MoTSPPP: Multi-objective Traveling Salesman Problem with Profits and Passengers

Juvenal Bruno Andrade da Silva[1] **(Author)**
Islame Felipe da Costa Fernandes[1] **(Advisor)**

[1]Programa de Pós-Graduação em Ciência da Computação (PGCOMP)
Institute of Computing, Federal University of Bahia (UFBA)
Salvador – BA – Brasil

juvenal.bruno@ufba.br, islame.felipe@ufba.br

Abstract. *Ridesharing systems have emerged as a promising solution to urban mobility challenges, requiring routing algorithms that handle conflicting objectives such as travel cost, travel time, and driver bonuses. Previous studies addressed this context through extensions of the Traveling Salesman Problem, notably the Traveling Salesman Problem with Profits (TSPP) and the Bi-objective Traveling Salesman Problem (BiTSP), which consider subsets of these objectives. However, the literature lacks a multi-objective formulation that simultaneously optimizes cost, time, and bonuses in ridesharing systems. This work proposes the Multi-objective Traveling Salesman Problem with Profits and Passengers (MoTSPPP), an NP-hard optimization problem that minimizes travel cost and time while maximizing bonus collection. Eight algorithms are investigated, including an exact method based on a mathematical formulation, three naive heuristics, and four state-of-the-art evolutionary metaheuristics (NSGA-II, MOEA/D, IBEA, and SPEA2). An experimental study on 252 symmetric and asymmetric instances with varying edge-weight correlations evaluates processing time, solution quality, and diversity using statistical analysis. Results indicate that MoTSPPP is computationally more challenging than TSPP and BiTSP, and that metaheuristics significantly outperform naive heuristics.*

1. Introduction

Ridesharing systems have become a strategic component of urban mobility, enabling drivers and passengers to share transportation resources while reducing operational costs [Agatz et al. 2012]. In emerging digital-platform ecosystems, however, ridesharing increasingly coexists with service-based activities [Silva et al. 2020]. A representative real-world scenario involves a professional registered on two platforms: one for providing local services (e.g., technical inspections, equipment repairs, or quality audits) and another for transporting passengers or goods. The professional must establish a route to perform services at selected locations, starting and ending at a depot. Each completed service generates a non-monetary reward, such as service evaluation, platform credits, reputation score, or eligibility for future assignments. This paper refers to these rewards as *bonus*.

During the execution of this route, the professional travels with available vehicle capacity, enabling passengers to be transported on compatible segments to reduce travel expenses through cost sharing. The travel cost of each edge is divided among all occupants. However, bonus collection requires additional time at visited vertices, which may cause passengers to exceed their maximum travel time or budget constraints. Consequently, maximizing bonus collection may restrict passenger feasibility, while prioritizing passenger inclusion may limit bonus acquisition or increase travel cost. The conflicts among route cost, route time, and bonus collection motivated the development of a multi-objective optimization formulation.

The current literature does not adequately capture this real-life situation. Prior studies have modeled ridesharing systems through single-objective *TSP* extensions [Marques et al. 2019, Silva et al. 2020], typically enforcing minimum-bonus quotas under cost minimization. In contrast, multi-objective variants such as the *TSPP* [Feillet et al. 2005] and the *BiTSP* [Yue et al. 2026] consider trade-offs between two objectives but do not incorporate passenger-dependent constraints. Although numerous *TSP* variants have been proposed [Dell'Amico et al. 1995, Tsiligirides 1984, Awerbuch et al. 1995, Balas 1989], to the best of our knowledge, no existing formulation simultaneously integrates routing decisions, passenger-specific feasibility constraints, and bonus collection within a unified multi-objective optimization framework.

To address this gap, this dissertation proposes the *Multi-objective Traveling Salesman Problem with Profits and Passengers* (*MoTSPPP*), a novel formulation designed to model real-life bonus-related ridesharing systems. The *MoTSPPP* is an NP-hard tri-objective problem that simultaneously minimizes travel cost and travel time while maximizing collected bonuses. In this formulation, vertices are associated with bonus values and corresponding service times, and edges are characterized by travel cost and travel time. The driver visits a subset of vertices exactly once, transports passengers along the route, and may collect bonuses at selected locations. This dissertation proposes the first multi-objective mixed-integer linear programming (MILP) formulation for the *MoTSPPP*. In addition, eight algorithmic approaches are investigated: an exact solver based on the proposed MILP model, three naive heuristics, and four evolutionary metaheuristics.

These algorithms were evaluated on 252 benchmark instances, including symmetric and asymmetric graphs with distinct edge-weight correlation patterns. The results, supported by statistical tests, demonstrated that the *MoTSPPP* is computationally more challenging than the *TSPP* and the *BiTSP*, and highlighted the high solution quality achieved by the metaheuristics. Source code and results are available at `https://github.com/juvenalbruno/MoTSPPP`.

In summary, this dissertation advances Computer Science through the following contributions:

- **Theoretical contribution:** Introduction of a novel multi-objective optimization problem that intrinsically combines routing decisions, passenger feasibility, bonus collection, and tri-objective dominance relations.
- **Application novelty:** Modeling realistic digital-platform scenarios in which service providers simultaneously perform location-based tasks and transport passengers, offering decision-support insights for next-generation ridesharing systems.
- **Methodological contribution:** Development of the first multi-objective MILP formulation for the *MoTSPPP*, including linearization strategies and computational complexity analysis.
- **Algorithmic innovation:** Design and systematic investigation of eight algorithms, including exact methods, constructive heuristics, and evolutionary me-

taheuristics. They are the first solution approaches proposed for the *MoTSPPP*. The dissertation also introduced problem-specific evolutionary operators to handle the problem correctly.

- **Experimental insight:** First large-scale, statistically rigorous evaluation revealing the increased computational complexity of the *MoTSPPP* compared to *TSPP* and *BiTSP*, and demonstrating the effectiveness of tailored metaheuristics.

Collectively, these contributions expand the multi-objective literature and strengthen its applicability to complex real-world routing systems. Part of this research was published at the *Genetic and Evolutionary Computation Conference* (*GECCO*) (Qualis A1) [Silva et al. 2025], the leading international conference on evolutionary computation. The full dissertation text is available at `https://repositorio.ufba.br/handle/ri/43842`.

2. Related work

Table 1 summarizes representative classes of single-vehicle *TSP* extensions related to the *MoTSPPP*, highlighting their main characteristics and distinctions. The differences between these problems and the *MoTSPPP* lie in the objective functions, the bonus collection mechanism, the presence of bound or quota constraints, and the incorporation of passenger-dependent feasibility.

Tabela 1. Summary of related problems.

Problems	Single- or multi-objective	Objective functions			Bonus collection		Bound constraints		Passengers
		Route cost	Route time	Total bonus	Penalty	Collection time	Cost	Bonus	
PTP	Single	✓		✓	✓				
OP	Single			✓	✓		✓		
QTSP	Single	✓						✓	
PCTSP	Single	✓			✓			✓	
QTSPPC	Single	✓				✓		✓	✓
QTSPPIC	Single	✓				✓		✓	✓
BiTSP	Multi	✓	✓						
TSPP	Multi	✓		✓					
MoTSPPP	Multi	✓	✓	✓		✓			✓

Among single-objective formulations, the *Profitable Tour Problem* (*PTP*) [Dell'Amico et al. 1995] minimizes the difference between route cost and collected bonuses, while the *Orienteering Problem* (*OP*) [Tsiligirides 1984] maximizes collected bonuses subject to an upper bound on route cost. Although several *PTP* and *OP* variants have been proposed in the literature, they remain single-objective and do not model passengers [Gunawan et al. 2016].

Similarly, the *Quota Traveling Salesman Problem* (*QTSP*) [Awerbuch et al. 1995] minimizes route cost subject to a minimum bonus quota, and the *Prize Collecting Traveling Salesman Problem* (*PCTSP*) [Balas 1989] adds penalties for unvisited vertices. Two extensions of the *QTSP*, the *QTSPPC* [Marques et al. 2019] and the *QTSPPIC* [Silva et al. 2020], are the only formulations that incorporate passengers by modeling a driver who shares travel costs with them. However, both remain single-objective, failing to capture the conflicts among route cost, time, and bonus collection.

Classical multi-objective extensions of the TSP consider multiple objectives but still do not incorporate passenger-related constraints. The *BiTSP* [Yue et al. 2026] minimizes route cost and travel time, whereas the *TSPP* [Feillet et al. 2005] minimizes route cost while maximizing bonus collection. Although these problems have been extensively studied, no existing formulation incorporates passenger constraints [Cai et al. 2023].

The literature reveals a clear gap. Existing single-objective formulations model bonuses through quota constraints, without explicitly representing trade-offs among conflicting objectives. Conversely, multi-objective TSP variants address cost, bonus, or time trade-offs but disregard passenger-dependent feasibility and cost-sharing effects. By jointly integrating these elements, *MoTSPPP* introduces structural interdependencies between feasibility and objective dominance that are absent in prior models. Therefore, this dissertation fills an important gap in the literature by capturing a more expressive multi-objective formulation for realistic ridesharing systems.

3. Problem definition

The *MoTSPPP* is defined on a graph $G = (V, E)$, where $V = \{1, \ldots, n\}$ is the set of vertices and $E = \{(i, j) \mid i, j \in V\}$ the set of edges. Each edge $(i, j) \in E$ has travel cost c_{ij} and travel time y_{ij}. Each vertex $i \in V$ is associated with a non-negative bonus b_i and a collection time g_i. The salesman visits a subset of vertices exactly once and may collect a bonus at each visited vertex, incurring an additional time g_i.

The vehicle accommodates up to R passengers in addition to the driver, allowing a set of passengers to be transported along the route. Each passenger ξ specifies an origin o_ξ, a destination d_ξ, a maximum fare w_ξ, and a maximum ride time t_ξ. The ride cost of each passenger is computed as the sum of the edge costs along the subpath from o_ξ to d_ξ, where each edge cost is equally divided among all vehicle occupants, including the driver. The passenger's ride time corresponds to the traversal time of this subpath plus the bonus collection time at intermediate vertices. The request of passenger ξ is feasible if the capacity R is respected and if the ride cost and time do not exceed w_ξ and t_ξ, respectively.

A feasible solution consists of a route that starts and ends at the depot, subject to vehicle capacity and passenger feasibility constraints. The *MoTSPPP* simultaneously minimizes total travel cost (f_1), defined as the sum of shared edge costs along the route, minimizes total travel time (f_2), comprising edge traversal times plus bonus collection times, and maximizes total collected bonuses (f_3).

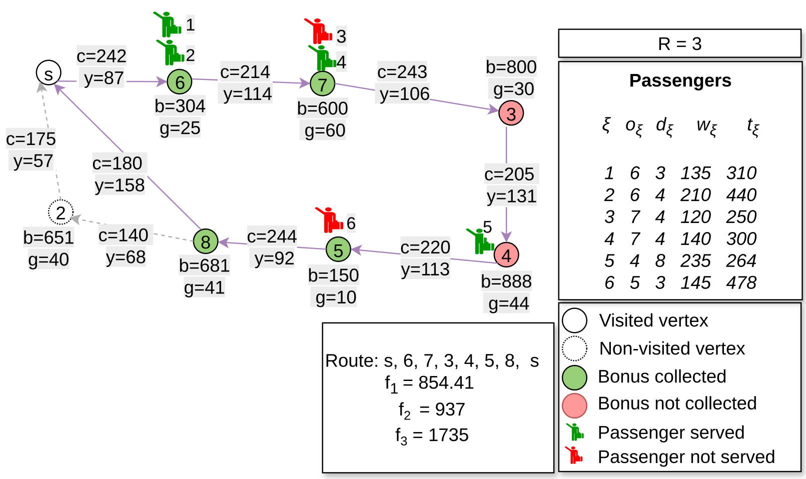

Figura 1. A feasible solution for the *MoTSPPP*

Figure 1 illustrates a feasible solution of the *MoTSPPP*. The route starts and ends at the depot s, following the sequence $s \rightarrow 6 \rightarrow 7 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow s$. Bonuses are collected at vertices 6, 7, 5, and 8, adding their respective collection times g_i to the total travel time. The vehicle capacity is $R = 3$. Among the six passenger requests, passengers $\xi \in \{1, 2, 4, 5\}$ are accepted, while $\xi \in \{3, 6\}$ are rejected. Passengers $\xi = 1$ and $\xi = 2$ board at vertex 6. At vertex 7, only passenger $\xi = 4$ is accepted because serving $\xi = 3$ would violate the maximum fare constraint w_3. Moreover, collecting the bonus at vertex 3 is skipped to ensure that passenger $\xi = 4$'s maximum ride time t_4 is not exceeded. Passenger $\xi = 5$ boards at vertex 4, whereas passenger $\xi = 6$ is rejected because its destination vertex has already been visited.

4. Algorithms

This dissertation proposed eight algorithms for solving the *MoTSPPP*: one exact method, three naive heuristics, and four evolutionary metaheuristics. The exact approach, referred to as *Solver*, is based on the proposed MILP formulation and Gurobi. The naive heuristics (*NH1*, *NH2*, and *NH3*) decompose the problem into sequential phases, independently addressing routing, bonus collection, and passenger assignment. *NH1* and *NH2* first compute the Pareto-optimal set of the *BiTSP*. For each route obtained, *NH1* collects all available bonuses and assigns passengers using a greedy strategy [Silva et al. 2020]. In contrast, *NH2* assigns passengers first and collects bonuses afterward. *NH3* follows a different structure: it first computes the Pareto front of the *TSPP* and subsequently performs passenger assignment and travel-time evaluation, updating objective values accordingly.

Four evolutionary algorithms were implemented: *NSGA-II* [Deb et al. 2002], *MOEA/D* [Zhang and Li 2007], *IBEA* [Li et al. 2017], *SPEA2* [Zitzler et al. 2001]. *NSGA-II* employs non-dominated sorting with crowding distance to maintain diversity. *MOEA/D* decomposes the objective space into scalar subproblems using the Tchebycheff approach. *IBEA* relies on indicator-based fitness evaluation, whereas *SPEA2* combines strength-based fitness assignment with density estimation. All evolutionary algorithms adopted the same solution encoding. A solution is a triple (Π, B, Ψ), where P is a permutation vector representing the visiting order of vertices, B is a binary vector indicating whether the bonus at each visited vertex is collected, and Ψ is a list of accepted passengers. The initial population was generated randomly.

This dissertation proposed a one-point crossover operator and four mutation operators. In the one-point crossover, a cut position is randomly selected in the first parent permutation Π, and the prefix up to this position is copied to the offspring. The remaining vertices are inserted according to their relative order in the second parent, avoiding duplicates to preserve permutation feasibility. Four mutation operators were also proposed: (i) the first exchanges two vertices in Π; (ii) the second inserts a vertex into Π; (iii) the third removes a vertex from Π; and (iv) the fourth flips a bonus decision in B.

5. Computational experiments

5.1. Methodology

All algorithms were implemented in C++ and executed on an Intel Core i5 (2.4 GHz) with 8 GB RAM. The *Solver* and the naive heuristics used Gurobi 11.0 with a 7200-second time limit. The metaheuristics were configured with a population of 300, a stopping criterion of

8×10^4 evaluations, parameters tuned using the irace package [López-Ibáñez et al. 2016], and 20 independent runs per instance.

The algorithms were evaluated on 252 benchmark instances [Marques et al. 2019] with 5 to 200 vertices, comprising symmetric and asymmetric graphs in three classes based on edge cost-time correlation: *A1* (random), *A2* (positively correlated), and *A3* (negatively correlated). Negative correlation increases the number of Pareto-optimal solutions, making *A3* the most challenging class.

Solution quality and diversity were measured by the Hypervolume (*HV*) and the Inverted Generational Distance (*IGD*). The *GAP* metric in Expression (1) quantifies the relative deviation between the *Solver*'s absolute *HV* (HV_s) and that of each algorithm (HV_h). Performance differences were tested using the Friedman test [Friedman 1937] with Nemenyi post-hoc [Nemenyi 1963] at a 0.05 significance level.

$$GAP = \frac{(HV_s - HV_h) \times 100}{HV_s} \tag{1}$$

5.2. Results of the Solver

The *Solver* was able to completely enumerate the Pareto front only for instances with up to 10 vertices, within the time limit of 7200 seconds. For instances with 20 or more vertices, no solution was found due to the exponential growth of computational complexity. All 5-vertex instances had their Pareto fronts completely determined. For 8-vertex instances, the *Solver* recovered the full Pareto front in four of the six symmetric instances across the three classes (*A1*, *A2*, and *A3*), while for the asymmetric instances, the complete front was obtained in 3 (*A1*), 4 (*A2*), and 5 (*A3*) of the six instances. For 10-vertex instances, the limitation became even more pronounced. Among symmetric instances, the complete front was obtained in 1 (*A1*), 2 (*A2*), and 0 (*A3*) of the six instances. Among asymmetric instances, the complete front was obtained in 1 (*A1*), 3 (*A2*), and 0 (*A3*) of the six instances.

Class *A3* posed the greatest challenge: the *Solver* failed to compute the complete front for any 10-vertex instance in this class, regardless of graph type. The negative correlation between edge cost and time in *A3* increases the number of Pareto-optimal solutions, demanding more computational effort. In contrast, the positive correlation in class *A2* reduced the size of the Pareto front, making these instances comparatively easier to solve. Additionally, asymmetric instances produced more Pareto-optimal solutions than symmetric ones, as directed edges increase route diversity.

5.3. Heuristics and metaheuristics results relative to the Solver

Figure 2 plots the *GAP* values for class *A3*, the most challenging class. *NH1* and *NH2* yielded the largest *GAP* values across all instances, indicating poor solution quality even when the *Solver* computed only a partial front. *NH3* reduced the *GAP* relative to *NH1* and *NH2*, but remained inferior to all metaheuristics.

In contrast, the metaheuristics attained $GAP = 0$ for all 5-vertex instances across every class, matching the complete Pareto-optimal front. Negative *GAP* values occurred for some 8- and 10-vertex instances where the *Solver* reached the time limit, indicating

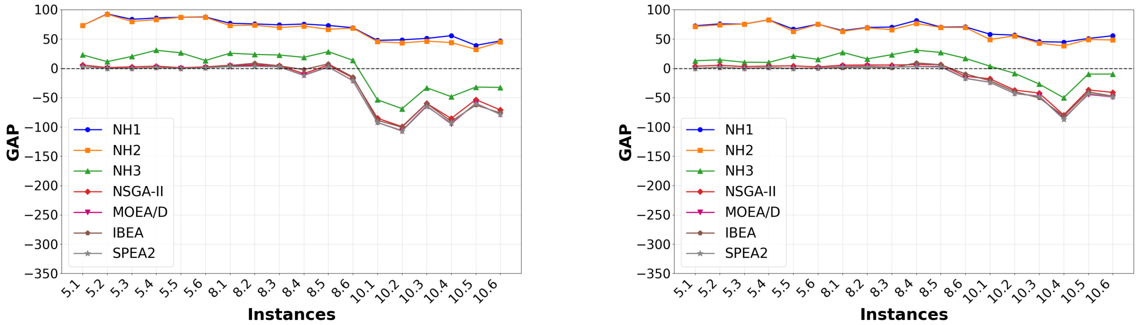

Figura 2. GAP between the absolute HV of the Solver and those achieved by heuristics and metaheuristics for class A3: symmetric (left) and asymmetric (right).

that the metaheuristics found more non-dominated solutions than the *Solver's* incomplete front.

5.4. Processing time of heuristics and metaheuristics

While the *Solver* was limited to instances with at most 10 vertices, the naive heuristics computed solutions for instances with up to 200 vertices within the 7200-second limit. However, class *A3* remained the most demanding: *NH1* and *NH2* reached the time limit at 50 vertices, and *NH3* at 100 vertices. Since these heuristics internally solve the *BiTSP* or *TSPP* exactly, the fact that they handled larger instances than the *Solver* confirms that the *MoTSPPP* is computationally harder than both subproblems.

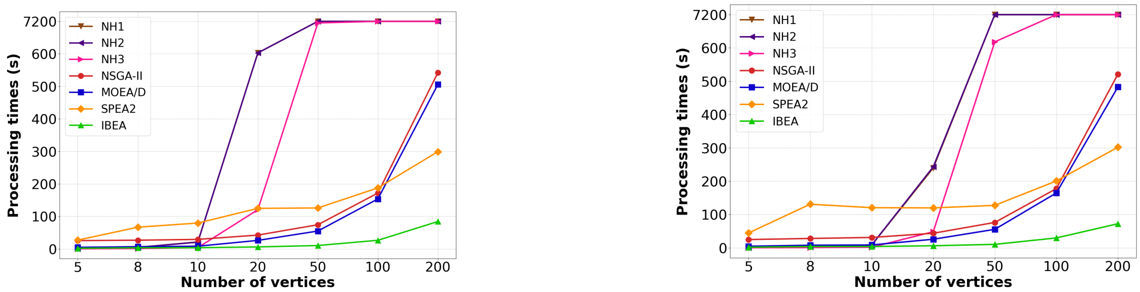

Figura 3. Average processing times of metaheuristics for class A3: symmetric (left) and asymmetric (right).

As shown in Figure 3, metaheuristic processing times increased with instance size but never reached the 7200-second limit. *SPEA2* was the slowest for instances with up to 100 vertices due to its fitness and density estimation procedures, while *NSGA-II* and *MOEA/D* became more time-consuming at 200 vertices as non-dominated sorting and decomposition operations scaled with the Pareto front size. *IBEA* was consistently the fastest across all instance sizes.

5.5. Solution quality of heuristics and metaheuristics

Figure 4 shows the average *HV* and *IGD* evolution during the execution of metaheuristics on a 200-vertex instance from class *A1*. *IBEA* exhibited rapid initial improvements but stagnated early, while the remaining metaheuristics achieved steadier gains over time.

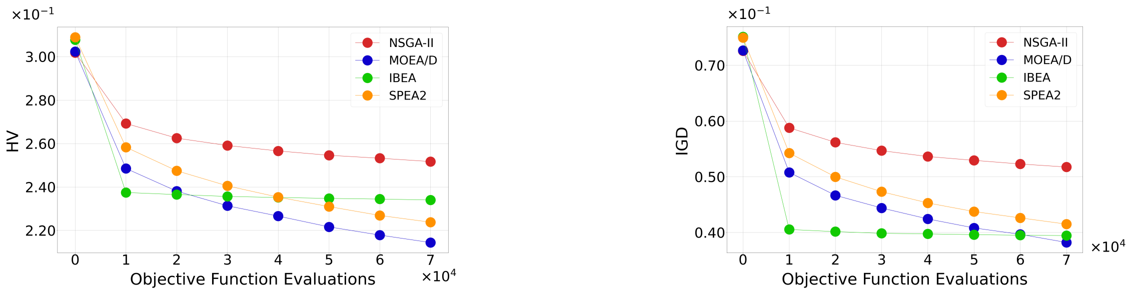

Figura 4. Average HV (left) and IGD (right) evolution on the 200.1 asymmetric instance from class A1.

MOEA/D ultimately reached the best *HV* and *IGD* values, followed by *SPEA2*, *IBEA*, and *NSGA-II*.

These observations are supported by the statistical analysis in Figure 5, which summarizes the results of the Friedman and Nemenyi post-hoc tests across all instances. Algorithms connected by bold lines show no statistically significant differences, and rank 1 indicates the best algorithm.

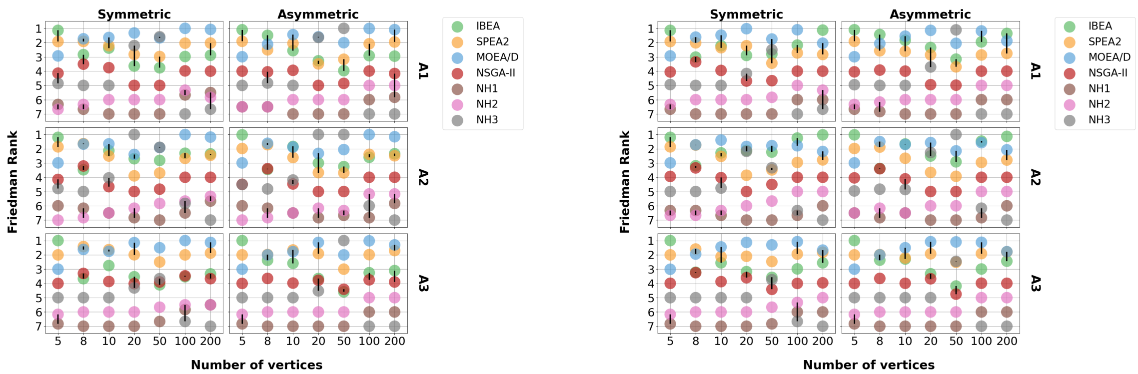

Figura 5. Summary of Friedman and Nemenyi post-hoc test results for HV (left) and IGD (right).

For 5-vertex instances, *IBEA* and *SPEA2* ranked first and second across all classes, with *MOEA/D* typically third. At sizes 8 and 10, *MOEA/D* and *SPEA2* alternated in the top rank with no significant differences between them, while both outperformed *IBEA*. For 20 or more vertices, *MOEA/D* consistently ranked first with statistical superiority in most scenarios, except for 200-vertex instances from class *A3*, where it was statistically tied with *SPEA2*. *NSGA-II* generally ranked below the other three metaheuristics.

The naive heuristics yielded the worst overall results. A notable exception was *NH3* in 50-vertex instances, where it achieved the best *HV* with statistical superiority in asymmetric instances from all three classes. However, *NH3*'s performance declined substantially at 100 and 200 vertices, consistently yielding the worst *HV* and *IGD* values. *NH1* and *NH2* ranked lowest in almost all scenarios, with *NH2* consistently outperforming *NH1* as instance size increased.

5.6. Discussion

The results reveal two key findings about the *MoTSPPP*. First, the sequential decomposition employed by the naive heuristics proved ineffective: despite solving the *BiTSP* or *TSPP* exactly, these heuristics were consistently outperformed by all metaheuristics. This confirms that treating routing, bonus collection, and passenger assignment as independent phases neglects the circular dependency among the objective functions.

Second, the relative performance of the metaheuristics varied with problem size and Pareto front structure. *IBEA* and *SPEA2* excelled on small instances, where hypervolume-based and density-driven selection are effective given the limited number of non-dominated solutions. As the Pareto front grows, particularly in class *A3* and instances with 100 or more vertices, *MOEA/D*'s decomposition strategy proved more effective at balancing diversity and convergence. *NSGA-II*'s dominance-based ranking was the least competitive, struggling to maintain diversity in scenarios with large Pareto fronts.

6. Conclusion

This dissertation proposed a mixed-integer linear programming model for the *MoTSPPP* and evaluated eight algorithms spanning exact, heuristic, and metaheuristic paradigms on 252 benchmark instances. The *MoTSPPP* captures a circular dependency among travel cost, travel time, and bonus collection that arises from the cost-sharing mechanism with passengers, a structure absent in previous formulations.

The results showed that the *MoTSPPP* is computationally harder than both the *BiTSP* and the *TSPP*, and that the circular dependency among its objectives cannot be effectively handled by sequential decomposition. Among the metaheuristics, *MOEA/D* achieved the best overall performance for medium and large instances, suggesting that decomposition-based strategies are particularly suited for problems with large Pareto fronts.

These findings open several research directions, including the design of specialized operators exploiting the cost-sharing structure of the *MoTSPPP*, hybrid approaches combining evolutionary algorithms with local search, and the integration of real-world datasets and fitness landscape analysis to support practical deployment in collaborative transportation and on-demand mobility.

Referências

Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. (2012). Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303.

Awerbuch, B., Azar, Y., Blum, A., and Vempala, S. (1995). Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 277–283.

Balas, E. (1989). The prize collecting traveling salesman problem. *Networks*, 19(6):621–636.

Cai, J., Zhu, Q., Lin, Q., Ma, L., Li, J., and Ming, Z. (2023). A survey of dynamic pickup and delivery problems. *Neurocomputing*, page 126631.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Dell'Amico, M., Maffioli, F., and Värbrand, P. (1995). On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research*, 2(3):297–308.

Feillet, D., Dejax, P., and Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation Science*, 39(2):188–205.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

Gunawan, A., Lau, H. C., and Vansteenwegen, P. (2016). Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332.

Li, W., Özcan, E., John, R., Drake, J. H., Neumann, A., and Wagner, M. (2017). A modified indicator-based evolutionary algorithm (mIBEA). In *IEEE Congress on Evolutionary Computation (CEC)*, pages 1047–1054. IEEE.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., and Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.

Marques, T. S., Cezario, S. F., Goldbarg, E. F. G., Goldbarg, M. C., and Maia, S. M. D. M. (2019). Quota traveling salesman with passengers and collection time. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 299–304. IEEE.

Nemenyi, P. B. (1963). *Distribution-Free Multiple Comparisons*. Princeton University.

Silva, B. C., Fernandes, I. F., Goldbarg, M. C., and Goldbarg, E. F. (2020). Quota travelling salesman problem with passengers, incomplete ride and collection time optimization by ant-based algorithms. *Computers & Operations Research*, 120:104950.

Silva, J. B., Brito, T. A., Rios, R. A., Rios, T. N., and Fernandes, I. F. (2025). Multi-objective traveling salesman problem with profits and passengers: Exact, heuristic and evolutionary approaches. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 419–422.

Tsiligirides, T. (1984). Heuristic methods applied to orienteering. *Journal of the Operational Research Society*, 35(9):797–809.

Yue, C., Song, J., Liu, M., Bi, Y., Guo, W., Lin, H., and Liang, J. (2026). An evolutionary algorithm for multimodal multi-objective traveling salesman problems. *Expert Systems with Applications*, 297:129540.

Zhang, Q. and Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731.

Zitzler, E., Laumanns, M., and Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. *TIK Report*, 103.