ISSN: 1692-7257 - Volume 1 – Number 45 - 2025

RCTA
Revista Colombiana de Tecnologías de Avanzada
UNIPAMPLONA

Digital Object Identifier: 10.24054/rcta.v1i45.3079

Solution to the work plan generation problem based on the traveling salesman agent problem using the genetic algorithm Chu-Beasley

Solución al problema de generación de planes de trabajo basado en el problema del agente viajero utilizando el algoritmo genético Chu-Beasley

PhD.(c) John Fredy Castaneda Londoño[1],
PhD. Ramón Alfonso Gallego Rendón[1],
PhD. Eliana Mirledy Toro Ocampo[2]

[1] ***Universidad Tecnológica de Pereira,*** *Facultad de Ingenierías, Pereira, Risaralda, Colombia.*
[2] ***Universidad Tecnológica de Pereira,*** *Facultad de Ciencias Empresariales, Pereira, Risaralda, Colombia.*

Correspondence: *jfcastaneda@utp.edu.co*

Received: *august 16, 2024.* ***Accepted:*** *december 17, 2024.* ***Published:*** *january 01, 2025.*

How to cite: *J. F. Castañeda Londoño, R. A. Gallego Rendón, and E. M. Toro Ocampo, "Solution to the work plan generation problem based on the traveling salesman agent problem using the genetic algorithm Chu-Beasley", RCTA, vol. 1, no. 45, pp. 66–73, jan. 2025. Recovered from https://ojs.unipamplona.edu.co/index.php/rcta/article/view/3079*

Abstract: This article introduces a methodology to address logistics management by generating work plans based on the Multiple Traveling Salesman Problem (MTSP). The primary objective is cost minimization, which can manifest in two forms: distance or the time required for executing work plans. Two methods are used to measure distance: the first calculates spherical distances using the Haversine formula, and the second leverages Google Maps data to obtain traffic-related travel time information. One of the goals of this methodology is to validate the benefits of carrying out optimizations on multiple routes affected by traffic variables, resulting in travel times, within a modern, modular, and rapidly implementable software architecture. To solve the mathematical model, the Chu-Beasley genetic algorithm is used with enhancements through the Or-Opt operator, aiming to obtain high-quality solutions with reasonable times in daily logistics planning. In the results analysis stage, tests were conducted using instances obtained from real company locations whose operations are affected by logistics performance. The study's results were compared to those of an expert in the field using the nearest neighbor algorithm as a reference, focusing on the distance variable, which demonstrated significant improvements and confirmed the benefits of using logistics planning with optimization algorithms considering traffic variables.

Keywords: traveling salesman problem, genetic algorithm, traffic.

Resumen: Este artículo introduce una metodología para abordar la gestión logística al generar planes de trabajo basado en el problema del Agente Viajero Múltiple (MTSP) y tienen como objetivo la minimización de costos, que puede manifestarse en dos formas: la distancia o el tiempo requerido para la ejecución de los planes de trabajo. Se emplean dos

métodos para medir la distancia: el primero calcula distancias esféricas utilizando la fórmula de Haversine, y el segundo aprovecha datos de Google Maps para obtener información de tráfico asociada a tiempos de viaje. Uno de los propósitos de esta metodología es validar los beneficios de llevar a cabo optimizaciones en rutas múltiples que se ven afectadas por variables de tráfico, que se traducen en tiempos de recorrido, sobre una arquitectura de software moderna, modular y de rápida implementación. Para resolver el modelo matemático, se utiliza el algoritmo genético Chu-Beasley con mejoramientos a través del operador Or-Opt, con el objetivo de obtener soluciones de calidad con tiempos razonables en la planificación logística diaria. En la etapa de análisis de resultados, se llevaron a cabo pruebas con instancias obtenidas de ubicaciones reales de empresas cuyas operaciones se ven afectadas por el rendimiento logístico. Los resultados del estudio se compararon simulando un experto en el área utilizando el algoritmo del vecino más cercano como referencia y centrándose en la variable de distancia, evidenciando mejoras significativas y confirmando el beneficio del uso del planeamiento logístico usando algoritmos de optimización con variables de tráfico.

Palabras clave: problema de agente viajero, algoritmo genético, condiciones de tráfico.

1. INTRODUCTION

The Traveling Salesman Problem (TSP) is a fundamental combinatorial optimization problem that has received significant attention in recent decades. Its relevance stems from its wide range of real-world applications, including logistics operations, courier and package delivery services, circuit board drilling, scheduling maintenance for correcting faults in utility company systems, and many others.

Mathematically, the problem is formulated as follows: Given a list of n cities along with the distances between each pair of them, the objective is to find a route that visits each city exactly once and minimizes the total length of the route. The TSP is classified as NP-complete, implying that there is no algorithm that can solve it in deterministic time regardless of the number of cities.

However, there are various approximate algorithms that can find acceptable solutions in reasonable times, especially for moderately sized instances, depending on what is considered reasonable time. When the problem requires finding a solution for more than one agent, it is referred to as the Multiple Traveling Salesman Problem (MTSP).

To frame the problem in real logistics scenarios, Google Maps distances were used. These distances are obtained through Dijkstra's algorithm, which is an efficient algorithm for calculating the shortest distance between two points in a graph. An additional advantage of using Google Maps distances is the possibility of combining objectives associated with the distance between customers, such as time. The time between two customers is affected by traffic, thus the Google Maps distance matrix increases accuracy for solving problems closer to common logistics scenarios.

A brief review of the development of the TSP over the years since its definition takes us to 1954, where the Traveling Salesman Problem (TSP) was formally introduced by George Dantzig, Ray Fulkerson, and Selmer Johnson. This marks the beginning of research in combinatorial optimization problems in the context of route planning. In 1956, Menger and Blumenthal independently extended the TSP to the Multiple Traveling Salesman Problem (MTSP) by introducing the concept of multiple salesmen. They also proposed a heuristic to solve the problem [1].

1970s-1980s: Significant progress is made in the development of algorithms to solve the MTSP. Researchers explore ways to optimize the assignment of cities to salesmen, as well as the order in which each salesman visits the cities. Dynamic programming and mathematical programming techniques are also applied to the problem [2].

1990s: Researchers increasingly focus on practical applications of the MTSP, including vehicle routing problems and distribution logistics. This leads to the development of specialized algorithms and software tools to solve real-world instances of the MTSP [3].

2000s: Advances in computing power and optimization techniques lead to more efficient and scalable algorithms for solving large-scale instances

of the MTSP. Metaheuristic approaches such as genetic algorithms, simulated annealing, and ant colony optimization have become popular for addressing the problem [4].

2010s: The MTSP remains a relevant and active research problem, with applications in fields such as transportation, telecommunications, and supply chain management. Researchers explore hybrid algorithms that combine multiple optimization techniques to achieve better results [5].

2020 to present: The MTSP continues to be an area of ongoing research, focusing on the development of algorithms that can handle large and complex instances of the problem. Researchers are also exploring variations of the MTSP, such as the Time-Dependent MTSP and the Stochastic MTSP, to address more realistic scenarios [6], [7], [8].

It is important to note that this provided timeline is a broad overview, and there have been numerous individual research contributions and developments in the field of the MTSP over the years.

Regarding applications identified in the literature where the TSP is addressed with distances from Google Maps services, an initial work in [9] discusses a novel approach to optimize the MTSP using a modified genetic algorithm. The approach introduces a separate chromosome for each salesman, allowing for a better representation of the problem. The article also presents a comprehensive methodology for optimal route planning with multiple salesmen and secondary constraints, using Google Maps as a starting point. The approach provides a complete framework for solving problems such as MTSP with time windows.

[10] presents the development of a web-based and Android-based food ordering system that uses heuristic algorithms to optimize product delivery. The TSP is proposed to find the shortest route between customer addresses. Additionally, GPS technology and Google Maps are leveraged to allow visualization of the routes on a map.

[11] discusses the development of a mobile application to solve the TSP on devices with Android and iOS operating systems. The study uses Google Maps APIs to obtain real-world data on locations and distances between them. It employs Genetic Algorithms (GA) and Ant Colony Optimization (ACO) algorithms to find optimal routes for the TSP. The study tested the application on different datasets and found that ACO provides better route solutions, while GA is faster in terms of computation time.

[12] propose a methodology to obtain accurate solutions for the TSP in real-world instances using Google Maps APIs. Real-world instances of the TSP often face challenges related to map accuracy and updates, especially in neighborhoods. The authors propose a methodology that uses Google Maps APIs to obtain spatial information, preprocesses the data to create a valid graph, and then applies a metaheuristic to solve the TSP. The results are compared with digital maps, highlighting the advantages of using Google Maps APIs for accurate solutions. The authors emphasize the importance of obtaining precise coordinates for points of interest and the quality of road data for a successful solution.

[13] employs the Genetic Algorithm (GA) to tackle the TSP. The study involves the collection and preprocessing of geographic coordinates of city areas, which are then used as input for the TSP-specific Genetic Algorithm (TSGA). The TSGA iteratively generates and evolves populations of possible routes, calculating their fitness based on distance. Through selection, crossover, and mutation operations, it efficiently explores and refines potential solutions, ultimately converging to the shortest route to visit all areas. MATLAB is used for implementation and analysis.

[14] develops a mobile application that employs a Genetic Algorithm (GA) and the Google Maps navigation system to solve the Route Optimization Problem (ROP) for sales routes. The ROP is a challenging problem in route planning and logistics, where the goal is to find the shortest route to visit a set of customers efficiently. The application, designed for Android devices, uses GA to calculate optimal sales routes based on factors such as distance and customer locations.

The contribution of this work is to make a comparison between using Haversine distances and Google Maps API distances to develop efficient solutions for the Traveling Salesman Problem by simulating real scenarios in companies. It uses distances and times affected by traffic as the objective function and demonstrates the benefits obtained from implementing these applications in companies where their value chain has a significant logistic component in their costs. Additionally, it aims to develop a modern, modular, and quickly implementable architecture for companies of any size.

Based on the above, the research objective of this article is defined as the development of a methodology to address the Multiple Traveling Salesman Problem (MTSP) that minimizes cost-time in daily logistics planning using two distance matrices: one calculated using the Haversine formula for spherical distances and another incorporating Google Maps information to account for traffic variables and travel time on an efficient implementation platform.

2. METHODOLOGY

The conventional definition of Traveling Salesman Problems (TSP) can be described using an undirected complete graph. In this representation, the set of customers is denoted as $V=\{0,..,n\}$ where 0 designates the central distribution hub or logistics center. The collection of edges that form the underlying structure of the graph is denoted a $A=\{(i,j)i\neq j,\forall i,j\in V\}$. Being a combinatorial optimization problem, it can be formulated as a mixed integer linear programming problem. For this, it is defined as follows:

Sets:

I	Set of logistics centers or depots
J	Customers
V	Set $I\cup J$
G	Set of active agents

Decision variables:

x_{ij}	Represents the active edges in the graph (binary).
y_i	Represents whether an agent is active (binary).
f_{ij}	A customer *j* served from the depot $i\in I$. (integer)

Parameters:

d_{ij}	The distance/cost between customers *i-j* (real)
0_i	The cost of using the workforce group *i*. (O_i=*1*) (real)
m	Number of agents (integer)

The model is defined as:

$$\min \sum_{\substack{i,j\in J\\ i\neq j}} d_{ij}x_{ij}+\sum_{j\in V} o_i y_i \quad (1)$$

Subject to:

$$\sum_{\substack{i\in V,\, j\in J\\ i\neq J}} x_{ij}=1 \quad (2)$$

$$x_{ij}+x_{ji}\leq 1 \quad \forall i,j\in V \quad (3)$$

$$\sum_{\substack{i\in V,\, j\in J\\ i\neq J}} x_{ij}\leq card(V) \quad (4)$$

$$\sum_{\substack{i\in I,\, j\in J\\ i\neq J}} f_{ij}\leq 1 \quad (5)$$

$$f_{ij}=x_{ij} \quad \forall i\in I, j\in J \quad (6)$$

$$\sum_{j\in J} x_{0j}\leq m \quad (7)$$

$$\sum_{i\in G} y_i\geq 1 \quad (8)$$

Equation (1) represents the cost function, which is simplified by using distances as the dependent variable of the function $c_{ij}=f(d_{ij})$. In addition, the costs of using a specific workforce group are included. Constraints (2) and (3) ensure that each node in the graph, except the depot, has two edges

connecting it to other nodes. This guarantees that each node is visited exactly once by each route. Constraint (4) ensures that the number of edges in the graph equals the number of nodes, as a function of the cardinality of the set *V*. Constraints (5) and (6) ensure that each customer is served by a specific depot. This is necessary to meet the use case requirements, which specify that each customer must be served by a single agent. Constraint (7) indicates that the number of routes must equal the number of available workforce groups. This is necessary to ensure that there are enough resources to serve all customers. Constraint (8) ensures that at least one workforce group is available. This is necessary to ensure that the problem has at least one feasible solution.

To obtain solutions to the proposed model, the Chu-Beasley Genetic Algorithm (CBGA) is implemented. The main characteristics of the CBGA are as follows: (i) Initial Population: The initial population is generated by creating a random permutation of the customers. (ii) Selection: Parents are selected from a subset of the population through random selection. Only the two fittest individuals are chosen for crossover. (iii) Crossover: Crossover is performed using the Partially Matched Crossover (PMX) operator [15]. This operator ensures that the resulting offspring remain feasible within the defined constraints. (iv) Mutation: Mutation is carried out by reversing a random section of an individual's route. (v) Improvement: After mutation, the Or-opt operator is used to perform local improvements on the individuals [16].

3. RESULTS

For this implementation, random locations were used to simulate assets or customers of an electrical distribution company. The dataset consists of instances with 50, 100, and 150 customers, distributed in both urban and rural areas. Additionally, specific operational conditions were considered, such as the existence of a depot or central office in the most critical urban areas. In this scenario, there are four central offices, each associated with a constant number of workforce groups.

Table 1: *Implemented Instances*

Instance	Number of Working Groups
150	20
100	7
50	5

Under these conditions, the process of assigning locations to individual headquarters, along with a designated number of working groups, involves a two-step clustering procedure: (1) Initial Clustering: A K-means algorithm is executed with the number of clusters configured to match the number of working groups. This operational constraint ensures the utilization of all available working groups. (2) Headquarters Assignment: Subsequently, a headquarters is assigned to each group using the K-means algorithm. This step involves using the centroids of the initial clusters in conjunction with the locations of the headquarters. These experiments were implemented in Python 3.9 on a laptop with an 8th generation Intel i5 vPro processor at 1.896 GHz and 4 cores, running Windows 10. The CBGA algorithm was implemented using the NumPy library.

Tables 2, 3, and 4 show the results obtained for the instances used in this work, as presented in Table 1. The results for both the Haversine distances and the Google Maps API distances reference the results using the nearest neighbor or k-NN algorithm, which simulates the most efficient empirical planning. This means that for a company whose planning is based on expert knowledge without any clear scheduling rules or strategies, the benefits would be greater. For the Haversine distances, the travel time matrix was calculated by setting a constant speed of 35 km/h. This approach is not very precise but serves as a useful reference in this experiment for making comparisons with the times between locations affected by average traffic.

From the results tables, the following is obtained: (1) Incumbent Distance: This is the distance value in kilometers for all routes. These distances are always shorter for the Haversine calculations compared to Google's, as the former are linear, and the latter are by road. (2) Savings: This corresponds to the comparison between the incumbent calculated with the Google distance matrix and the use of the K-NN algorithm. (3) Average Cluster Distance: This is the Haversine distance between the cluster centroid and the depot or logistics center. (4) Incumbent Time: This corresponds to the time for all routes. For Haversine, it is calculated by setting a constant speed of 35 km/h. For Google, it is based on information from the API. (5) Time Savings: This is calculated with the Google time matrix using the results obtained by the K-NN algorithm as a reference. (6) O.F. Time: This column represents the results when the objective function uses the Google time matrix. That is, the travel time is minimized, not the distance.

Table 2: *Results for the instance of 150 locations and 20 available working groups.*

Type of Distance	Haversine	Google	O.F Time
Incumbent Distance	1257.32 km	2771.00 km	2811.00 km
Savings	41.32 km	122.00 km	82.00 km
Savings (%)	3,18%	4,22%	2,83%
Average Cluster Distance	22.60 km	22.60 km	22.60 km
Incumbent Time	83.83 Hours	95.30 Hours	94.33 Hours
Time Savings	2.76 Hours	3.35 Hours	4.36 Hours
Execution Time	74,7 seg	404,01 seg	541,52 seg

Table 3: *Results for the instance of 100 locations and 7 available working groups.*

Type of Distance	Haversine	Google	O.F Time
Incumbent Distance	689.63 km	1570.91 km	1583.91 km
Savings	63.90 km	178.60 km	165.60 km
Savings (%)	8,48%	10,21%	9,47%
Average Cluster Distance	17.31 km	17.31 km	17.31 km
Incumbent Time	45.96 Hours	58.66 Hours	58.22 Hours
Time Savings	4.25 Hours	6.71 Hours	7.15 Hours
Execution Time	42,3 seg	285,44 seg	310,73 seg

Table 4: *Results for the instance of 50 locations and 5 available working groups.*

Type of Distance	Haversine	Google	O.F Time
Incumbent Distance	558.24 km	1191.15 km	1219.33 km
Savings	21.96 km	36.34 km	8.16 km
Savings (%)	3,79%	2,96%	0,66%
Average Cluster Distance	17.73 km	17.73 km	17.73 km
Incumbent Time	37.22 Hours	46.00 Hours	45.27 Hours
Time Savings	1.47 Hours	2.09 Hours	2.83 Hours
Execution Time	21,99 seg	97,88 seg	91,99 seg

Based on these results, using Google API information provides a competitive advantage for obtaining optimal route planning close to real logistical scenarios. Time minimization is achieved using the Google Maps API time matrix instead of the distance matrix, where this approach is the most suitable for obtaining routes. Execution using the Haversine distance matrix is generally faster because it is performed directly by the algorithm, unlike the Google Maps API, which depends on the response received via the internet using the HTTP protocol. Fig. 1 shows the visual solution for the results of the 150-20 instance using Google distances, minimizing distance, and displaying the 4 headquarters or depots. Fig. 2 shows a preliminary route in Google Maps for route 6.

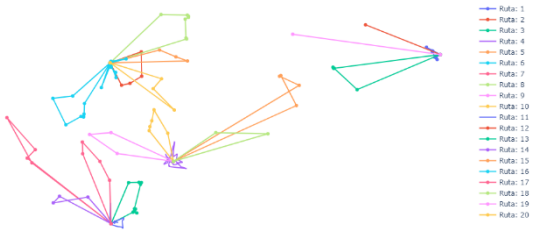

Fig. 1. *Results for the instance of 150 locations with 20 working groups.*

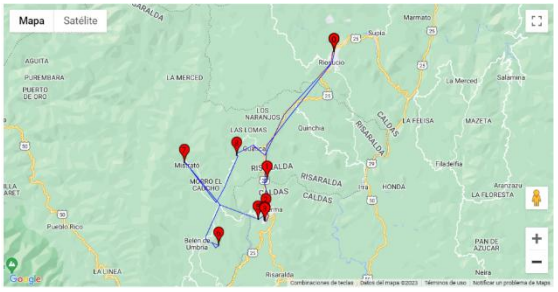

Fig. 2. *Route 6 of the 150-20 instance solution as a route sketch in Google Maps.*

For the implementation, the software architecture shown in Fig. 3 was used. It features a modular construction with a front-end for the interface or interaction and a back-end that processes the information. The computational load is handled by a RESTful API hosted on Microsoft Azure cloud services, allowing it to be easily implemented on any type of device, whether mobile or stationary, without requiring high-performance hardware.

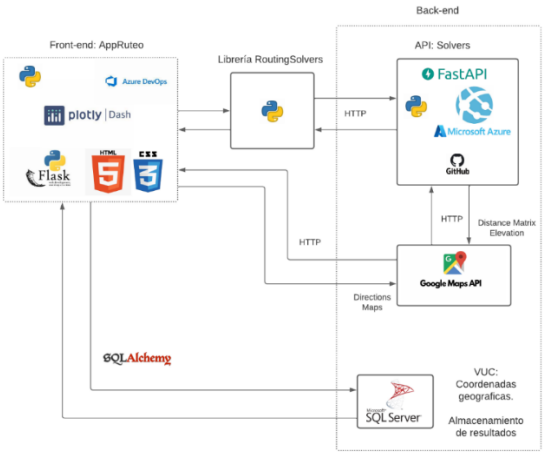

Fig. 3. *Implementation Architecture*

4. CONCLUSIONS

This work addresses the Traveling Salesman Problem (TSP), which is important in the logistics industry and even in utility companies. A methodology is presented that uses two distance matrices: one based on the Haversine formula for spherical distances and another that leverages Google Maps information to incorporate traffic and travel time variables. This methodology aims to validate the benefits of route optimization affected by traffic.

A genetic algorithm known as Chu-Beasley with the Or-Opt operator is used to obtain high-quality solutions in reasonable times. Test instances with real but random locations were used to simulate real logistical scenarios.

The results obtained in the experiments were satisfactory compared to the nearest neighbor algorithm, which simulates empirical planning with acceptable performance. The use of Google Maps information to obtain real distances and travel times on roads between clients, considering traffic, was highlighted as a competitive advantage for achieving optimal route planning in realistic logistical scenarios. Additionally, a modular implementation architecture is proposed that allows for rapid and versatile deployments.

Future work aims to develop a bi-objective methodology that, in addition to cost, minimizes greenhouse gas emissions. An additional solution algorithm incorporating local search mechanisms is also planned to enhance performance in terms of solution quality and time.

5. ACKNOWLEDGEMENTS

The authors wish to thank the Universidad Tecnológica de Pereira for its support in project 7-22-1 and Central Hidroeléctrica de Caldas for its support and feedback during the implementation.

6. REFERENCES

[1] W. R. Abel and L. M. Blumenthal, “Distance Geometry of Metric Arcs,” *Am. Math. Mon.*, vol. 64, no. 8P2, pp. 1–10, Oct. 1957, doi: 10.1080/00029890.1957.11989113.

[2] K. C. Gilbert and R. B. Hofstra, “A New Multiperiod Multiple Traveling Salesman Problem with Heuristic and Application to a Scheduling Problem,” *Decis. Sci.*, vol. 23, no. 1, pp. 250–259, 1992, doi: 10.1111/J.1540-5915.1992.TB00387.X.

[3] A. Langevin, F. Soumis, and J. Desrosiers, “Classification of travelling salesman problem formulations,” *Oper. Res. Lett.*, vol. 9, no. 2, pp. 127–132, Mar. 1990, doi: 10.1016/0167-6377(90)90052-7.

[4] T. Bektas, “The multiple traveling salesman problem: an overview of formulations and solution procedures,” *Omega*, vol. 34, no. 3, pp. 209–219, Jun. 2006, doi: 10.1016/J.OMEGA.2004.10.004.

[5] R. I. Bolaños, E. M. Toro O, and M. Granada E, “A population-based algorithm for the multi travelling salesman problem,” *Int. J. Ind. Eng. Comput.*, vol. 7, no. 2, pp. 245–256, Mar. 2016, doi: 10.5267/J.IJIEC.2015.10.005.

[6] O. Cheikhrouhou and I. Khoufi, “A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy,” *Computer Science Review*, vol. 40. 2021. doi: 10.1016/j.cosrev.2021.100369.

[7] C. Colombaroni, M. Mohammadi, and G. Rahmanifar, “Makespan minimizing on multiple travel salesman problem with a learning effect of visiting time,” *WSEAS Trans. Syst. Control*, vol. 15, pp. 477–489, 2020, doi: 10.37394/23203.2020.15.50.

[8] R. G. Mbiadou Saleu, L. Deroussi, D. Feillet, N. Grangeon, and A. Quilliot, “The Parallel Drone Scheduling Problem with Multiple Drones and Vehicles,” *Eur. J. Oper. Res.*, vol. 300, no. 2, pp. 571–589, Jul. 2022, doi: 10.1016/j.ejor.2021.08.014.

[9] A. Király and J. Abonyi, “A Google Maps based novel approach to the optimization of multiple Traveling Salesman problem for limited distribution systems,” *Acta Agrar. Kaposváriensis*, vol. 14, no. 3, pp. 1–14, 2010, [Online]. Available: https://journal.uni-mate.hu/index.php/aak/article/view/1952

[10] R. D. H. Tobing, “A food ordering system with delivery routing optimization using global positioning system (GPS) technology and google maps,” *Internetworking Indones. J.*, vol. 8, no. 1, pp. 17–21, 2016.

[11] İ. İlhan, “An Application on Mobile Devices with Android and IOS Operating Systems Using Google Maps APIs for the Traveling Salesman Problem,” *Appl. Artif. Intell.*, vol. 31, no. 4, pp. 332–345, 2017, doi: 10.1080/08839514.2017.1339983.

[12] L. G. Hernández-Landa and R. E. Mata-

Martínez, “Accurate solutions for real instances of the traveling salesman problem using Google Maps APIs,” *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, vol. 2018, no. SEP, pp. 837–843, 2018.

[13] Z. Al-Jabbar, “Using Genetic Algorithm to Solve Travelling Salesman Optimization Problem Based on Google Map Coordinates for Duhok City Areas,” *Acad. J. Nawroz Univ.*, vol. 7, no. 3, pp. 99–114, 2018, doi: 10.25007/ajnu.v7n3a207.

[14] C. Zambrano-Vega, G. Acosta, J. Loor, B. Suárez, C. Jaramillo, and B. Oviedo, “A Sales Route Optimization Mobile Application Applying a Genetic Algorithm and the Google Maps Navigation System,” *Adv. Intell. Syst. Comput.*, vol. 918, pp. 517–527, 2019, doi: 10.1007/978-3-030-11890-7_50.

[15] D. E. Goldberg and R. Lingle, “Alleles, loci, and the traveling salesman problem,” in *Proceedings of an International Conference on Genetic Algorithms*, 1985, pp. 10–19.

[16] R. A. Gallego Rendón, E. M. Toro Ocampo, and A. H. Escobar Zuluaga, *Técnicas Heurísticas y Metaheurísticas.* Universidad Tecnológica de Pereira. Vicerrectoría de Investigaciones, Innovación y Extensión. Ingenierías Eléctrica, Electrónica, Física y Ciencias de la Computación, 2015. Accessed: Sep. 03, 2023. [Online]. Available: https://unilibros.co/gpd-tecnicas-heuristicas-y-metaheuristicas.html