Optimizing the Travelling Salesman Problem for Brazil and Burma using Evolutionary Algorithms with Varied Operators and Parameters

Esosa Orumwese

14th November, 2023.

1 Introduction

In this work, the challenge involves solving the well-known Travelling Salesman Problem (TSP) for two distinct datasets: one comprising 58 cities in Brazil and the other with 14 cities in Burma. The goal of TSP is to determine the most cost-effective route for a salesperson to visit all cities exactly once before returning to the starting point. The fitness of each route is computed based on the cumulative travel cost. I aim to implement an evolutionary algorithm to optimize the cost function for the TSP in both the Brazilian and Burmese city datasets [5].

To achieve this, the following objectives will be pursued: exploring the impact of varying population size, tournament size, crossover, mutation, and replacement operators on the evolutionary algorithm's outcomes; identifying the most effective parameters for the EA in solving the TSP and minimizing the cost function to derive the most optimal route with the lowest fitness (cost) for both datasets [3].

2 Methodology

To tackle the TSP, candidate solutions were represented in a k-array, allowing each cell to represent a city with a constraint to avoid repetition. Experiment designs encompassed varied population sizes, tournament sizes, crossover, mutation, and replacement operators (refer to Table 1) [3]. Using Python's xml.etree.ElementTree module, XML data for both countries was parsed. The cost data was stored as square matrices with rows corresponding to the number of cities (58 for Brazil, 14 for Burma). A total of 144 experiments were conducted to test the effect of each parameter/operator, each with 10,000 iterations, concluded at the 10,000th fitness evaluation. Random seeds (see Table 1) facilitated 10 trials to address EA's stochastic nature, assess performance variability, and ensure statistical significance.

For each trial and experiment, I tracked the best fitness per iteration, the top fitness solution, and the execution time (wall-clock time). After 10 trials, I calculated the average and standard deviation for these recorded values. To isolate the impact of tournament sizes, mutation operators, and replacement operators on the mean best fitness, I maintained consistent crossover types and population sizes. This facilitated the identification of the most effective combination resulting in the mean best fitness as seen in Table 2.

3 Results and Analysis

This study optimized the Travelling Salesman Problem (TSP) for Brazil and Burma datasets using evolutionary algorithms. Across 10 trials, 144 parameter and operator variations were examined. Table 2 summarizes the best fitness for each crossover-type-population-size combination.

For Brazil, one instance led to the best fitness with a single unique solution. In contrast, Burma had 700 instances with 111 unique solutions. Appendix A outlines the parameters for the best solution in each TSP problem.

Table 1: Operators and parameters used[2].

S/N	Operator/Parameter	Type/Values
1.	Selection method	Tournament selection
2.	Crossover operators	Single point crossover, Edge-3 crossover
3.	Mutation operators	Swap mutation, Insert mutation, Inversion mutation, Scramble mutation
4.	Replacement operators	Replace weakest, Replace 1st weakest
5.	Population sizes	50, 100, 200
6.	Tournament sizes	5, 10, 20
7.	Number of iterations	10,000
8.	Number of random trials	10
9.	Encoding	k-array of length = to number of cities
10.	Crossover rate	1
11.	Mutation rate	1
12.	Termination criterion	After the 10000th fitness evaluation
13.	Seeds	37, 235, 908, 72, 767, 905, 715, 645, 847, 960

Table 2: Best MBF for each CRSV-Pop size Combination

TSP	**Experiment**	**Pop size**	**Tourn size**	**CRSV func**	**Mut func**	**Repl. func**	**Mean Best Fitness**	**MBF Marg.Err**	**Mean Runtime (secs)**
Brazil58	Edge50	50	10	Edge-3	Inversion	Weakest	25945.2	±372.03	31.73
	Edge100	100	20	Edge-3	Inversion	Weakest	26282.2	±405.35	32.86
	Edge200	200	20	Edge-3	Inversion	Weakest	26234.8	±354.96	33.29
	SinglePoint50	50	5	SinglePt	Swap	Weakest	46592.4	±4244.71	3.65
	SinglePoint100	100	10	SinglePt	Swap	Weakest	43066.8	±2502.72	3.95
	SinglePoint200	200	20	SinglePt	Swap	Weakest	44196.5	±2535.19	4.71
Burma14	Edge50	50	5	Edge-3	Insert	1st Weakest	3323.0	±0.00	5.44
	Edge100	100	5	Edge-3	Insert	1st Weakest	3323.0	±0.00	5.63
	Edge200	200	5	Edge-3	Insert	1st Weakest	3323.0	±0.00	5.92
	SinglePoint50	50	5	SinglePt	Swap	1st Weakest	3355.5	±35.28	2.82
	SinglePoint100	100	5	SinglePt	Swap	1st Weakest	3328.2	±4.80	3.00
	SinglePoint200	200	5	SinglePt	Scramble	1st Weakest	3359.1	±50.17	3.54

4 Convergence

The convergence curves plotted in Figure 1 represent only the experiments outlined in Table 2. To highlight the differences between each curve, the mean of the log10 of the best fitness per iteration for each trial was plotted against the number of iterations.

In the Brazil dataset, noticeable distinctions emerge between two distinct groups of curves. Those employing the edge crossover demonstrated lower performance, converging around the 900th iteration. However, among these, only Edge50 appeared to achieve convergence, while the others displayed potential for further improvement given more time. Conversely, the single-point crossover experiments, particularly SinglePt50, showed initial convergence around the 800th iteration, yet slower improvement led it to be the least effective by the termination criterion. For the Burma dataset, all algorithms performed reasonably well. The edge crossover, especially Edge50, displayed the steepest dip and earliest convergence, while other edge experiments reached a similar fitness level near the 200th iteration. Among the SinglePoint experiments, SinglePoint100 stood out, demonstrating improved performance and convergence around the 800th iteration, aligning with the Edge experiments' mean best fitness.

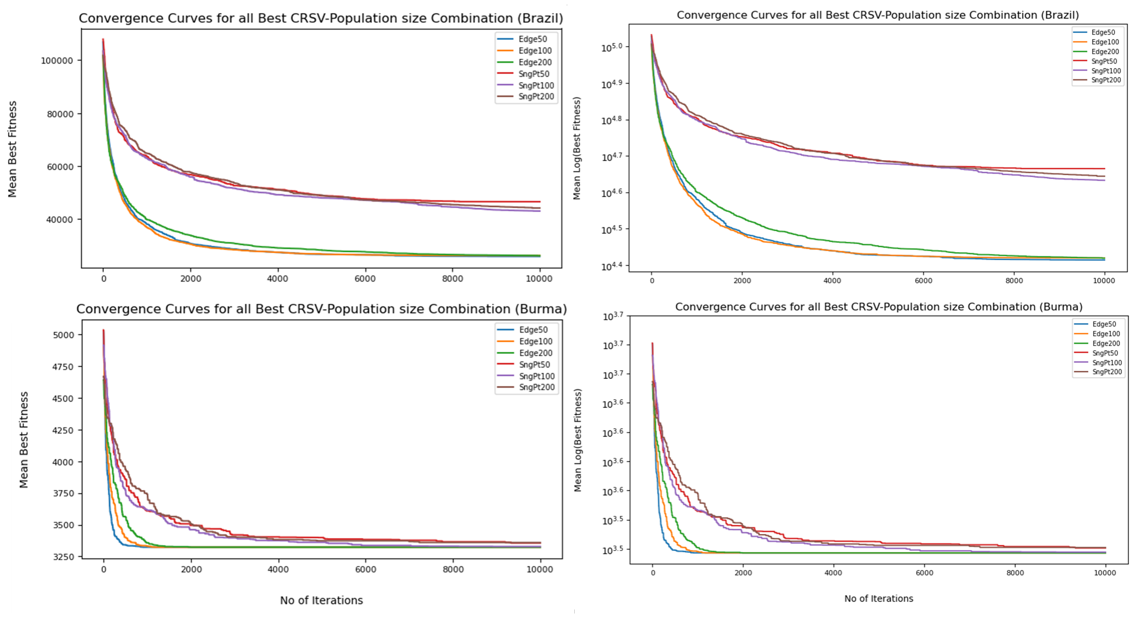

Figure 1: Convergence curves for best experiments involving CRSV=Pop size combination for Brazil and Burma

5 Analysis

5.1 Which combinations of parameters produced the best results?

From Table 2, it's evident that in Brazil, a population size of 50 with a tournament size of 10, an edge crossover, inversion mutation, and 'replace weakest parent' produced an MBF of 25,945.2 ($\pm$372.03) and a runtime of 31.73 secs. In Burma, population sizes 50, 100, or 200 with a tournament size of 5, an edge crossover, insert mutation, and 'replace 1st weakest parent' led to an MBF of 3323.0 ($\pm$0.00). Notably, a population size of 50 facilitated quicker convergence, evident in Figure 1.

5.2 What do you think is the reason for your findings in Q1

The selection pressure, ranging between 0.1 to 0.2 for Brazil and 0.025 to 0.2 for Burma, balanced exploration and exploitation. The preference for edge-3 over single-point crossover was observed in both cases. Edge crossover allows exploration while preserving parental information, unlike single-point crossover. In Brazil, inversion with edge crossover navigates a larger solution space effectively, capturing optimal routes. Meanwhile, insert mutation with edge crossover in Burma efficiently explores and exploits shorter routes. For Brazil, 'replace weakest' with edge crossover suits larger populations robustly. Conversely, in Burma, 'replace 1st weakest' with edge crossover efficiently converges in smaller populations, exploiting top performers [2].

5.3 How do each of the parameter settings influence the performance of the algorithm?

Heatmaps (Figure 2) were used to gauge parameter influences on algorithm performance and it is based on The Normalized Mean Best Fitness (NMBF) which is calculated as $\frac{\text{MBF}-\text{Min MBF}}{\text{Range}}$. In Brazil, single-point crossover with insert and inversion mutations resulted in consistently poor performances, exhibiting high normalized mean best fitness values across population sizes 50 and 100. However, edge crossover generally demonstrated good performance, showing lower normalized mean best fitness values ranging from 0.00 to 0.12, except for when it was combined with scramble mutation, particularly with population size 50 and tournament size 20. In Burma, single-point crossover with insert and inversion mutations also performed sub-optimally, displaying high normalized mean best fitness values. Conversely, edge crossover showcased overall good performance, revealing lower normalized mean best fitness values ranging from 0.00 to 0.03. Notably, insert mutation yielded the best results across various parameter combinations [2][3].

5.4 Can you think of a local heuristic function to add?

To enhance the results, incorporating the Lin-Kernighan algorithm proves advantageous. This algorithm, categorized as a local search method, focuses on executing k-opt moves within tours. Specifically designed to exploit the current solution's neighborhood, it systematically introduces incremental modifications. Operating on a feasible tour, the algorithm continually executes k-opt moves by substituting k edges within the solution, aiming to attain an improved solution state. The process terminates upon reaching a point where no superior solution can be found [4].

5.5 Can you think of any variation for this algorithm to improve your results?

Following the suggestion by Yan, Xue-song, et al.,[7] enhancing convergence speed for smaller populations could involve implementing the inver-over crossover operator and employing a dynamically adaptive mutation rate. The inver-over crossover operator integrates insights gleaned from other individuals within the population, resulting in the production of high-quality solutions. Essentially, this operator combines the principles of inversion mutation, applied to a single individual, and a recombination operator, with the selection and size of the inverted segment contingent on information from other individuals within the population [6].

5.6 Can you think of other nature inspired algorithms that might have provided better results?

To enhance performance in addressing the TSP problem, considering the utilization of the CGAS (Cooperative Genetic Ant Systems) hybrid algorithm, proposed by Dong, Guo, & Tickle, (2012), appears promising. This approach combines genetic algorithms (GA) and ant colony optimization (ACO), leveraging the strengths of both methods to enhance ACO's performance in solving TSPs. The strategy involves information exchange between the ant system (AS) and GA at the conclusion of each iteration. This cooperative interaction facilitates the selection of superior solutions for subsequent iterations and fosters an improved potential for attaining the global optimal solution. Furthermore, the independent operation of GA sustains a high diversity level within the new generation of solutions, augmenting their suitability for further processing. Notably, their method demonstrated consistent success in exploring global optimal solutions for TSPs, especially in scenarios involving smaller TSPs [1].

6 Conclusion

From this research, we can see that for the TSP problem, having a crossover that explores the landscape of the solution space effectively while also keeping the information of the parents seen such as the optimal routes seen, helps a great deal in improving the fitness of the algorithm. Also, for convergence, when the algorithms were paired with mutation operators that took smaller steps in the neighbourhood of the solutions, it allowed the algorithm to exploit that neighbourhood eventually leading to finding solutions with better fitness. When paired with these kinds of mutation and crossover operators, having a large population size was not necessary as a population of size 50 and 100 paired with a selection pressure of 0.05 to 0.1 resulted in better results at a faster convergence speed.

7 Appendix A: Best Solutions Table

Table 3: Showing the best solution seen for all instances along with the operators/parameters that led to it. While Brazil had one instance, Burma had 700 with 111 unique solutions.

TSP	Seed	Pop size	Tourn size	CRSV func	Mut func	Repl func	Solution	Fitness	Runtime (secs)
Brazil58	960	200	20	Edge-3	Inversion	Weakest	[52, 49, 3, 7, 21, 15, 37, 41, 6, 30, 10, 38, 20, 35, 16, 25, 18, 5, 27, 13, 36, 14, 33, 45, 55, 44, 32, 28, 2, 47, 54, 53, 1, 40, 34, 9, 51, 50, 46, 48, 42, 26, 4, 22, 11, 56, 23, 57, 43, 17, 0, 29, 39, 12, 8, 24, 31, 19]	25435	33.09
Burma	37	50	5	SinglePt	Swap	1st Weakest	[10, 8, 9, 12, 1, 13, 2, 3, 4, 5, 11, 6, 0, 7]	3323	2.95

8 Appendix B: Heatmap Analysis

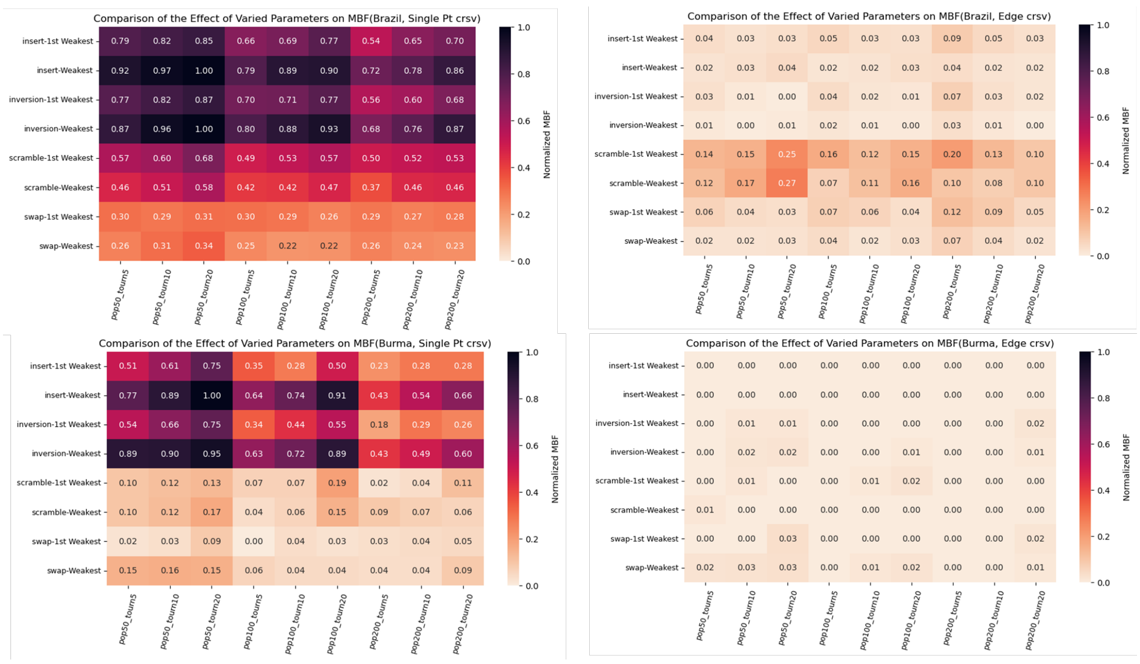

Figure 2: Heatmap showing the effect of all the combination of parameters/operators on Normalized Mean Best Fitness for both Brazil and Burma. Note, MBF was normalized by subtract the mininum MBF seen and dividing by the range.

9 Appendix C: Fitness Comparison Graph

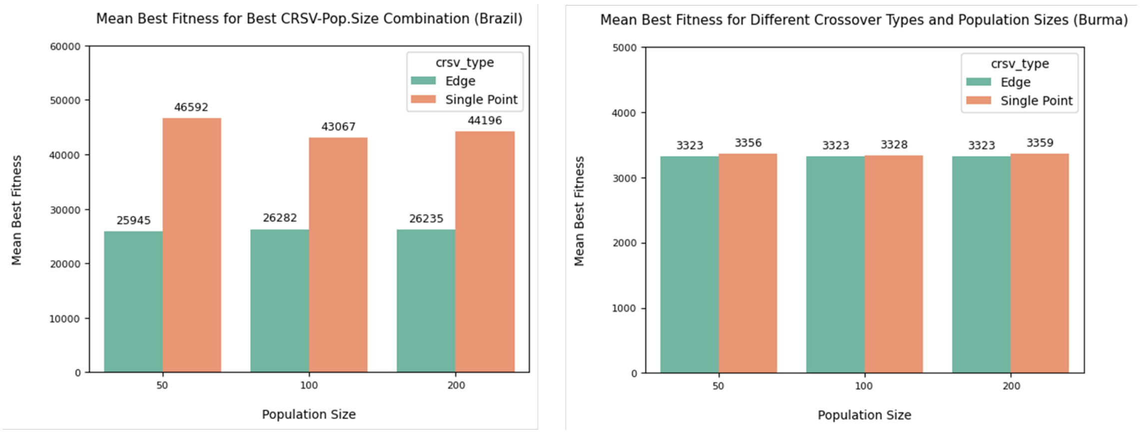

Figure 3: Barchart showing the Mean Best Fitness achieved by each crossover-type-population-size combination for Brazil and Burma.

References

[1] G. Dong, W.W. Guo, and K. Tickle. Solving the traveling salesman problem using cooperative genetic ant systems. *Expert systems with applications*, 39(5):5006–5011, 2012.

[2] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag Berlin Heidelberg, 2015.

[3] A Helal. Search landscapes and hill climbing. Lecture notes for Nature Inspired Computation, ECMM409, 2023. University of Exeter.

[4] Keld Helsgaun. General k-opt submoves for the lin–kernighan tsp heuristic. *Mathematical Programming Computation*, 1:119–163, 2009.

[5] F. Jolai and A. Ghanbari. Integrating data transformation techniques with hopfield neural networks for solving travelling salesman problem. *Expert Systems with Applications*, 37(7):5331–5335, 2010.

[6] G. Tao and Z. Michalewicz. Inver-over operator for the tsp. In *International Conference on Parallel Problem Solving from Nature*, pages 803–812, Berlin, Heidelberg, September 1998. Springer Berlin Heidelberg.

[7] Xue-Song Yan, Hui-Ming Liu, Jie Yan, and QH Wu. A fast evolutionary algorithm for traveling salesman problem. In *Third International Conference on Natural Computation (ICNC 2007)*, volume 4, pages 85–90. IEEE, August 2007.