Implementation Work: Construtive Heuristics

Nicolas Von Dolinger Moreira Rocha
2022035571
Federal University of Minas Gerais - UFMG

September 5, 2025

Abstract

The Traveling Salesperson Problem (TSP) is a fundamental NP-hard challenge in combinatorial optimization, requiring the determination of the shortest possible route that visits a set of cities and returns to the origin. This work focuses on the implementation and performance analysis of a constructive heuristic to generate high-quality approximate solutions for the TSP. The chosen method is the Clarke-Wright Savings algorithm, an approach traditionally applied to the Vehicle Routing Problem (VRP). A key adaptation was made for its application to the TSP: the heuristic is executed iteratively, with each vertex in the graph serving as a potential depot, and the best tour found across all iterations is selected as the final solution.

1 Algorithm

The constructive heuristic chosen for this work is the **Clarke-Wright Savings Algorithm**. While originally designed for the Vehicle Routing Problem (VRP), it is a powerful method for generating high-quality initial solutions for the Traveling Salesperson Problem (TSP) as well.

The core principle of the algorithm is to iteratively merge routes to achieve the greatest possible cost reduction. The process begins with a trivial solution where each city is on its own separate round-trip route from a designated central depot. The algorithm then calculates the potential "saving" for every pair of cities (i, j) that are not yet on the same route. This saving, denoted as s_{ij}, represents the distance saved by linking cities i and j directly and eliminating their individual trips back to the depot.

The saving is calculated using the formula:

$$s_{ij} = d_{0i} + d_{0j} - d_{ij}$$

where d_{0i} is the distance from the depot (node 0) to city i, d_{0j} is the distance from the depot to city j, and d_{ij} is the distance between city i and city j.

After calculating all possible savings, they are sorted in descending order. The algorithm then processes this sorted list, greedily merging the routes of the two cities that offer the highest saving, provided the merge is valid. A merge is considered valid if it does not create a premature cycle (a subtour that does not include the depot) and if it connects two nodes that are currently endpoints of their respective partial tours. This merging process continues until all cities are connected in a single tour.

Since the standard TSP does not have a predefined depot, the algorithm is adapted by iterating through every vertex in the graph, treating each one as a potential depot. The complete heuristic is run for each case, and the final solution reported is the tour with the minimum total cost found across all iterations. This ensures a more thorough exploration of the solution space.

2 Results

The performance of the adapted Clarke-Wright Savings heuristic was evaluated across 21 standard instances from the TSPLIB. The results, detailed in Figure 1, provide a comprehensive overview of the algorithm's effectiveness in terms of solution quality and computational time.

Figure 1 presents a direct comparison between the tour costs obtained by our heuristic and the known optimal values. While the heuristic consistently finds solutions reasonably close to the optimal ones, it is important to note outliers such as the `att48` instance, where the obtained result was significantly distant from the optimum. This highlights that the heuristic alone does not guarantee a near-optimal result for every type of instance, even for those with a relatively small number of vertices.

To quantify this proximity more broadly, Figure 2 illustrates the percentage gap, which is the relative difference between the obtained and optimal costs. The analysis shows that while the gap is generally low, specific instances can yield less favorable outcomes. Finally, Figure 3 displays the execution time required for each instance. A clear trend emerges, demonstrating that the computational cost grows considerably with the increase in input size, a direct consequence of the heuristic's iterative depot selection process.

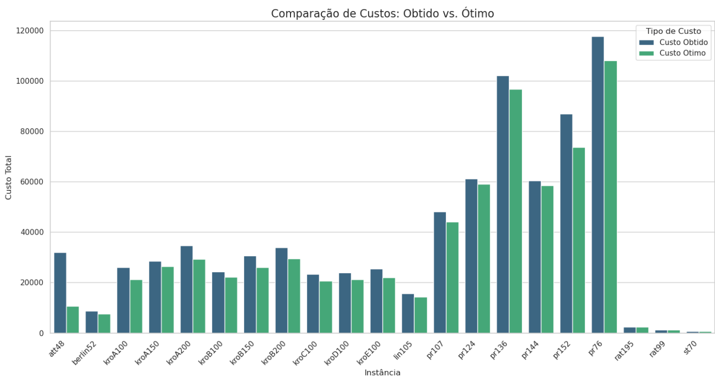

Figure 1: Comparison between the obtained tour cost (blue) and the optimal cost (green) for each instance.

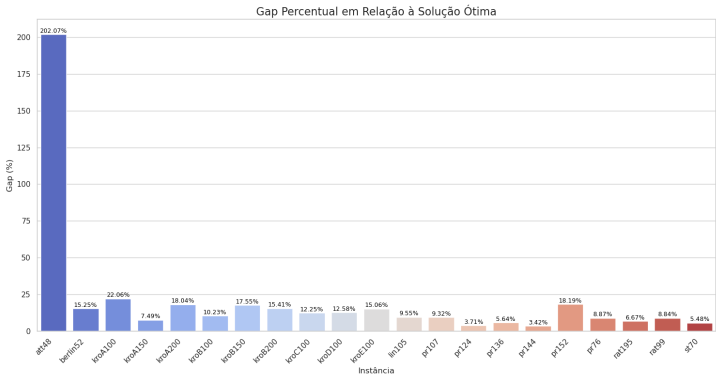

Figure 2: Percentage gap of the obtained solution relative to the optimal solution.

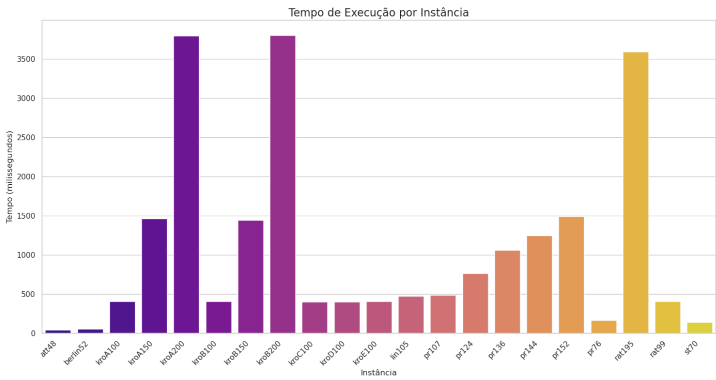

Figure 3: Total execution time in milliseconds (ms) for each instance, including the iterations for all possible depots.

3 Conclusion

In this work, an adapted version of the Clarke-Wright Savings heuristic was successfully implemented and evaluated for the Traveling Salesperson Problem. By systematically iterating through each vertex as a potential depot, the algorithm demonstrated its ability to consistently produce high-quality solutions across a range of standard TSPLIB instances.

The experimental results confirmed that the heuristic achieves a low average percentage gap relative to the known optimal solutions, establishing it as an effective and efficient constructive method. The tours generated by this approach provide a strong foundation for subsequent improvement phases, serving as excellent initial solutions for local search algorithms or metaheuristics to further refine toward optimality.

References

[1] GEEKSFORGEEKS. *Introduction to Disjoint Set Data Structure or Union Find Algorithm*. Disponível em: https://www.geeksforgeeks.org/dsa/introduction-to-disjoint-set-data-structure-or-union-find-algorithm/. Acesso em: 5 set. 2025.

[2] REINELT, G. *TSPLIB95*. Heidelberg University, 1995. Disponível em: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf. Acesso em: 5 set. 2025.

[3] VONDOLINGER, N. *algorithms_2_UFMG*. GitHub repository, 2024. Disponível em: https://github.com/nicolasvondolinger/algorithms_2_UFMG. Acesso em: 5 set. 2025.