Optimization of the traveling salesman problem through a genetic algorithm guided by self-organizing maps

Fahad Rafique[1,*], **Hengjian Cui**[1], **Abid Hussain**[2], and **Sadaf Amin**[1]

[1] *School of Mathematical Sciences, Capital Normal University Beijing, West Third Ring Road North, Haidian District, 105, 100048, Beijing, China.*
[2] *Department of Statistics, PMAS-Arid Agriculture University Rawalpindi, Shamsabad, Muree Road Rawalpindi, Pakistan.*
E-mail: ⟨fahadgchaudry@gmail.com, hjcui@bnu.edu.cn, abid0100@gmail.com, 4200538002@cnu.cdu.cn⟩

Abstract. Optimization is a fundamental process for achieving objectives, exemplified by the Traveling Salesman Problem (TSP), which minimizes resource consumption. This study focuses on optimizing travel routes to reduce fuel costs and maximize tourist visits through efficient time management. We employ a genetic algorithm (GA), a powerful optimization technique, to determine routes. A key challenge in GAs is premature convergence, which can prevent optimal solutions. To address this, we introduce a novel approach integrating Self-Organizing Maps with GAs, specifically through a new selection operator designed to enhance GA performance. An application was developed using real-world coordinates of Pakistani cities. Simulation results demonstrate that our proposed method significantly outperforms existing selection operators in terms of final best distance, average best distance, standard deviation, and computational time. Field validation further confirmed an 18.3% distance reduction and 4.53 million in annual savings in a logistics case study.

Keywords: genetic algorithms; premature convergence; selection operators; self-organizing map; traveling salesman problem.

Received: April 14, 2025; accepted: November 4, 2025; available online: February 9, 2026

DOI: 10.17535/crorr.2026.0014

Original scientific paper.

---

1. Introduction and Background

In contemporary problem-solving, identifying a solution is insufficient; solutions must be optimized for performance within constraints like limited resources, time, and budget. Optimization techniques are thus developed to find the best possible solution efficiently [20, 21]. As traditional methods often fail with complex real-world problems [1], meta-heuristic algorithms provide approximate solutions for high-dimensional, intractable problems in fields like finance, routing, and scheduling [13].

A key combinatorial challenge is the TSP, which seeks the shortest route visiting each city exactly once and returning to the origin effectively the minimal Hamiltonian cycle in a weighted graph. The TSP is NP-hard, with variations like the Hamiltonian path problem where return is unnecessary [5, 19]. Determining a Hamiltonian path's existence is NP-complete, illustrating the TSP's complexity and relevance to real-world issues like trip planning.

GAs, inspired by natural selection [9], are meta-heuristic techniques valued for global search and robustness [2]. A standard GA involves: (I) initial population creation, (II) fitness evaluation, (III) selection of fitter parents, (IV) crossover to create offspring, and (V) mutation

*Corresponding author.

This is an open access article under the CC BY-NC-ND 4.0 license
http://www.hdoi.hr/crorr-journal ©2026 Copyright of this article is retained by the author(s).

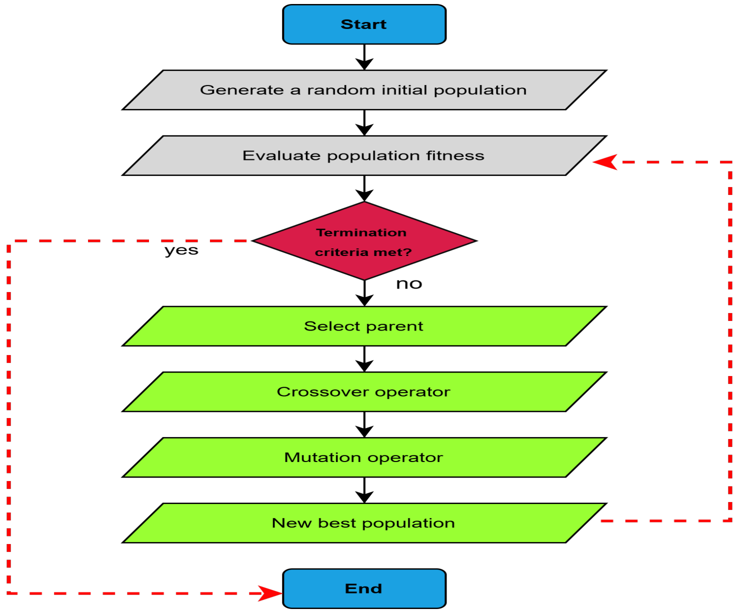

Figure 1: *Typical flowchart of standard GA.*

to prevent premature convergence. This iterative process aims to achieve optimal solutions [18], as shown in Figure 1. However, GAs face premature convergence due to low population diversity, trapping them in local optima [10, 11, 12, 16]. The selection operator is thus critical for maintaining diversity and balancing exploration with exploitation.

This research introduces a novel selection operator to enhance GA convergence by integrating a SOM, forming a hybrid GA-SOM approach. The SOM, an artificial neural network, organizes individuals by normalized fitness ranking, dynamically balancing exploration and exploitation. Validated on TSP and a Pakistan logistics case study where fuel costs are 60% of operational expenses our method reduces journey distance by 18.3% for TCS Express, the largest local provider, servicing 200+ cities with 800 vehicles. Table 1 summarizes related work and our contribution.

2. Selection procedure

The selection process in Genetic Algorithms (GAs) comprises two primary phases. Initially, selection probabilities are assigned to each individual based on their fitness values. This assignment generates a probability vector, given by $P = (p_1, p_2, \cdots, p_N)$, where $p_i \in [0, 1]$ and $\sum_{i=1}^{N} p_i = 1$, with N representing the population size. For a comprehensive understanding of selection probability assignment, readers may refer to [10] and [12]. Subsequently, a sampling procedure is utilized to identify the most suitable parents from the current population for reproduction. This procedure adheres to the Darwinian principle of "survival of the fittest."

Beyond theoretical performance, the efficacy of selection operators is proven in applied Operational Research (OR) contexts like logistics and vehicle routing. Recent work emphasizes hybrid GA approaches to handle real-world constraints. For instance, [14] surveyed hybrid metaheuristics for complex humanitarian vehicle routing, while [7] combined RL with a GA for electric vehicle routing and charging. This trend of hybridizing GAs for specialized logistics challenges motivates our work. We contribute to this domain by integrating SOMs to enhance GA scalability and convergence for large-scale routing problems, as validated in our case study.

This research significantly contributes to the evolution of GA selection methodologies by

Study	Year	Method	Key Features	Limitations
Applegate et al. [3]	2006	Exact (Branch-and-Bound)	Comprehensive TSP study; exact solutions.	High computational cost for large-n.
Whitley et al. [23]	2012	GA selection survey	Review of selection mechanisms.	No novel operator proposed.
Dorigo et al. [6]	2016	ACO	Meta-heuristic TSP solver.	Premature convergence.
Zelinka [24]	2016	GA-SOM	SOM clustering + GA optimization.	Static selection operators.
Papa et al. [17]	2021	Adaptive GA	Dynamic rank-based selection.	Standalone GA only.
Abualigah et al. [1]	2022	Hybrid GA	Survey of hybrid frameworks.	Limited focus on selection operators.
Maroof et al. [14]	2023	Stochastic TSP	Real-world application (developing economies).	Traditional methods.
Ebrahimi et al. [7]	2024	RL + ES (Hybrid)	EV routing	Charging; battery life optimization. & Computational complexity.
Tang et al. [22]	2024	GA + heuristics	Jumping gene operators for TSP.	No hybrid frameworks.
Proposed	2025	**GA-SOM + RRS**	**Novel RRS; Hybrid GA-SOM; OX/CX comparison; TSPLIB + Case-Study validation.**	—

Table 1: *Summary of existing literature and contributions*

proposing a novel selection operator. This new operator is expected to enhance the characteristic traits of the population while optimizing the balance between exploitation and exploration.

2.1. Assignment of probability

This study utilizes a selection of established selection operators, the fundamental characteristics of which are detailed below.

Fitness Proportional Selection (FPS), initially proposed by [9], assigns selection probabilities to individuals in a population in direct proportion to their fitness values. The selection probability, p_i, for the i^{th} individual is calculated as

$$p_i = \frac{1}{N-1}\left(1 - \frac{f_i}{\sum_{j=1}^{N} f_j}\right); \quad i \in \{1, 2, \cdots, N\}, \tag{1}$$

where f_i represents the fitness of the i^{th} individual, and N denotes the population size. While FPS offers conceptual simplicity, it is recognized for its susceptibility to scaling issues, as discussed by [8].

To mitigate the issue of premature convergence often associated with FPS, [4] introduced Linear Rank Selection (LRS). LRS aims to apply a more graduated selection pressure, thereby facilitating the selection of individuals with lower fitness. In LRS, the selection probability, p_i, for the i^{th} ranked individual is determined by

$$p_i = \frac{1}{N}\left(\theta^- + (\theta^+ - \theta^-)\frac{i-1}{N-1}\right); \qquad i \in \{1, 2, ..., N\}, \tag{2}$$

where i denotes the individual's rank, and θ^- and θ^+ represent the selection probabilities assigned to the worst and best-ranked individuals, respectively. These parameters are subject to the constraint $\theta^- + \theta^+ = 2$. However, despite its rank-based approach, LRS may still struggle to effectively capture subtle fitness differences, potentially reducing the discriminatory power of the selection process and compromising the representation of significant fitness variations.

To address the limitations of LRS, [15] proposed an alternative rank-based selection operator: Exponential Rank Selection (ERS). Unlike LRS, ERS assigns selection probabilities that increase exponentially with fitness rank, from the least to the most fit individual. The selection probability for an individual ranked i^{th} is defined as

$$p_i = \frac{\gamma^{N-i}(1-\gamma)}{1-\gamma^N}, \quad i \in \{1, 2, \cdots, N\}, \tag{3}$$

where $0 < \gamma < 1$, and γ represents a constant ratio reflecting the relative weights assigned to individuals based on their fitness ranks. [15] recommends values of γ approaching unity to maximize selection gain.

[12] employed a probabilistic threshold mechanism for selecting tournament winners, termed Probabilistic 2-Tournament Selection (PTS). In this scheme, the winner advances with a probability q, where $0.5 < q < 1$, while the loser is given a subsequent opportunity to compete with a probability of $1 - q$. The selection probability for the i^{th} ranked individual within the PTS method is defined by the following equation

$$p_i = \frac{2(i-1)}{N(N-1)}q + \frac{2(N-i)}{N(N-1)}(1-q), \quad i \in \{1, 2, \cdots, N\}. \tag{4}$$

[16] introduced a fitness-based selection operator. In their procedure, m_i be the size of i-*th* individual, where $i = 1, 2, \cdots, N$. For the i-*th* individual, f_i be the corresponding fitness value. Then,

$$M_{f,t} = \text{Median}(f_{(1)}, f_{(2)}, f_{(3)}, \cdots, f_{(k)}),$$

where $f_{(1)}, f_{(2)}, f_{(3)}, \cdots, f_{(k)}$ are the fitness values arranged in ascending order of magnitude, i.e., from worst to best individuals. The expected number of individuals at the next generation will be

$$M_{i,t+1} = m_{i,t} \cdot n \cdot P_{i,t},$$

where the total number of individuals sums to n. The probability of selection for the i-th individual at the t-th generation is

$$P_{i,t} = \frac{f_i + M_{f,t}}{\sum_{j=1}^{k}(m_{j,t}f_j + M_{f,t})}, \quad i = 1, 2, 3, \cdots, N, \tag{5}$$

where f_i is the fitness value of the i-*th* individual and $M_{f,t}$ is the median of the current population at the t-*th* generation.

3. Proposed selection operator

3.1. The rationale

The existing literature describes various selection strategies for GAs. Two common methods, LRS and FPS, each present distinct trade-offs. LRS prioritizes population diversity, which can lead to a slower convergence rate due to reduced selection pressure. Conversely, FPS exerts strong selection pressure, potentially compromising diversity and increasing the risk of premature convergence. To address these limitations, this study adopts the relative rank methodology proposed by [10]. This approach preserves the magnitude differences between successive fitness values. We integrate these relative ranks into a linear ranking operator, creating a hybrid methodology that combines the benefits of both FPS and LRS. This new method aims to achieve a more effective balance between exploration and exploitation within the evolutionary process, maintaining adequate selection pressure while preserving population diversity. Furthermore, we incorporate SOMs to enhance solution initialization and improve scalability for future optimization efforts. This integration contributes to a more robust and efficient GA system. The following sections will detail the fundamental components and operational procedures of our proposed GA system.

3.2. Relative-rank-based selection operator

The Relative-Rank-Based Selection (RRS) operator is designed to assign selection probabilities to individuals in a population while providing explicit control over the trade-off between selection pressure and population diversity. This is achieved by first establishing a nuanced "relative rank" for each individual, which accounts for its position within the ordered fitness landscape, and then mapping these ranks to probabilities using adjustable parameters.
Here's how RRS operates:

- Let $\Omega = (f_1, f_2, \cdots, f_N)$ represent the ascendingly ordered set of fitness values for all individuals in the population.
- The relative rank of the worst individual, $\mathcal{R}_1^{(r)} = 1$, is defined as 1.
- For individuals $i = 2, 3, \cdots, N$, the subsequent expression

  $$z_i = i - 1 + \frac{(N-1)(f_{(i)} - f_{(i-1)})}{f_{(N)} - f_{(1)}},$$

  is used to calculate an intermediate value that incorporates both the individual's integer rank and its fitness proximity to neighbors.
- The relative rank $\mathcal{R}_i^{(r)}$ for $i = 2, 3, \cdots, N$ is then defined as the $(i-1)$th smallest value of $\{z_2, z_3, \ldots, z_N\}$. This meticulous assignment ensures a strict ordering of relative ranks:

  $$\mathcal{R}_1^{(r)} < \mathcal{R}_2^{(r)} < \cdots < \mathcal{R}_N^{(r)}. \tag{6}$$

- Finally, selection probabilities p_i are assigned using the expression:

  $$p_i = \frac{1}{N}\left(\alpha^- + (\alpha^+ - \alpha^-)\frac{\mathcal{R}_i^{(r)} - 1}{N-1}\right); \qquad i \in \{1, 2, ..., N\}. \tag{7}$$

  Here, α^- and α^+ denote the selection probabilities assigned to the worst and best-ranked individuals, respectively. These parameters are constrained by $\alpha^- + \alpha^+ = 2$, ensuring a balanced probability distribution.

3.3. Validating the RRS probability distribution

1. The assigned probabilities must form a complete probability distribution, meaning their sum across all individuals must equal 1.

For any set of probabilities to be valid, their sum must always be 1. In the context of RRS, this means that even though we're assigning probabilities based on relative ranks and using adjustable parameters (α^-, α^+), the scaling within the formula must naturally ensure that all chances of selection add up to 100%. The algorithmic ingenuity resides in the interplay between its linear time complexity and the invariant structural property of the relative rank sum, $(\mathcal{R}_i^{(r)} = \frac{N(N+1)}{2})$ work together to guarantee this sum, regardless of the population size N or the specific values of α^- and α^+. Essentially, the design ensures that increasing the probability of the best-ranked individuals by adjusting α^+ is precisely balanced by a corresponding decrease in the probabilities of the worse-ranked individuals, all while maintaining the total sum of 1. Its formal derivation is provided in Result 1.

Result 1. The expression (7) is a complete probability distribution, i.e. $\sum_{i=1}^{N} p_i = 1$.

Proof: We begin by summing the probability expression (7) over all individuals i=1,...,N:

$$\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} \frac{1}{N}\left(\alpha^- + (\alpha^+ - \alpha^-)\frac{\mathcal{R}^{(r)}i - 1}{N-1}\right)$$

We can factor out the common term $\frac{1}{N}$ and separate the summation:

$$= \frac{1}{N}\left(\sum i = 1^N \alpha^- + \sum_{i=1}^{N}(\alpha^+ - \alpha^-)\frac{\mathcal{R}^{(r)}i - 1}{N-1}\right)$$

Since α^- is a constant, $\sum i = 1^N \alpha^- = N\alpha^-$. We can also factor out the constant term $\frac{\alpha^+ - \alpha^-}{N-1}$:

$$= \frac{1}{N}\left(N\alpha^- + \frac{\alpha^+ - \alpha^-}{N-1}\sum_{i=1}^{N}(\mathcal{R}_i^{(r)} - 1)\right)$$

$$= \frac{1}{N}\left(N\alpha^- + \frac{\alpha^+ - \alpha^-}{N-1}\left(\sum_{i=1}^{N}\mathcal{R}_i^{(r)} - N\right)\right)$$

At this juncture, we leverage a previously established result by [10], which proves that the sum of the relative ranks $\mathcal{R}_i^{(r)}$ for $i = 1, 2, \cdots, N$ is equivalent to the sum of the first N natural numbers: $\sum_{i=1}^{N}\mathcal{R}_i^{(r)} = \frac{N(N+1)}{2}$. Substituting this identity into our expression:

$$= \frac{1}{N}\left(N\alpha^- + \frac{\alpha^+ - \alpha^-}{N-1}\left(\frac{N(N+1)}{2} - N\right)\right)$$

$$= \frac{1}{N}\left(N\alpha^- + \frac{\alpha^+ - \alpha^-}{N-1}\left(\frac{N(N-1)}{2}\right)\right)$$

We can cancel out the $(N-1)$ term in the numerator and denominator:

$$= \frac{1}{N}\left(N\alpha^- + \frac{\alpha^+ - \alpha^-}{1}\left(\frac{N}{2}\right)\right)$$

$$= \frac{1}{N}\left(N\alpha^- + \frac{N(\alpha^+ - \alpha^-)}{2}\right)$$

Finally, applying the constraint $\alpha^- + \alpha^+ = 2$, the desire result can be obtained.

2. The selection must adhere to Darwin's "survival-of-the-fittest" principle, where fitter individuals are assigned a higher or equal probability of selection than less fit individuals ($p_j \geq p_i$ for $j > i$).

It should align with Darwin's survival-of-the-fittest principle, specifically represented as $p_{i+1} > p_i$ for $j > i$ where both i and j are integers falling within the range of $1, 2, \cdots, N$. Its formal derivation is provided in Result 2.

Result 2. According to Darwin's theory, the expression (7) should follows the survival-of-fittest-criterion, i.e. $p_{i+1} > p_i$.

Proof: As $p_{i+1} > p_i$

$$\frac{1}{N}\left(\alpha^- + (\alpha^+ - \alpha^-)\frac{\mathcal{R}_{i+1}^{(r)} - 1}{N-1}\right) > \frac{1}{N}\left(\alpha^- + (\alpha^+ - \alpha^-)\frac{\mathcal{R}_i^{(r)} - 1}{N-1}\right)$$

$$\left((\alpha^+ - \alpha^-)\frac{\mathcal{R}_{i+1}^{(r)} - 1}{N-1}\right) > \left((\alpha^+ - \alpha^-)\frac{\mathcal{R}_{i}^{(r)} - 1}{N-1}\right)$$

$$\left((\alpha^+ - \alpha^-)\frac{\mathcal{R}_{i+1}^{(r)}}{N-1}\right) > \left((\alpha^+ - \alpha^-)\frac{\mathcal{R}_{i}^{(r)}}{N-1}\right)$$

$$\mathcal{R}_{i+1}^{(r)} > \mathcal{R}_{i}^{(r)}.$$

This inequality is always true, it can be seen in equation (7). Thus, individuals with better fitness are assigned higher selection probabilities.

4. Adopted methodology

This study employs two main computational approaches to solve the TSP.

A genetic algorithm (GA) maintains a population of candidate routes, each encoded as an ordered city list. The population evolves through selection, crossover, and mutation, with fitness measured by total route distance. Selection favors shorter routes to serve as parents, crossover merges parent segments to produce valid offspring, and mutation introduces random changes to maintain diversity and avoid premature convergence. This process iterates until a termination condition is met, aiming to approach the optimal minimal-distance route (see Algorithm 1 in the online appendix).

The second method integrates the GA with a Self-Organizing Map (SOM), an artificial neural network inspired by the animal visual cortex. The SOM uses competitive learning to reduce dimensionality while preserving input topology. Nodes' weight vectors $\mathbf{W}_v$ are updated iteratively based on the best matching unit u for input $D(t)$ by

$$\mathbf{W}_v(s+1) = \mathbf{W}_v(s) + \theta(u,v,s) \cdot \alpha(s) \cdot (D(t) - \mathbf{W}_v(s)),$$

where s is the current iteration, θ is the neighborhood function, and α is the learning rate. Traditional static SOM structures may underperform on complex data, so the GA optimizes the SOM architecture, resulting in the hybrid GA-SOM method that adapts to dataset complexity. This approach improves mapping quality, verified by error metrics across multiple datasets (see Algorithm 2 in the online appendix).

5. Experimental setup

All experiments were conducted on a computing system running Windows 11. The system was equipped with an AMD Ryzen 7 4700U processor featuring Radeon Graphics and 32 GB of RAM. The application was developed in R (version 4.4.2), leveraging the following libraries: `parallel`, `ggplot2`, `Rfast`, `TSP`, and `kohonen`. Effectively addressing a problem necessitates not only a meticulous selection of appropriate algorithms but also a judicious choice of parameters. The significant impact of parameter selection on the results is demonstrated in (see Table A2 in the online appendix).

6. Results and discussion

A summary of key results for the most effective method combinations is presented in Table 2. Comprehensive results for all operators, crossover methods, and problem sizes are provided in the Online Appendix (Tables 4-8).

This section evaluates the performance of the proposed Relative-Rank-Based Selection (RRS) operator, both within a standalone Genetic Algorithm (GA) and a hybrid GA-Self

Organizing Map (GA-SOM) framework. Performance is assessed using benchmark TSPLIB instances and a real-world case study with Pakistani city coordinates, measured by solution quality (FinalBest, AvgBest), stability (Standard Deviation, SD), and computational efficiency (Average Run Time, ART).

6.1. Overall performance and comparative analysis

The proposed RRS operator consistently outperformed existing selection operators (FPS, LRS, ERS, PTS) across all problem sizes and configurations. A summary of key results for the most effective method combinations is presented in Table 2.

Problem Context	**Method Combination**	**FinalBest (km)**	**AvgBest (km)**	**SD (km)**	**Time (s)**
TSPLIB ($n = 500$)	RRS (GA-SOM + CX)	1910.56	1950.12	103.44	4.21
TSPLIB ($n = 300$)	RRS (GA + OX)	2042.73	2089.45	115.80	1.85
Pakistan ($n = 200$)	RRS (GA-SOM + CX)	612.72	628.15	13.08	0.46
Pakistan ($n = 200$)	RRS (GA + CX)	629.06	645.91	15.50	0.17

Table 2: *Summary of Key Results for Top-Performing RRS Combinations*

Standalone GA Performance: When integrated into the standard GA, RRS demonstrated superior solution quality and faster convergence than other operators. For example, on the Pakistan cities dataset, RRS with Cycle Crossover (CX) found a 629 km route, significantly shorter than routes found by FPS (720 km) or LRS (∼690 km). Furthermore, RRS reduced the average runtime by approximately 30% compared to other operators, highlighting its computational efficiency.

Hybrid GA-SOM Performance: The integration of RRS within the GA-SOM framework yielded the most robust and effective results, particularly for larger problems. The GA-SOM's ability to pre-cluster cities reduced the search space complexity by an estimated 60%, leading to more stable convergence and higher-quality solutions. As shown in Table 2, **RRS (GA-SOM + CX)** achieved the best overall performance, delivering the lowest FinalBest and AvgBest distances with the highest stability (lowest SD). This combination effectively balanced exploration and exploitation, preventing premature convergence.

6.2. Real-World logistics case study and economic impact

The practical value of the proposed method was validated through a composite case study based on the operations of Trazum Courier Service (TCS) in Pakistan.

This optimization translates to substantial economic and environmental savings:

- **Annual Fuel Savings:** Calculated as $\frac{750-612.72}{8}$ liters × \$1.10/liter × 800 vehicles × 300 days = \$**4.53 Million**.

- **CO_2 Reduction:** Estimated at **28.5 tons** per day.

Note: Calculations of Saved fuel = $\frac{750-612.72}{8}$ liters; Emissions = 2.68kg CO_2/liter.

The logistics application shows even greater practical impact. For 200 cities, RRS (GA-SOM + CX) achieves 18.3% shorter routes than current industry operations shown in Table 3, translating to \$4.53M annual savings for a typical medium-sized operator.

Method	Distance (km)	Fuel Cost (USD/day)	Savings
TCS Current	750	103.13	-
FPS (GA)	720.10	99.01	4.0%
RRS (GA-SOM + CX)	**612.72**	**84.25**	**18.3%**

Table 3: *Economic Impact of optimized routes (200 cities)*

6.3. Discussion of optimal combinations and visual analysis

The comparative performance of different RRS configurations reveals distinct advantages for specific scenarios, supported by comprehensive visual evidence in Table 4 and Figures 3-4.

RRS (GA-SOM + CX) is the recommended combination for large-scale, critical routing problems where solution quality and reliability are paramount. Its low SD indicates consistent performance, while Figure 3 visually demonstrates the more efficient route pathing achieved by this hybrid approach.

RRS (GA + CX) is the optimal choice when computational speed is the highest priority, especially for smaller problems, offering a good balance of speed and solution quality. The computational efficiency of RRS-based approaches is clearly illustrated in Figure 2, which ranks selection methods by average run times.

RRS (GA + OX) performed well for medium-sized problems, demonstrating the versatility of the RRS operator. The comprehensive comparison in Table 4 further validates the consistent superiority of RRS operators across different crossover methods and problem sizes.

These visual representations collectively highlight the computational efficiency of the RRS operator compared to existing methods and demonstrate the practical applicability of the optimized routes generated by both standalone RRS and hybrid GA-SOM approaches.

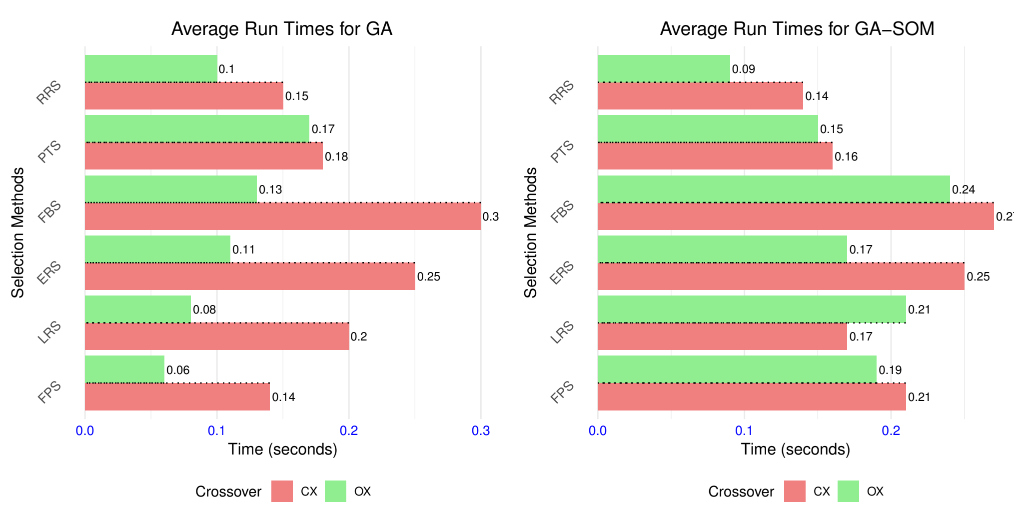

Figure 2: *Ranking of Selection methods by average run times.*

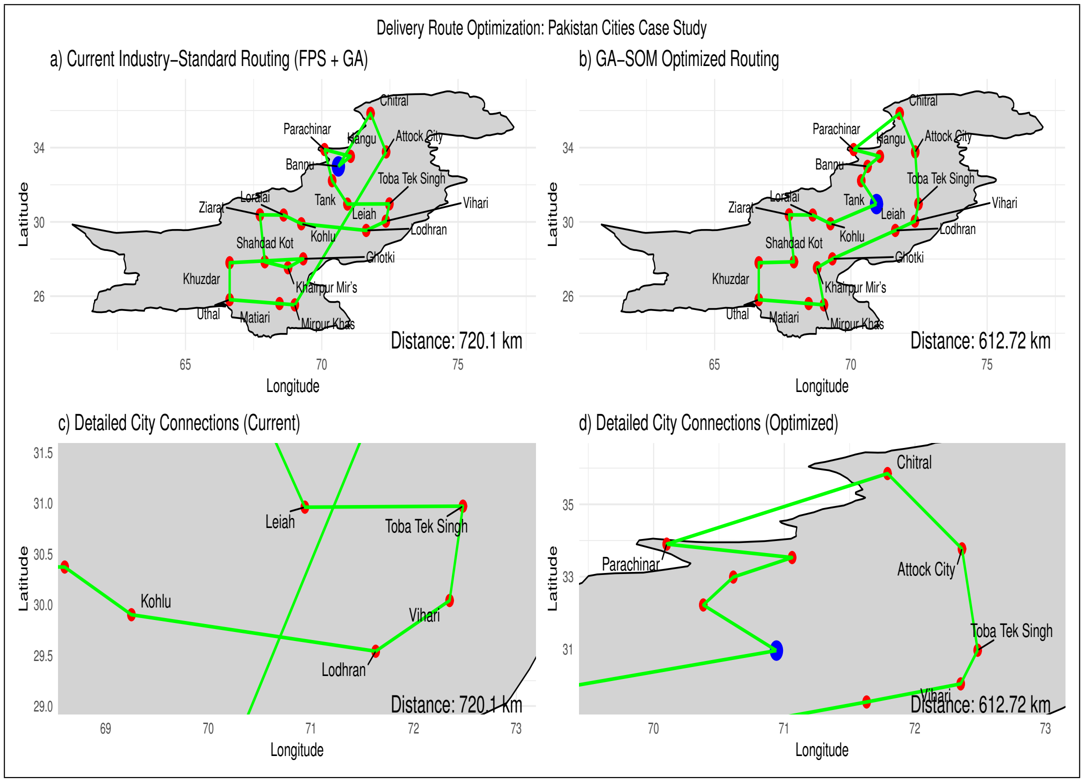

Figure 3: *Delivery route optimization: Current industry-standard routing (750km), GA-SOM optimized routes (613km), and detailed city connections.*

Method	n	Operator	GA				GA-SOM			
			FinalBest	AvgBest	SD	AvgRunTime	FinalBest	AvgBest	SD	AvgRunTime
CX	20	FPS	708.32	1099.62	25.17	0.14	650.04	1035.76	20.17	0.06
		LRS	696.63	1003.81	33.17	0.20	699.19	1013.88	41.59	0.08
		ERS	700.12	1010.71	34.42	0.25	700.72	1020.41	42.10	0.11
		FBS	705.11	1015.23	35.21	0.30	710.07	1030.18	43.19	0.13
		PTS	660.21	680.10	24.37	0.18	615.32	705.12	22.00	0.17
		RRS	619.87	726.33	27.26	0.15	609.18	711.70	24.91	0.10
OX	20	FPS	691.02	870.52	24.12	0.21	690.08	817.13	18.17	0.19
		LRS	692.63	830.93	29.81	0.17	674.00	814.02	24.67	0.21
		ERS	685.12	790.35	30.21	0.25	702.32	877.41	25.14	0.17
		FBS	719.41	895.13	31.05	0.27	705.00	870.52	26.71	0.24
		PTS	640.32	720.06	23.30	0.16	670.62	690.41	20.61	0.15
		RRS	629.06	718.73	24.86	0.14	612.89	706.03	26.92	0.09

Table 4: *Performance comparison of existing selection operators and RRS method using CX and OX crossover and EM mutation operators for 20 Pakistan' cities.*

7. Summary

This study meticulously compares the performance of a novel RRS operator against current selection methods, employing diverse crossover techniques and problem sizes. The RRS operator consistently surpassed existing operators across all evaluated key metrics, including solution quality, stability, and computational efficiency. When integrated with the GA-SOM framework RRS+GA-SOM, its performance exceeded that of RRS+GA, yielding more effective and practical solutions across all tested scenarios. Among the assessed combinations, RRS (GA-SOM + CX) demonstrated superior efficacy. This combination consistently produced the lowest SD values, achieved optimal FinalBest and AvgBest results, and maintained commendably competitive average run times. This particular configuration proves especially advantageous for extensive optimization challenges where stability and solution quality are paramount. Furthermore, our RRS (GA + OX) exhibited exceptional performance in specific medium-sized problems, thereby reinforcing the adaptability and robustness of the RRS operator across various optimization scenarios.

Experimental results obtained from the Pakistan city coordinates dataset further underscore the advantages of the RRS operator in terms of solution quality, stability, and computational economy. The hybrid GA-SOM framework significantly amplifies these benefits, particularly for extensive optimization techniques. Among all combinations, RRS + CX (GA-SOM) proved to be the most efficacious, consistently yielding the lowest FinalBest and AvgBest values, demonstrating the most steady convergence (minimal SD), and achieving notably competitive run times. These findings collectively illustrate the practical utility of the RRS operator and the hybrid GA-SOM framework for real-world logistics and routing challenges. Our composite logistics case study indicates annual savings of $4.53 million for a medium-sized operator, thereby substantiating its direct utility in developing economies where fuel expenses are a primary driver of operational costs.

This study is limited to symmetric TSP; future work should assess performance on asymmetric and constrained vehicle routing problems. While RRS shows superior performance, its parameters require tuning, suggesting a need for adaptive control mechanisms. Scalability to ultra-large networks (>10,000 nodes) also warrants investigation through parallelization.

Policymakers should incentive logistics firms to adopt AI driven optimization through tax benefits or other grants. Establishing a national digital logistics platform could democratize access to these tools. Furthermore, integrating fuel savings and emission reductions into transportation policies would align economic and environmental objectives.

Data accessibility and Online Appendix

In the Online Appendix, containing detailed results for all algorithmic pseudocode, parameter setup and all large comprehensive tables is available at `https://doi.org/10.5281/zenodo.17509716`.

The primary dataset used to validate the conclusions presented in this manuscript was obtained from the well-established TSPLIB repository, a publicly available resource widely recognized for its collection of traveling salesman problem instances. This repository can be accessed at `https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95`.

The findings presented in this manuscript are supported by data obtained from the publicly available Kaggle platform through following link: `https://www.kaggle.com/datasets/tayyarhussain/pakistani-cities-latitude-andlongitude-dataset?resource=download`.

State Bank of Pakistan financial Stability Review retrieved from `https://www.sbp.org.pk/fsr`.

Pakistan logistics performance assessment retrieved from `https://www.worldbank.org`.

Tranzum Courier Service (TCS) company overview can accessed at `https://www.tcsexpress.com`.

Pakistan Automotive Manufacturers Association commercial vehicle fuel efficiency report assessed at `https://www.pama.org.pk/publications/Table7:Avg.7.5-8.5km/literformediumtrucks`.

Oil & Gas regulatory authority fuel Price can accessed at `https://www.ogra.org.pk`.

References

[1] Abualigah, L., Elaziz, M. A., Khasawneh, A. M., Alshinwan, M., Ibrahim, R. A., Al-qaness, M. A. A., Mirjalili, S., Sumari, P., and Gandomi, A. H. (2022). Meta-heuristic optimization algorithms for solving real-world mechanical engineering design problems: a comprehensive survey, applications, comparative analysis, and results. Neural Computing and Applications, 34, 4081–4110. doi: 10.1007/s00521-021-06747-4

[2] Alanzi, E., Menai, and M. E. B. (2025). Solving the traveling salesman problem with machine learning: a review of recent advances and challenges. Artificial Intelligence Review, 58(9), 267. doi:10.1007/s10462-025-11267-x

[3] Applegate, D. L., Bixby, R. E., Chvátal, V., and Cook, W. J. (2006). *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.

[4] Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. In: *Proceedings of an International Conference on Genetic Algorithms and their Applications*, 101-111. Hillsdale, New Jersey.

[5] Bokhari, S. W. A., Ali, N., and Hussain, A. (2025). Optimizing maritime routes: A multi-method analysis from Shanghai to Vladivostok. Croatian Operational Research Review, 16(1), 73–83. doi:10.17535/crorr.2025.0007

[6] Dorigo, M., Birattari, M., and Stutzle, T. (2016). *Ant Colony Optimization*. IEEE Computational Intelligence Magazine, 1(4), 28 - 39. doi:10.1109/MCI.2006.329691

[7] Ebrahimi, D., Kashefi, S., de Oliveira, T. E. A., and Alzhouri, F. (2024). Efficient routing and charging strategy for electric vehicles considering battery life: A reinforcement learning approach. In: *2024 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 429-435. doi:10.1109/CCECE59415.2024.10667251

[8] Grefenstette, J.J. (1986). Optimization of control parameters for genetic algorithms. IEEE Transactions on Systems, Man and Cybernetics, 16(1), 122-128. doi:10.1109/TSMC.1986.289288

[9] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

[10] Hussain, A., and Muhammad, Y. S. (2020). Trade-off between exploration and exploitation with genetic algorithm using a novel selection operator. Complex & Intelligent Systems, 6(1), 1-14. doi:10.1007/s40747-019-0102-7

[11] Hussain, A. and Cheema, S. A. (2020). A new selection operator for genetic algorithms that balances between premature convergence and population diversity. Croatian Operational Research Review, 11(2020), 107–119. doi:10.17535/crorr.2020.0009

[12] Julstrom, B. A. (1999). It's all the same to me: Revisiting rank-based probabilities and tournaments. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99*, 2, 1501-1505. IEEE. doi:10.1109/CEC.1999.782661

[13] Khurshid, B., Maqsood, S., Khurshid, Y., Naeem, K., and Khalid, Q. S. (2024). A hybridization of evolution strategies with iterated greedy algorithm for no-wait flow shop scheduling problems. Scientific Reports, 14(1), 2376. doi:10.1038/s41598-023-47729-x

[14] Maroof, A., Khalid, Q. S., Mahmood, M., Naeem, K., and Saeed, S. (2023). Vehicle Routing Optimization for Humanitarian Supply Chain: A Systematic Review of Approaches and Solutions. IEEE Access, 11, 127157–127175. doi:10.1109/ACCESS.2023.3331062

[15] Michalewicz, Z., Janikow, C. Z., and Krawczyk, J. B. (1992). A modified genetic algorithm for optimal control problems. Computers & Mathematics with Applications, 23(12), 83–94. doi:10.1016/0898-1221(92)90094-X

[16] Naqvi, F. B. and Shad, M. Y. (2022). Seeking a balance between population diversity and premature convergence for real-coded genetic algorithms with crossover operator. Evolutionary Intelligence, 1-16. doi:10.1007/s12065-021-00636-4

[17] Papa, G. (2021). Applications of dynamic parameter control in evolutionary computation. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1064–1088. doi:10.1145/3449726.3461435

[18] Rafique, F., Cui, H., and Amin, S. (2025). An adaptive multi-strategy selection method for improving genetic algorithm performance. Advances and Applications in Statistics, 92(8), 1105–1142. doi:10.17654/0972361725050

[19] Ruan, Y., Cai, W., and Wang, J. (2024). Combining reinforcement learning algorithm and genetic

algorithm to solve the traveling salesman problem. The Journal of Engineering, 2024(6), e12393. doi:10.1049/tje2.12393

[20] Seyyedabbasi, A., and Kiani, F. (2023). Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. Engineering with Computers, 39(4), 2627–2651. doi:10.1007/s00366-022-01604-x

[21] Shabani, A., Asgarian, B., Salido, M., and Gharebaghi, S. A. (2020). Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems. Expert Systems with Applications, 161, 113698. doi:10.1016/j.eswa.2020.113698

[22] Tang, L., Wang, X., He, Z., Wang, S., and Lin, J. (2024). Review of Traveling Salesman Problem Solution Methods. In Pan, L., Wang, Y., Lin, J. (Eds.), Bio-Inspired Computing: Theories and Applications. BIC-TA 2023. Communications in Computer and Information Science (Vol. 2062, 3-16). Springer, Singapore. doi:10.1007/978-981-97-2275-4_1

[23] Whitley, D. and Sutton, A. M. (2012). Genetic algorithms A survey of models and methods. In G. Rozenberg, T. Back, J. N. Kok (Eds.), Handbook of Natural Computing (Vol. 4, 637–671). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-92910-9_21

[24] Zelinka, I. (2016). SOMA—self-organizing migrating algorithm. Self-Organizing Migrating Algorithm: Methodology and Implementation, 3-49. Springer International Publishing. doi:10.1109/ITIS50118.2020.9321044