflect the behavioral diversity of the method: Clusters 49 and 250, where the method exhibited its strongest performance; Clusters 117 and 213, where the improvement was moderate; and Cluster 18, where the method did not improve upon the baseline. In addition, a map-based comparison for Cluster 213 is included to illustrate more explicitly how route changes are manifested within the real urban context, as shown in Figures 4–8.

Figure 4, which presents Cluster 49, represents a case in which the method achieves one of its strongest performances. In this cluster, the baseline rationality value is as high as 3.932, whereas Genetic Scissors reduces it to 2.725, corresponding to an improvement of 30.70%. Inspection of the route geometries shows that the long deviations present in the baseline configuration are markedly shortened in the Genetic Scissors solution and that the routes acquire a more coherent spatial direction. The example of Cluster 250 shown in Figure 5 confirms that the same dynamic remains effective in a cluster of different scale, where the 16.75% improvement is obtained mainly through the correction of deviations in the outer route sections.

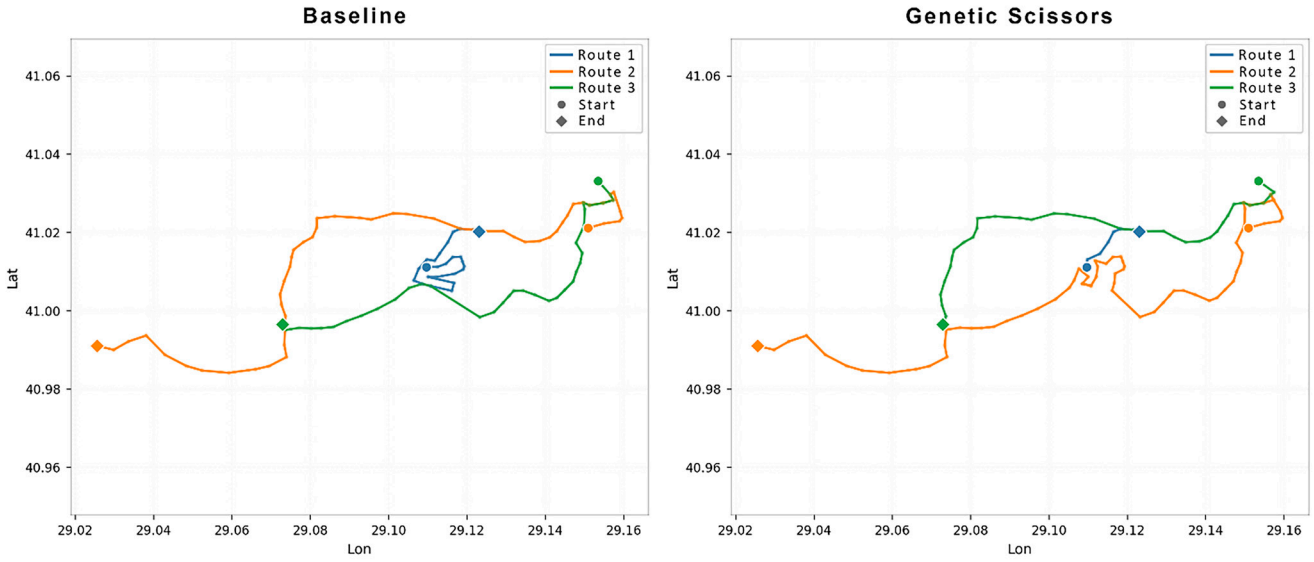


Figure 4. Route comparison for Cluster 49 (N = 113, k = 3) baseline (rat = 3.932) and Genetic Scissors output (rat = 2.725, +30.70%).

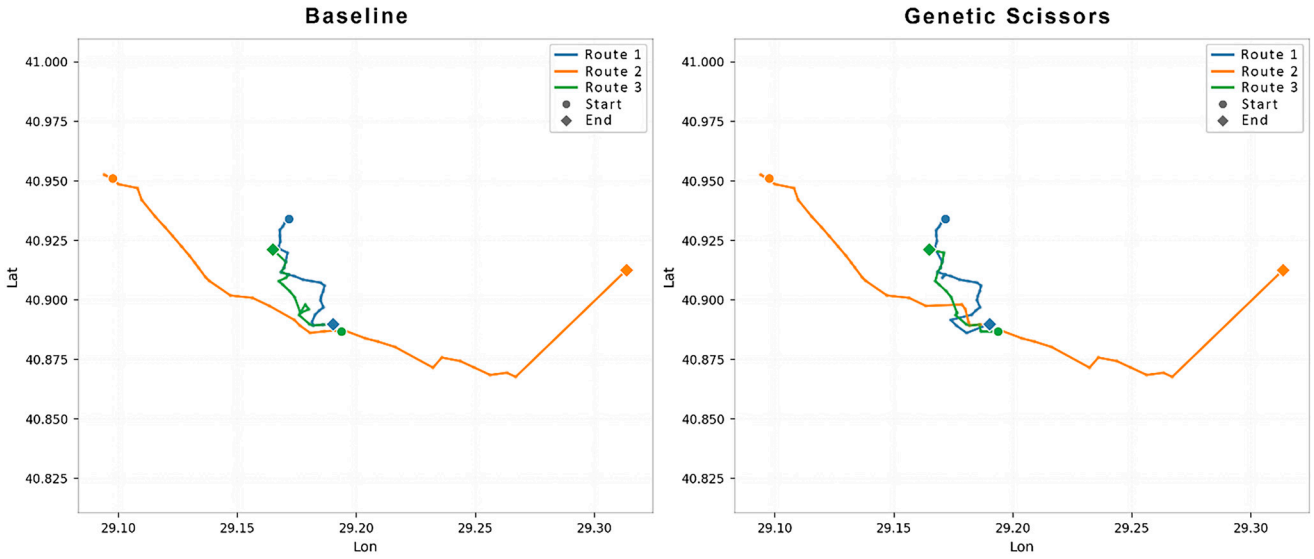


Figure 5. Route comparison for Cluster 250 ($N = 82$, $k = 3$) baseline ($rat = 2.835$) and Genetic Scissors output ($rat = 2.360$, +16.75%).

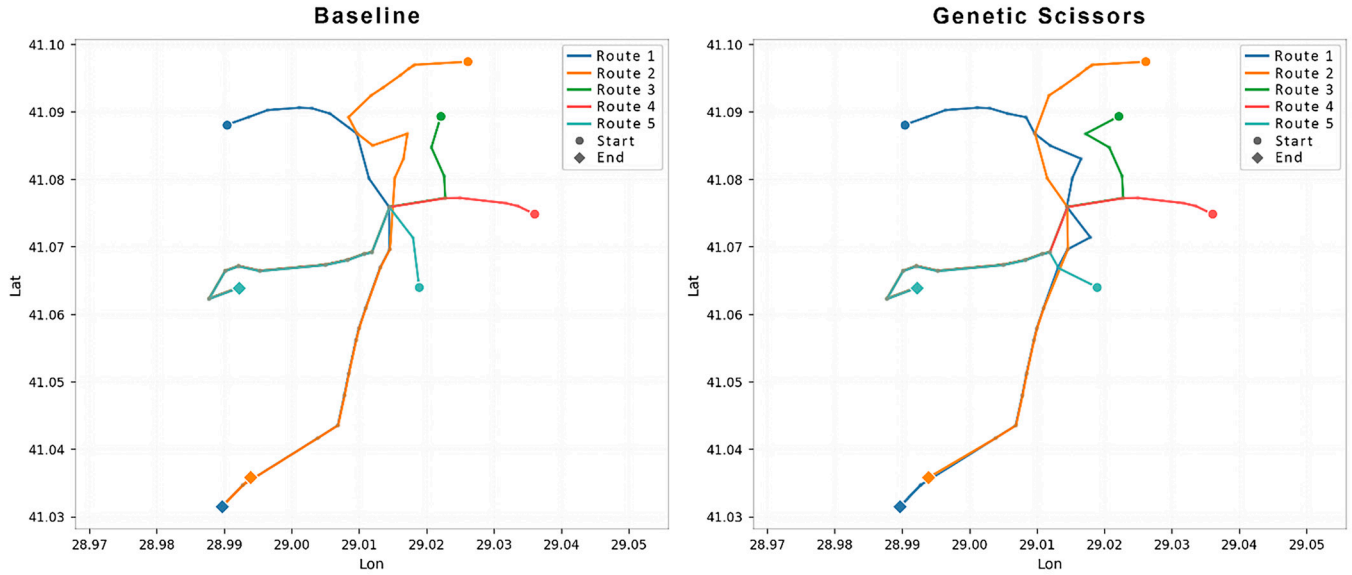


Figure 6. Route comparison for Cluster 117 ($N = 86$, $k = 5$) baseline ($rat = 2.480$) and Genetic Scissors output ($rat = 2.394$, +3.47%).

The examples of Cluster 117 in Figure 6 and Cluster 213 in Figure 8 show that, when the baseline route structure is already relatively direct, the method can still produce limited but consistent gains. The improvement is 3.47% in Cluster 117 and 1.20% in Cluster 213. In these two cases, Genetic Scissors does not fundamentally alter the existing structure, but instead produces fine-grained adjustments. For Cluster 213, the map-based representation in Figure 8 makes the spatial effect of such fine modifications more tangible in the real urban network and allows the structural differences between the baseline and Genetic Scissors versions of the three routes to be observed within a single visual frame.

By contrast, the example of Cluster 18 in Figure 7 shows, quite transparently, that the method does not succeed in every problem instance. However, the ablation results reported in Table 3 show that the AkAy reference method yields a rationality value of 3.267 and the AkAy + GS-IRS combination yields 2.994, whereas Genetic Scissors reaches 2.745, which is lower than all alternative configurations. In other words, although Genetic Scissors cannot improve upon the baseline in this difficult instance, it still produces the best result

among the compared algorithmic structures. The large-scale ($N = 301$) and multi-route ($k = 5$) nature of Cluster 18 points to the limits of the current computational budget, as discussed later.

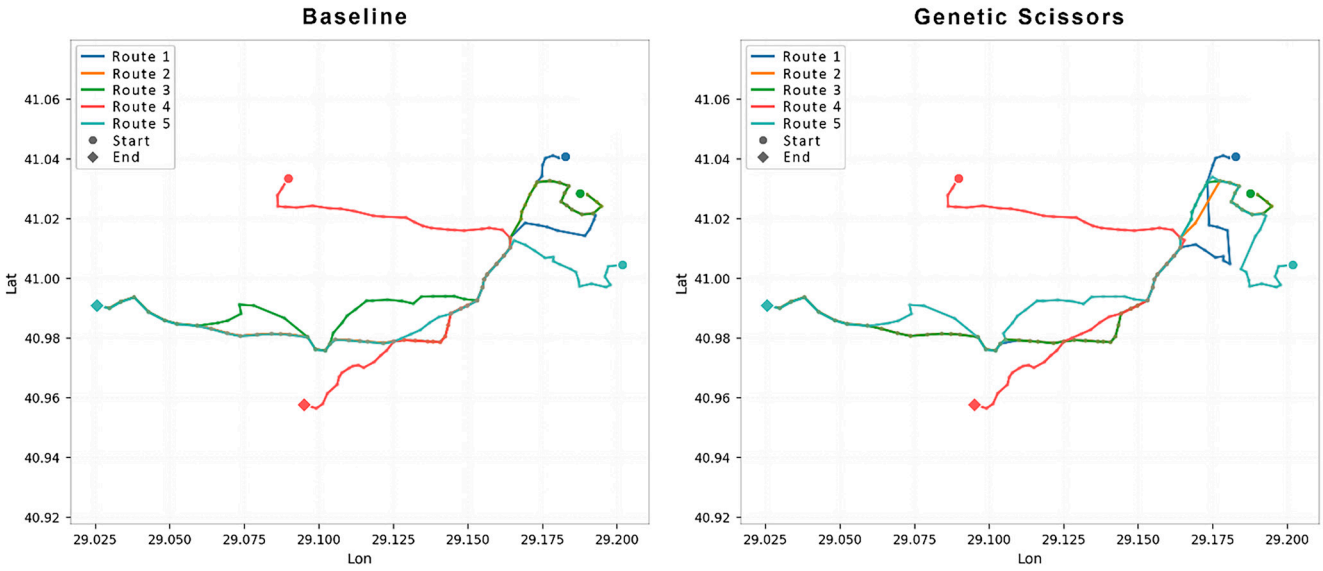


Figure 7. Route comparison for Cluster 18 ($N = 301$, $k = 5$) baseline ($rat = 2.643$) and Genetic Scissors output ($rat = 2.745$, -3.86%).

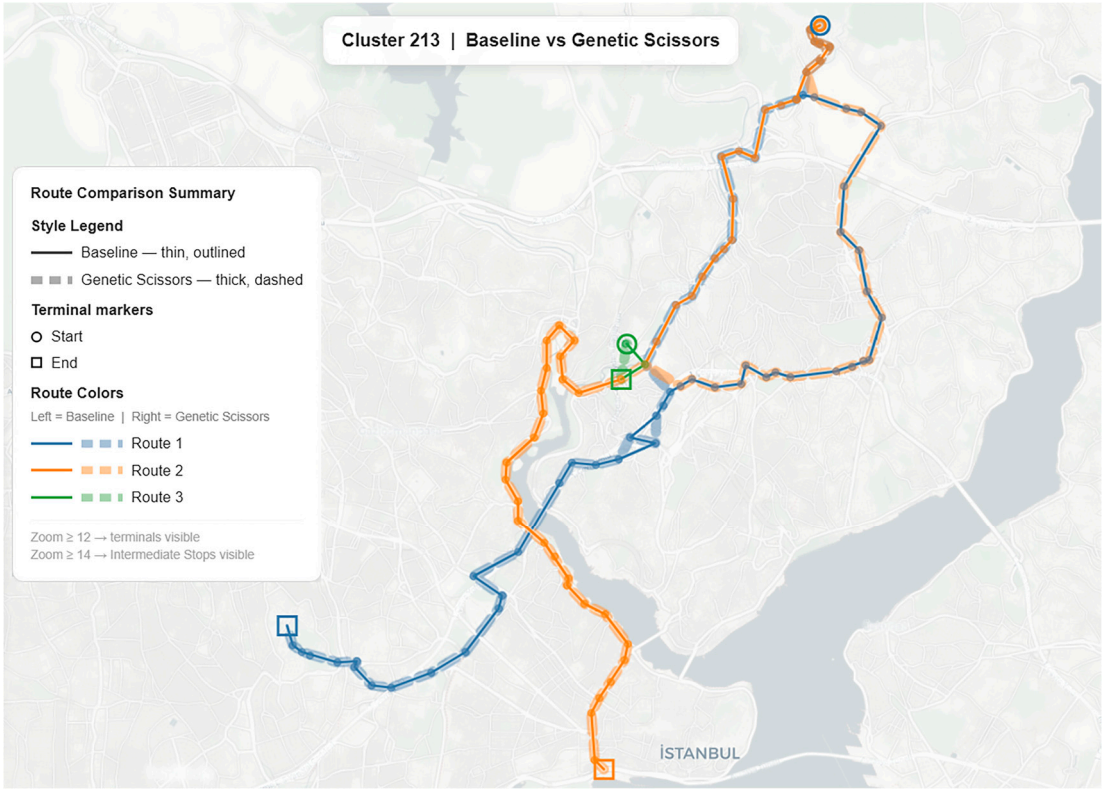


Figure 8. Map-based visualization of the route comparison for Cluster 213 ($N = 109$, $k = 3$) in Istanbul, Türkiye: baseline ($rat = 2.675$, thin solid line) and Genetic Scissors output ($rat = 2.643$, +1.20%, thick dashed line). Circular and square markers indicate route start and end points, respectively. Map data © OpenStreetMap contributors (ODbL); basemap tiles © CARTO.

The five cases reveal three behavioral modes of Genetic Scissors: large-scale rearrangement when the initial geometry permits substantial improvement (Clusters 49, 250);

fine-grained refinement when the baseline is already direct (Clusters 117, 213); and a scale boundary where average performance is constrained by computational budget (Cluster 18). Method performance therefore depends not only on algorithmic design but also on cluster geometry, improvement potential, and problem scale.

6.5. Interpretation of the Findings: Why Does Genetic Scissors Work?

When the experimental findings are considered jointly, the superiority of the Genetic Scissors framework can be attributed to three integrated mechanisms. First, the two-phase fitness function addresses the scale incompatibility between the absolute distance measure and the normalized rationality measure through a controlled phase transition. This effect is clearly visible in the ablation results through the improvement from pt to $pt \rightarrow rat$, which corresponds to a reduction of approximately 17%. Second, the six intensification operators produce complementary interventions at the node, segment, and block levels. This multi-scale structure enables access to regions of the search space that cannot be reached by configurations relying on a single type of operator. Third, the selective application of GS-IRS, governed by the warm-up threshold and the target-individual selection mechanism, provides memetic intensification without spreading the computational cost across the entire population. The integrated operation of these three mechanisms constitutes the main reason why the method produces systematic and statistically significant superiority on the broader evaluation sample of 109 clusters.

It is worth noting that the apparent complexity of the Genetic Scissors framework is not a design preference but a structural response to the extended problem formulation. The same-location constraint and the strict positional integrity of route boundaries fundamentally restrict the feasible solution space. Within such a restricted space, single-scale local search mechanisms, including the operators of the reference method, fail to escape local optima, as confirmed by the ablation analysis in Section 6.1. Each of the six operators (GS-SR, GS-FR, GS-CX, GS-BT, GS-BI, GS-NLS) targets a different scale of intervention (node, segment, or block level), and the selective in-route evolutionary search (GS-IRS) operates at the route level. The novelty of the framework therefore does not lie in any individual operator, many of which build on standard memetic principles, but in the integrated multi-scale architecture specifically tailored to the extended problem structure. The ablation results in Section 6.1 demonstrate that no single component, when used in isolation, reaches the performance of the integrated framework, providing evidence that the observed gains arise from the interaction of these components rather than from a single dominant mechanism.

7. Limitations

Although this study provides a meaningful methodological contribution to public transit route optimization, several limitations should be taken into account when interpreting the results.

The first concerns computational budget and large-scale problems. The ablation results reported in Table 3 show that, in large and multi-route clusters such as Cluster 18 ($N = 301$, $k = 5$), Genetic Scissors outperforms AkAy and all other alternative configurations, yet still fails to improve upon the baseline. This suggests that the applied parameter budget, approximately $n_g \approx 2272$ and $ps \approx 206$ for this cluster, may remain insufficient to explore the solution space adequately at that scale. Longer generation budgets, adaptive early-stopping mechanisms, or problem-informed initialization strategies may improve performance in large-scale instances, but these directions fall outside the scope of the present paper.

The second limitation concerns the structural scope of parameter optimization. The regression-based parameterization builds on the parameter framework of Akgol, Demir, Aydogdu and Murat [7] and transfers it to the extended problem setting. Although this choice is methodologically necessary in order to ensure a fair comparison with the reference method, the present study does not test whether a separate fine-tuning process specifically designed for the Genetic Scissors framework could yield additional performance gains. In particular, operator-specific parameter profiles, such as the application probabilities of individual intensification operators or regression models for the inner ps and n_g values of GS-IRS, represent a promising direction for future work.

The third limitation concerns the scope of the problem, the omission of operational dimensions, and the absence of formal robustness and sensitivity analyses regarding certain structural assumptions. The present study focuses on the spatial directness of route geometry, as represented by the rationality measure. Operational aspects such as capacity constraints, service frequency and scheduling, passenger demand assignment, transfer costs, and detailed operating costs are excluded from the scope of this work. This exclusion is a deliberate design decision intended to preserve methodological focus. Accordingly, the results should be interpreted primarily as a decision-support framework for route geometry rather than as a complete operational optimization model. In this context, the uniform trip-weight matrix ($TM_{ij} = 1$) was adopted to isolate pure geometric directness and ensure comparability across diverse clusters, but it does not reflect actual passenger flow patterns. Similarly, the same-location penalty constant (M = 106) and distance thresholds were set deterministically as mathematical bounds rather than being tuned through sensitivity testing. Future studies should incorporate empirical demand matrices and conduct sensitivity analyses on these structural parameters.

The fourth limitation concerns data source and generalizability. The evaluation was conducted on 213 clusters derived from the Istanbul public transit network, together with the 109-cluster fair-comparison subset. Although this dataset offers substantial variety, with clusters ranging from $N = 3$ to $N = 301$ and route counts ranging from $k = 1$ to $k = 10$, its derivation from a single metropolitan network requires caution when generalizing the findings to other urban structures. Evaluations on networks with different spatial topologies, different density profiles, and different transit systems, such as tramways, metro feeder routes, or BRT corridors, would provide stronger support for the generalizability of the method.

The fifth limitation concerns the filtering threshold. The clusters included in the analysis were determined by the condition $baseline_{rat}(c) < 6$, as defined in Equation (41). This threshold is methodologically justified as a way of excluding structurally extreme clusters arising either from routes in which outbound and inbound movements occur on the same geometric path or from data-quality problems, as discussed earlier. Nevertheless, the extent to which alternative values of θ_f or alternative filtering criteria would affect the results remains an open question that may be addressed through sensitivity analysis in future work.

A further limitation concerns the granularity of the ablation analysis. The configurations evaluated in Section 6.1 treat the six intensification operators (GS-SR, GS-FR, GS-CX, GS-BT, GS-BI, GS-NLS) as a unified architectural block. While this structural-level ablation isolates the contribution of the operator family as a whole, it does not quantify the individual performance attribution of each operator. Disentangling the relative contributions of individual operators and developing operator-specific application probability profiles remain important directions for future work.

A final limitation relates to the benchmarking scope. Because the proposed framework addresses an extended problem structure featuring a fractional rationality objective, fixed

route-boundary positional constraints, and a same-location penalty, direct application of generic state-of-the-art routing solvers (such as LKH or standard ALNS implementations) is not straightforward, these solvers are natively designed for absolute additive cost minimization and would require substantial structural reformulation to handle the fractional objective and rigid boundary constraints adopted here. Consequently, the comparative evaluation in this study was strictly conducted against the most relevant problem-specific reference method (AkAy) alongside an internal ablation analysis. Adapting broader state-of-the-art routing metaheuristics to this extended problem setting to provide a wider benchmark comparison remains an important direction for future research.

These limitations do not weaken the main findings of the study. Rather, they provide the necessary context for interpreting the results and identify clear directions for subsequent research.

8. Conclusions

This study proposed Genetic Scissors, a memetic genetic algorithm framework for improving the spatial directness of public transit routes, and evaluated it on 213 clusters derived from the Istanbul public transit network.

The findings lead to four main conclusions. First, on the 109-cluster fair-comparison sample, Genetic Scissors improved upon the existing route structures in 81.7% of the cases, with a mean Δrat of 2.41%, and this improvement was confirmed to be highly significant by the Wilcoxon signed-rank test ($W = 856,\ p < 0.001,\ r = 0.750$), corresponding to a large effect size. Second, in direct comparison with the reference method [7], Genetic Scissors outperformed the reference in 84 of the 109 clusters, and this superiority was again found to be statistically significant ($p < 0.001,\ r = 0.714$), corresponding to a large effect. Third, the ablation analysis on nine representative clusters showed that the success of the method does not arise from a single dominant component, but from the integrated interaction of the two-phase structure, the multi-scale intensification operators, and the selective in-route evolutionary search mechanism. Fourth, the method exhibited a scale-invariant performance profile with respect to both the number of stops and the number of routes. In particular, the systematic performance loss observed for the reference method in large and multi-route problems disappeared under the Genetic Scissors framework.

Beyond providing a new family of operators, Genetic Scissors offers a memetic architecture tailored to the extended problem structure, addressing real-network constraints such as same-location pairs and positional integrity, with practical potential for operational decision-support systems.

Several directions emerge for future work. In very large-scale clusters, adaptive early-stopping mechanisms, problem-informed initialization strategies, and longer generation budgets may improve search depth. Operator-specific parameter profiles and problem-size-sensitive regression models for the inner GS-IRS process may further enhance performance. The current two-phase structure can be extended to additional objectives, such as travel time, capacity, transfer cost, and operational efficiency, transforming the framework into a multi-objective decision-support tool. Finally, evaluation on public transit networks from different cities (with diverse topologies, densities, and system types such as tramways or BRT) would strengthen the generalizability claim. Beyond these directions, broader benchmark comparisons against state-of-the-art memetic and evolutionary routing algorithms, adapted to the extended problem structure, would enrich the comparative scope. Furthermore, evaluating the framework on transit networks from cities with different topological structures, density profiles, and transit modes (such as tramway feeders or BRT corridors) would strengthen the generalizability claims.

Author Contributions: Conceptualization, K.A.; methodology, K.A. and M.K.; software, K.A.; validation, K.A. and M.K.; formal analysis, K.A. and M.K.; investigation, M.K.; resources, K.A.; data curation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, K.A.; visualization, M.K.; supervision, K.A.; project administration, K.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available because the dataset was derived from operational public transit records that include detailed route geometries and internal stop sequences not intended for unrestricted public release.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GA	Genetic Algorithm
GS	Genetic Scissors
GS-IRS	Genetic Scissors—In-Route Evolutionary Search
DSM	Distance Scaling Matrix
EM	Euclidean Matrix
TM	Trip Matrix

References

1. Pemberton, S. Optimising Melbourne's bus routes for real-life travel patterns. *Case Stud. Transp. Policy* **2020**, *8*, 1038–1052. [CrossRef]
2. Friedrich, M.; Müller-Hannemann, M.; Rückert, R.; Schiewe, A.; Schöbel, A. Robustness Tests for Public Transport Planning. In *Proceedings of the 17th Symposium on Algorithmic Approaches for Transportation Modelling Optimization and Systems-ATMOS*; Technische Universitat Wien: Vienna, Austria, 2017.
3. Soares, P.H. Zone-based public transport route optimisation in an urban network. *Public Transp.* **2021**, *13*, 197–231. [CrossRef]
4. Zhao, F. Bus Transit Service Optimization--The State-of-the-Art, State-of-the-Practice, and Challenges. In *Plan, Build, and Manage Transportation Infrastructure in China*; American Society of Civil Engineers: Reston, VA, USA, 2008; pp. 254–267. [CrossRef]
5. Diana, M.; Quadrifoglio, L.; Pronello, C. A methodology for comparing distances traveled by performance-equivalent fixed-route and demand responsive transit services. *Transp. Plan. Technol.* **2009**, *32*, 377–399. [CrossRef]
6. Harbering, J. Delay resistant line planning with a view towards passenger transfers. *Top* **2017**, *25*, 467–496. [CrossRef]
7. Akgol, K.; Demir, E.; Aydogdu, I.; Murat, Y.S. Efficiency in public transportation: A new flow direction method for optimizing multi-route networks. *Transp. Lett.-Int. J. Transp. Res.* **2025**, *17*, 1320–1332. [CrossRef]
8. Lenstra, J.K.; Kan, A. Complexity of Vehicle-Routing and Scheduling Problems. *Networks* **1981**, *11*, 221–227. [CrossRef]
9. Applegate, D.L.; Bixby, R.E.; Chvátal, V.; Cook, W.J. The traveling salesman problem: A computational study. In *The Traveling Salesman Problem*; Princeton University Press: Princeton, NJ, USA, 2011.
10. Khan, I.; Maiti, M.K. A swap sequence based artificial bee colony algorithm for traveling salesman problem. *Swarm Evol. Comput.* **2019**, *44*, 428–438. [CrossRef]
11. Baker, B.M.; Ayechew, M.A. A genetic algorithm for the vehicle routing problem. *Comput. Oper. Res.* **2003**, *30*, 787–800. [CrossRef]
12. Prins, C. A simple and effective evolutionary algorithm for the vehicle routing problem. *Comput. Oper. Res.* **2004**, *31*, 1985–2002. [CrossRef]
13. Iliopoulou, C.; Kepaptsoglou, K.; Vlahogianni, E. Metaheuristics for the transit route network design problem: A review and comparative analysis. *Public Transp.* **2019**, *11*, 487–521. [CrossRef]
14. Bielli, M.; Caramia, M.; Carotenuto, P. Genetic algorithms in bus network optimization. *Transp. Res. Part C Emerg. Technol.* **2002**, *10*, 19–34. [CrossRef]
15. Pattnaik, S.; Mohan, S.; Tom, V. Urban bus transit route network design using genetic algorithm. *J. Transp. Eng.* **1998**, *124*, 368–375. [CrossRef]
16. Arbex, R.O.; da Cunha, C.B. Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. *Transp. Res. Part B Methodol.* **2015**, *81*, 355–376. [CrossRef]

17. Moscato, P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurr. Comput. Program C3P Rep.* **1989**, *826*, 37.
18. Neri, F.; Cotta, C. Memetic algorithms and memetic computing optimization: A literature review. *Swarm Evol. Comput.* **2012**, *2*, 1–14. [CrossRef]
19. Wilson, A.J.; Pallavi, D.; Ramachandran, M.; Chinnasamy, S.; Sowmiya, S. A review on memetic algorithms and its developments. *Electr. Autom. Eng.* **2022**, *1*, 7–12.
20. Nguyen, Q.H.; Ong, Y.S.; Krasnogor, N. A study on the design issues of Memetic Algorithm. In Proceedings of the IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 2390–2397.
21. Radcliffe, N.J.; Surry, P.D. Formal memetic algorithms. In Proceedings of the AISB Workshop on Evolutionary Computing, Leeds, UK, 11–13 April 1994; pp. 1–16.
22. Roy, A.; Manna, A.; Maity, S. A novel memetic genetic algorithm for solving traveling salesman problem based on multi-parent crossover technique. *Decis. Mak. Appl. Manag. Eng. Lett.* **2019**, *2*, 100–111. [CrossRef]
23. Garg, P. A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm. *arXiv* **2010**, arXiv:1004.0574. [CrossRef]
24. El-Mihoub, T.A.; Hopgood, A.A.; Nolle, L.; Battersby, A. Hybrid Genetic Algorithms: A Review. *Eng. Lett.* **2006**, *13*, 124–137.
25. Lightner-Laws, C.; Agrawal, V.; Lightner, C.; Wagner, N. An evolutionary algorithm approach for the constrained multi-depot vehicle routing problem. *Int. J. Intell. Comput. Cybern.* **2016**, *9*, 2–22. [CrossRef]
26. Kouretas, K.; Kepaptsoglou, K. Planning Integrated Unmanned Aerial Vehicle and Conventional Vehicle Delivery Operations under Restricted Airspace: A Mixed Nested Genetic Algorithm and Geographic Information System-Assisted Optimization Approach. *Vehicles* **2023**, *5*, 1060–1086. [CrossRef]
27. Xu, Y.; Han, Y.X.; Sun, Z.; Gu, W.; Jin, Y.K.; Xue, X.Y.; Lan, Y.B. Path Planning Optimization With Multiple Pesticide and Power Loading Bases Using Several Unmanned Aerial Systems on Segmented Agricultural Fields. *IEEE Trans. Syst. Man Cybern.-Syst.* **2023**, *53*, 1882–1894. [CrossRef]
28. Greco, A.; Cannizzaro, F.; Bruno, R.; Pluchino, A. A nested genetic algorithm strategy for an optimal seismic design of frames. *Comput. Optim. Appl.* **2024**, *87*, 677–704. [CrossRef]
29. Santini, A.; Schneider, M.; Vidal, T.; Vigo, D. Decomposition Strategies for Vehicle Routing Heuristics. *Inf. J. Comput.* **2023**, *35*, 543–559. [CrossRef]
30. Debels, D.; Vanhoucke, M. A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem. *Oper. Res.* **2007**, *55*, 457–469. [CrossRef]
31. Zhang, Y.; Mei, Y.; Zhang, B.; Jiang, K. Divide-and-conquer large scale capacitated arc routing problems with route cutting off decomposition. *Inf. Sci.* **2021**, *553*, 208–224. [CrossRef]
32. Kerscher, C.; Minner, S. Spatial-temporal-demand clustering for solving large-scale vehicle routing problems with time windows. *arXiv* **2024**, arXiv:2402.00041.
33. Akbari, R.; Ziarati, K. A multilevel evolutionary algorithm for optimizing numerical functions. *Int. J. Ind. Eng. Comput.* **2011**, *2*, 419–430. [CrossRef]
34. Grudniewski, P.A.; Sobey, A.J. Behaviour of Multi-Level Selection Genetic Algorithm (MLSGA) using different individual-level selection mechanisms. *Swarm Evol. Comput.* **2019**, *44*, 852–862. [CrossRef]
35. Wu, S.X.; Banzhaf, W. A hierarchical cooperative evolutionary algorithm. In Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, Portland, OR, USA, 7–11 July 2010; pp. 233–240.
36. Sefrioui, M.; Périaux, J. A hierarchical genetic algorithm using multiple models for optimization. In Proceedings of the International Conference on Parallel Problem Solving From Nature, Paris, France, 18–20 September 2000; pp. 879–888.
37. Li, Q.S.; Liu, D.K.; Leung, A.Y.T.; Zhang, N.; Luo, Q.Z. A multilevel genetic algorithm for the optimum design of structural control systems. *Int. J. Numer. Methods Eng.* **2002**, *55*, 817–834. [CrossRef]
38. Sohrabi, M.; Fathollahi-Fard, A.M.; Gromov, V.A. Genetic Engineering Algorithm (GEA): An Efficient Metaheuristic Algorithm for Solving Combinatorial Optimization Problems. *Autom. Remote Control* **2024**, *85*, 252–262. [CrossRef]
39. Meniz, B. *Dinamik Yapıda Yeni Bir Genetik Algoritma Önerisi: Seçilim Operatörü Kararcı Genetik Algoritma (SOKGA) ve Asimetrik Kapasiteli Araç Rotalama Problemi Üzerine Bir Uygulama*; Yıldız Teknik Üniversitesi: İstanbul, Turkey, 2024.
40. Kralev, V.; Kraleva, R. Combining Genetic Algorithm with Local Search Method in Solving Optimization Problems. *Electronics* **2024**, *13*, 4126. [CrossRef]
41. Baressi Šegota, S.; Frank, D.; Lorencin, I.; Anđelić, N. Memetic Algorithm for Energy Optimization in Point-to-Point Robotized Operations. *Machines* **2026**, *14*, 35. [CrossRef]
42. Al Tamseh, A.; Osama, A.; Hussain, M.; Alsobky, A. Methodology for Selection of Sustainable Public Transit Routes: Case Study of Amman City, Jordan. *Infrastructures* **2024**, *9*, 147. [CrossRef]
43. Jin, H.; Li, Z.; Wang, G.; Zhang, S. Sustainable Customized Bus Services: A Data-Driven Framework for Joint Demand Analysis and Route Optimization. *Sustainability* **2026**, *18*, 250. [CrossRef]

44. Cai, J.; Li, Z.; Long, S. Integrated Optimization of Route and Frequency for Rail Transit Feeder Buses under the Influence of Shared Motorcycles. *Systems* **2024**, *12*, 263. [CrossRef]
45. Jiang, H.; Wang, J.; Song, B.; Xiang, X.; Ding, J.; Zhang, X. GTEA: A Game-Theoretic Evolutionary Algorithm for Solving Vehicle Routing Problem with Time Windows Under Uncertain Travel Times. *IEEE Trans. Intell. Transp. Syst.* **2026**, 1–15. [CrossRef]
46. Jiang, H.; Ye, Y.; Wang, C.; Xiang, X.; Zhou, T.; Zhang, X. A Data-Driven Evolutionary Algorithm for Dynamic Vehicle Routing Problems With Time Windows Under Limited Computational Time. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 22239–22250. [CrossRef]
47. Jiang, H.; Ai, C.; Wang, C.; Xiang, X.; Zhang, X. A Surrogate-Assisted Bi-Level Evolutionary Algorithm for Multi-Depot Vehicle Routing Problems with Uncertain Demand. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 16854–16868. [CrossRef]
48. Jinek, M.; Chylinski, K.; Fonfara, I.; Hauer, M.; Doudna, J.A.; Charpentier, E. A Programmable Dual-RNA-Guided DNA Endonuclease in Adaptive Bacterial Immunity. *Science* **2012**, *337*, 816–821. [CrossRef]
49. Bergstra, J.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
50. Cohen, J. *Statistical Power Analysis for the Behavioral Sciences*; Lawrence Erlbaum Associates: Hillsdale, NJ, USA, 1988.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.