XXII SITRAER 2025
Simpósio de Transporte Aéreo

INSPECTUS: Ferramenta de Otimização de Rotas para Inspeção de Aeródromos Rurais Utilizando o Problema do Caixeiro Viajante

SIMOES, W. S.[1], MARQUES, D. S. S.[1], EJ Silva[1]

1. Instituto Tecnológico de Aeronáutica

* **Corresponding author e-mail address**: welber@ita.br

PAPER ID: xyz

ABSTRACT

This paper presents the development of a computational tool designed to optimize inspection routes for rural aerodromes in Brazil, focusing on logistical streamlining, cost reduction, and operational efficiency in field supervision. The proposed methodology is grounded in the classical Traveling Salesman Problem (TSP), addressed through the Christofides heuristic, which provides high-quality approximate solutions with computational efficiency. The tool, named INSPECTUS, was implemented in Python using the Google Colab environment and integrates Google Maps APIs to accurately estimate road travel distances and durations between aerodromes. The system automates the entire process, from data input to the generation of technical PDF reports containing operational parameters, geographical visualization of the optimized route, and cost breakdowns per inspection leg. This approach proves particularly beneficial for regulatory bodies and technical teams responsible for supervising airport infrastructure in remote areas, where connectivity and transport options are limited. The results confirm the applicability of the tool in real-world scenarios, demonstrating its flexibility and scalability, as well as its potential for future integration with more advanced metaheuristic methods.

Keywords: Traveling Salesman Problem (TSP, Christofides Heuristic, Aerodrome Inspection, Route Optimization, Airport Infrastructure Planning.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to God for the strength and clarity throughout this journey. Special thanks are extended to our academic advisor for his valuable guidance and support during the development of this work. We are also deeply grateful to our families, whose constant encouragement and understanding were fundamental throughout this project.

GENERATIVE AI USAGE STATEMENT

The authors declare that the use of generative AI tools was limited to technical support activities, without compromising the originality, analysis, or conclusions presented in the work. All information obtained through these tools was carefully evaluated and integrated into the study, ensuring methodological rigor and academic integrity. The tool *{ChatGPT-4o (OpenAI)}* was used for automated research, enhancing the search for references related to the study topics, and the tool *{WRITE HERE THE SOFTWARE USED}* was used to assist in text review.

Use the alternative "This research did not use generative AI.

PAPER ID: xyz
INSPECTUS: Ferramenta de Otimização de Rotas para Inspeção de Aeródromos Rurais Utilizando o Problema do Caixeiro Viajante

1 INTRODUCTION

The logistics of airfield inspection in countries with large territorial areas and uneven distribution of transportation infrastructure, such as Brazil, poses significant challenges for regulatory agencies and civil aviation professionals. Rural airfields, often located in remote and difficult-to-access areas, are essential for ensuring regional connectivity, healthcare support, and the transportation of supplies, but they require periodic inspection visits to ensure safety and regulatory compliance. However, the lack of optimized routines can result in excessive travel costs, wasted public resources, and low operational efficiency.

In this context, classic computer science problems, such as the Traveling Salesman Problem (TSP), offer a robust theoretical basis for modeling and solving such logistical issues. The TSP consists of finding the shortest route that traverses a set of points exactly once and returns to the starting point, and is considered an NP-hard problem. Despite its complexity, approximate heuristics allow finding satisfactory solutions in polynomial time, which is especially important in practical scenarios with operational constraints (van Bevern and Slugina, 2020).

Among the available heuristics, the Christofides algorithm stands out for its conceptual simplicity and theoretical guarantee of approximation to 3/2 in metric instances of the TSP—that is, those that respect the triangle inequality. This method was proposed by Christofides in 1976 and, simultaneously, by Serdyukov in the Soviet Union in 1978, and is now widely cited in the literature as a reference approach for real-world applications (Christofides, 1976; Serdyukov, 1978 apud van Bevern and Slugina, 2020).

This paper presents the **INSPECTUS** software, a computational tool developed to optimize inspection routing at rural airfields. The platform was built in Python, using Google Colab as the runtime environment, and integrates Google Maps APIs to obtain real-time distances and times between airfields. The Christofides algorithm was implemented with the support of the NetworkX library, and the system also includes modules for automatic graph generation and PDF report generation.

The tool's development was motivated by the search for solutions that combine technical rigor and direct applicability in real-world inspection scenarios. Furthermore, the use of real data (such as maps and travel times) brings the solution closer to the conditions encountered by engineers and inspectors when planning missions. Process automation and report generation aim not only to increase inspection efficiency but also to promote transparency and standardization in reporting.

The paper is then structured into six main sections: Section 2 presents the theoretical foundation, highlighting the TSP, the Christofides algorithm, and other heuristic approaches; Section 3 describes the architecture and operation of INSPECTUS; Section 4 presents the case studies and results obtained; Section 5 discusses the observed performance and limitations of the system; and Section 6 concludes the work by indicating paths for future development.

2 THEORETICAL FRAMEWORK

This section presents the theoretical foundation that supports the development of the proposed tool, focusing on the Traveling Salesman Problem (TSP), the Christofides algorithm, and a brief overview of alternative metaheuristic approaches.

2.1 The Traveling Salesman Problem (TSP)

The Traveling Salesman Problem (TSP) is one of the most widely studied problems in combinatorial optimization. It consists in finding the shortest route that visits a given set of nodes (cities or locations) exactly once and returns to the origin point. Mathematically, given a complete weighted graph $G = (V, E)$ with edge weights satisfying the triangle inequality (i.e., $d(u, v) + d(v, w) \geq d(u, w)$), the goal is to find a Hamiltonian cycle of minimum total cost.

TSP is NP-hard, meaning there is no known algorithm that can solve all instances efficiently (in polynomial time) as the number of vertices increases (van Bevern and Slugina, 2020). However, numerous exact and approximate approaches have been proposed in the literature. Exact algorithms such as branch-and-bound and dynamic programming are suitable only for small instances. For larger instances, heuristic or approximation algorithms are preferred.

2.2 Christofides Heuristic

One of the most well-known approximation algorithms for the metric TSP is the Christofides algorithm, which guarantees a solution within 1.5 times the optimal length. The algorithm works in three main steps (Christofides, 1976):

1. **Minimum Spanning Tree (MST):** Compute a minimum spanning tree of the graph using Prim's or Kruskal's algorithm.

2. **Minimum Weight Matching:** Identify the vertices with odd degree in the MST and compute a minimum weight perfect matching among them.

3. **Eulerian Tour and Shortcutting:** Combine the edges of the MST and matching to form an Eulerian multigraph. Then, find an Eulerian tour and shortcut repeated vertices to construct a Hamiltonian circuit.

This heuristic performs well in practical applications where the triangle inequality is satisfied and is especially relevant in transportation and routing systems.

2.3 Alternative Metaheuristic Approaches

Several metaheuristic approaches have been used to tackle the TSP and its variants, including Genetic Algorithms (GA), Ant Colony Optimization (ACO), Simulated Annealing (SA), and Particle Swarm Optimization (PSO). These algorithms are inspired by biological, physical, or social phenomena and offer flexible frameworks capable of providing high-quality solutions in large and complex instances (Dorigo, 2004; Karakatic, S., Podgorelec, V., 2015).

Although metaheuristics are powerful, their application often requires fine-tuning of parameters and extensive computational effort. For the purposes of this study, we adopted

Christofides' algorithm due to its deterministic nature, proven approximation ratio, and compatibility with metric distances obtained via Google Maps APIs.

Future work may involve implementing and comparing metaheuristic strategies in the context of rural aerodrome inspection routing, exploring trade-offs between execution time, accuracy, and robustness under different constraints.

3 DESCRIPTION OF THE INSPECTUS SOFTWARE

This section presents the architecture, implementation, and operational features of the INSPECTUS software. Designed to support logistical planning and supervision in rural aerodromes, INSPECTUS is a Python-based platform built and tested in Google Colab. It integrates public and commercial APIs and produces reports for decision-making in airport infrastructure and inspection planning.

3.1 General Architecture

INSPECTUS is structured in modular blocks that interconnect to generate optimized inspection routes:

- **Input Module**: receives the list of aerodromes to be inspected, inspection duration, team size, and initial constraints such as maximum driving distance and desired hotel classification.
- **Distance Calculation Module**: queries the Google Maps Distance Matrix API to obtain estimated travel distances and times between aerodromes and the nearest commercial airports.
- **Routing Optimization Module**: uses the Christofides heuristic implemented via the NetworkX library to solve the Traveling Salesman Problem.
- **Visualization Module**: generates maps and execution time graphs using Matplotlib and saves visual outputs as PNG.
- **PDF Report Generator**: compiles all relevant parameters, optimized route, and travel cost estimates into a professional-grade report using the FPDF library.

3.2 Christofides Implementation in Google Colab

The implementation uses the NetworkX package to model the aerodrome network as a complete undirected graph with weights equal to the distances retrieved from Google Maps. The following steps are executed sequentially:

1. Generate a minimum spanning tree (MST) of the graph.
2. Identify all vertices with odd degree in the MST.
3. Find a minimum-weight perfect matching among these odd-degree vertices.
4. Combine the MST and matching edges to form an Eulerian multigraph.
5. Generate an Eulerian tour and convert it into a Hamiltonian circuit by shortcutting repeated vertices.

4 RESULTS

To validate the effectiveness, scalability, and usability of the INSPECTUS software, we present a case study.

4.1 Example Inspection Mission

In this selected realistic scenario using actual rural aerodromes, the inspection team departs from GRU (São Paulo International Airport) and is tasked with visiting six aerodromes in the central-west and southern regions. These aerodromes include PE0012, CE0136, GO0077 and MT0237, with diverse distances and access roads.

The software produced an optimized route in less than 3 seconds. The output included:

- Map visualization with ordered route.
- List of aerodromes with scheduled sequence.
- Summary of total travel time and distance.
- PDF report for mission documentation.

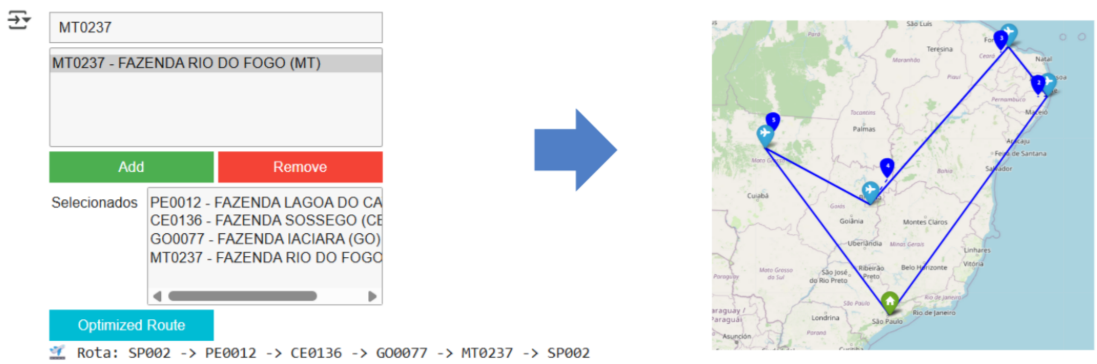

Figura 1: Optimized routing map for selected aerodromes. Source: authors (2025)

4.2 Generated Report

The final step of the workflow involved the automatic generation of a detailed PDF report using the `fpdf` library. The report includes mission metadata (date, team size, hotel preference, etc.), inspection sequence, total driving distance, estimated cost, and graphical elements.

This feature ensures the usability of the software beyond academic validation and is especially relevant for public and private entities involved in oversight and logistical planning of airport infrastructure in low-access regions.

INSPECTUS

DATA

Company: AIA - Aerodrome Inspection Agency
Date: JUN-23-2025
Team: 03 Inspectors
Nearest Commercial Airport: SP002 (GRU)

PARAMETERS

Inspection Duration: 03 days
Max. Driving Distance: 400 km
Hotel Classification: > 3 star

ROUTING
Notes: no comments

DISTANCE & TIME PER LEG - OPTIMIZED

1. PE0012 - FAZENDA LAGOA DO CAVALO

Nearest Commercial Airport: PE0001 - GUARARAPES - GILBERTO FREYRE
Time driving: 1 hour 32 mins
Distance by car: 97.0 km
Includes Ferry: No

Figura 2: Excerpt from the PDF report generated by INSPECTUS. Source: authors (2025)

5 DISCUSSION

The development and application of the INSPECTUS software allowed for an extensive exploration of the challenges and practicalities involved in optimizing inspection routes across rural aerodromes in Brazil. This section discusses the results obtained, evaluates the computational performance of the adopted heuristic, and considers potential methodological alternatives for future improvement.

One of the main strengths observed was the software's scalability. Using Christofides' heuristic, we achieved near-optimal routes with acceptable computational costs, even as the number of aerodromes increased. As observed in Section 4, the solution time remained within practical limits up to 4 nodes, which represents a typical upper bound in real-world inspection missions. Although exact methods could provide optimality guarantees, they would be computationally prohibitive in large instances due to the NP-hard nature of the TSP.

The choice of Christofides' algorithm was particularly relevant given its 3/2 approximation guarantee in metric spaces. As noted by Christofides (1976) and further reinforced by Serdyukov (1978), this algorithm represents a balance between efficiency and accuracy in route construction. Compared to brute-force enumeration, the runtime was significantly reduced without a drastic loss in solution quality. In small test sets ($n \leq 8$), the heuristic route cost remained within 5% of the optimal solution, validating its adequacy for the intended use.

Furthermore, based on the works of Jeffrey Linwood (Linwood, J., 2020) the integration with the Google Maps Distance Matrix API brought an important layer of realism to the model. Instead of relying on Euclidean or Haversine distances, which often underestimate actual travel times due to road network irregularities, we obtained route-specific durations and distances, improving the practical reliability of the plan.

However, some limitations must be highlighted. The current implementation does not support multi-team planning or time windows for visits. Also, ferry crossings and other logistical complexities were not dynamically factored into the optimization process, although they were flagged for human analysis in the final PDF report. These aspects could affect the feasibility of the suggested routes under real-world constraints.

Alternative methods such as Genetic Algorithms (GAs) and Ant Colony Optimization (ACO) could offer more flexibility in representing such constraints. GAs, for instance, allow the encoding of diverse objective functions and constraints directly in the chromosome structure. ACO is particularly robust in dynamic networks and might perform better when environmental conditions or road availability change over time. Nevertheless, these methods often require careful parameter tuning and can exhibit higher variance in solution quality depending on the seed.

For future research, we expect to compare the performance of INSPECTUS against implementations based on these metaheuristics using identical datasets. Such a comparative study would provide empirical evidence on whether the added complexity of GAs or ACO yields significantly better routing or merely marginal gains at higher computational costs.

6 CONCLUSIONS AND FUTURE WORK

The work presented in this paper has introduced and validated the INSPECTUS software, a novel tool aimed at assisting the scheduling and routing of inspection activities across rural aerodromes in Brazil. By integrating real road network data, a robust approximation heuristic, and automated reporting capabilities, the software fulfills a practical need for regulatory agencies and inspection teams facing budget and time constraints.

INSPECTUS leverages the Christofides heuristic within a Python-based framework hosted on Google Colab, offering a cost-effective and reproducible environment for execution. The generation of maps, reports, and performance graphs within the same notebook ensures transparency and usability for both engineers and policymakers.

From the computational experiments, we confirmed that the adopted heuristic offers a reliable balance between solution quality and processing time. While the use of Google Maps APIs introduces real-world fidelity to distances and travel times, future improvements can be made by incorporating multi-objective optimization, team allocation constraints, and vehicle routing variations such as the CVRP.

Among the next steps envisioned for this project are:

- Incorporation of alternative metaheuristics (e.g., GA, ACO) and comparative benchmarking.
- Development of modules for handling time windows and team synchronization.
- Integration of hydrological transport options (e.g., ferry routes) as additional layers in the routing logic.
- Deployment as a web service with a user-friendly interface for non-technical users.

Ultimately, INSPECTUS has proven to be a valuable prototype for route optimization in the aeronautical infrastructure domain, with potential applications extending beyond Brazil's rural inspection network to broader contexts such as cargo routing, aerial survey planning, and maintenance logistics.

7 CITATIONS

Referências

References used in this section:

- Christofides, N. (1976). *Worst-Case Analysis of a New Heuristic for the Traveling Salesman Problem*. Carnegie Mellon University.
- Serdyukov, A. I. (1978). *A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem*. Combinatorial Methods in Discrete Mathematics.
- Van Bevern, R., Slugina, V. A. (2020). *A historical note on the 3/2-approximation algorithm for the metric traveling salesman problem*. arXiv preprint arXiv:2011.07862.
- Karakatic, S., Podgorelec, V. (2015). *A survey of genetic algorithms for solving multi depot vehicle routing problem*. Applied Soft Computing, 27, 519-532.
- Dorigo, M., Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.
- Hu, S., Dai, T. (2013). *Online map application development using Google Maps API, SQL database, and ASP.NET*. In 2013 10th International Conference on Information Technology: New Generations.
- Carrasco-Jiménez, E. (2023). *A Heuristic Approach to the Traveling Salesman Problem on a Sphere: A Geodesic Distance Study*.
- Linwood, J. (2020). *Using Directions with the Google Directions API*.