Journal of Multidisciplinary Engineering Science Studies (JMESS)
ISSN: 2458-925X
Vol. 9 Issue 11, November - 2023

Technology Entrepreneurship Based On Effective Delivery Routes

Juan José Paniagua Medina
Universidad de Guanajuato
Departamento de Estudios Multidisciplinarios, Yuriria, Gto.
jj.paniaguamedina@ugto.mx

Jorge Armando López Lemus
Universidad de Guanajuato
Departamento de Estudios Multidisciplinarios, Yuriria, Gto.
lopezja@ugto.mx

***Abstract*— Currently, any logistics company is looking to maximize the efficiency and profitability of the supply chain. And for this purpose, the optimization systems of transport routes are basic, for this reason the creation of algorithms that allow to solve this problem is necessary. This research work focuses on the creation and design of four metaheuristic algorithms such as the Greedy algorithm, the stochastic algorithm based on random movements, the Iterative Local Search (ILS) algorithm, and finally the Simulated Annealing algorithm. The main characteristic of which is to start from an initial point, and through the exploration to the other points (neighborhood) vary the search solution, until returning to the starting point. This is achieved by means of the instances of the combinatorial problems to explore, a computational experimentation is implemented by means of which the behavior of each of the algorithmic methods that help to solve them can be observed. The use of these algorithms is intended to find the effective delivery routes for products or services, minimizing costs and delivery times. In addition, delivery route optimization can also contribute to environmental sustainability by reducing the distance traveled and carbon emissions.**

Keywords—Metaheuristics; greedy; simulated annealing; iterative local search; stochastic; route optimization; technological entrepreneurship.

I. INTRODUCTION

At the present time the competitiveness in the national and global market is very difficult because the products need to be transported from the place where they are produced to the hands of the customer, that is why companies invest in improving their logistics systems. Within these logistic systems the mobilization of products is one of the main aspects, transportation is responsible for moving finished products, raw materials and inputs, between companies and customers that are geographically dispersed, and adds value to the transported products when they are delivered on time, undamaged and in the required quantities [1]. Likewise, transportation is one of the key points in customer satisfaction. However, it is one of the highest logistics costs and constitutes a representative proportion of product prices. The costs associated with transportation are highly representative in the supply chain and are directly involved in the relationship with suppliers, customers and competitors [2]. This is why it is necessary to implement techniques to speed up and streamline the delivery of goods. One of the methods is to form delivery groups with customers that are closer to each other, in order to save fuel for delivery transports [3].

Companies such as Amazon, DHL, Estafeta, among others, have this type of systems that allow the optimization of delivery routes. However, these systems are expensive or difficult to acquire for small and medium enterprises (SMEs), these systems seek to optimize the delivery routes of the company, reducing costs, travel time and even carbon emissions emitted by the delivery vehicle [4]. That is why these small and medium-sized companies, such as the company used as an example in this work, have chosen to look for alternatives and design their own route optimization systems.

Within the state of the art we have found some works related to the problem of delivery route optimization, within these works we have proposed the development of some algorithms that allow us to face this problem, some works make use of some trajectory algorithms [5, 6], some others make use of population algorithms [7] or there are also works where neural networks are implemented [8-10], whose main objective is to obtain efficient routes for the delivery of products.

In this paper we will implement trajectory algorithms, such as the Greedy, the Iterative Local Search (ILS) algorithm, the stochastic algorithm based on random movements and the Annealing Simmulated algorithm to help an SME, dedicated to the sale of bulk products such as dry fruits, almonds, nuts and almonds, the stochastic algorithm based on random movements and the Annealing Simulated algorithm to help an SME, dedicated to the sale of bulk products such as dried fruits, almonds, nuts, etc., and the sale of spices such as bay leaves, cumin, cloves, etc., this company is located in Yuriria, Gto.

Path algorithms are known as metaheuristic algorithms and are characterized by the fact that they start from a point for the continuous improvement of a current solution by inspecting a neighborhood, and where the search finally ends when a maximum number of iterations is reached and a solution of acceptable quality is found, or a stalemate is detected in the exploration of the search space.

Metaheuristic algorithms were introduced in 1986 by Fred Glover [11]. The word metaheuristic derives from the composition of two Greek words heuristic which means "to find" and meta which can be defined as "beyond, at a higher level". That is why metaheuristic algorithms are referred to as a class of approximation methods designed for difficult combinatorial optimization problems. There are strategies of the same algorithms that allow to traverse the space of solutions of the problem, iteratively transforming an initial solution to these strategies are known as search metaheuristics, among which we can find the local search metaheuristics and global search metaheuristics.

In local search metaheuristics, we start from an initial solution and iteratively try to improve the initial solution until it is no longer possible to obtain better results. This type of search is based on an analysis of similar or fenced solutions, called neighboring solutions. The main disadvantage of these searches is that they do not exploit the universe and are trapped in a local optimum.

On the other hand, we have the global search metaheuristics which are designed with the purpose of escaping from local optima and not only perform a neighborhood exploration but an exploitation within the whole dataset.

The following are the metaheuristic algorithms that will be addressed in this work:

II. Algorithms

A. *Greedy*

Despite the progress that has been made today in the exact resolution of combinatorial optimization problems, heuristic algorithms such as Greedy continue to play an important role, because adequate solutions can be obtained with a short processing time compared to other methods used.

Greedy metaheuristic algorithms, characterized by using the available data of the problem to build a solution step by step, are one of the most widely used and best-known algorithms.

However, Greedy algorithms generally fail to find a globally optimal solution because they do not operate exhaustively on all the data, i.e. they perform local searches only. They may make decisions too early, and by default this prevents them from finding the best overall solution later on [12].

B. *Stochastic based on random movements (SRM)*

A stochastic algorithm based on random moves is a type of optimization algorithm that uses randomization and exploration components to search for solutions in a search space. These algorithms are applied to optimization problems, where one seeks to find the best possible solution within a set of possible solutions.

The key feature of a stochastic algorithm based on random moves is that it makes decisions using random or probabilistic elements. Often, these algorithms generate candidate solutions by perturbing current solutions or randomly exploring the search space so it belongs to a metaheuristic algorithm that performs local and global searches. Random moves may include random changes in a current solution or random selection of neighbors in the search space [13].

C. *Iterative Local Search (ILS)*

It is a metaheuristic optimization approach used to solve combinatorial optimization problems. This algorithm is based on local search, but incorporates a perturbation and intensified search component to escape from local optima and search for better quality solutions i.e., it explores globally as well.

The main idea behind ILS is to iteratively perform the following steps:

- Generation of an initial solution: It starts with an initial solution, which is often obtained randomly or by another heuristic approach. This initial solution may not be optimal.
- Local Search: Applies a local search algorithm to the current solution. Local search examines neighborhoods of the current solution to find a better solution within that neighborhood. If a better solution is found, it becomes the new current solution.
- Perturbation: After a local search phase, a perturbation is applied to the current solution. The perturbation modifies the current solution in some way, often randomly or by a specific perturbation process.
- Intensification: After the perturbation, a local search is performed again on the perturbed solution to try to improve it. This may result in a different and hopefully better-quality solution.
- Stopping criterion: The process of local search, perturbation and intensification is repeated for a specified number of iterations or until a predefined stopping criterion is met, such as a time limit or a maximum number of iterations without improvement.

The Iterative Local Search Algorithm is effective in solving combinatorial optimization problems where the optimal solutions may be in local optima, since perturbation and intensified search allow exploring different areas of the solution space. The iterative approach, by repeating this process, tends to converge to better quality solutions over time.

It is important to note that ILS performance is highly dependent on the appropriate choice of local search, perturbation, and other parameters specific to the problem being addressed [14].

D. Simulated Annealing (SA)

This algorithm consists of randomly generating a solution close to the current solution, which is accepted as good if it manages to reduce a certain cost function, or with a certain probability of acceptance. This probability of acceptance will be reduced according to the number of iterations and is also related to the degree of worsening of the cost, i.e., this algorithm can accept solutions that worsen the current solution, only that this acceptance will depend on a certain probability that depends on a parameter which is called temperature.

The process followed by this algorithm starts from an initial solution which is progressively transformed into another solution, which in turn improves by introducing small perturbations or changes (such as changing the value of a variable or exchanging the values of two variables).

If this change results in a "better" solution than the current solution, it is replaced with the new solution and the process continues until no further improvements are possible.

This means that the search results are at the local optimum, not necessarily global. One way to avoid this problem is to allow certain moves to be towards worse solutions. But if the research really leads to a good solution, these "escape" movements must be made in a controlled manner.

In this algorithm, this is done by controlling the frequency of output moves through a probability function that will reduce the probability of these moves to worse solutions as the development search progresses (and so, as expected, we are approaching a global optimum) [15].

1. Generic Elements: these are elements that do not have a direct dependence on the problem, although they must be fine-tuned for the specific problem to be solved.
2. Problem-dependent elements. These are the elements that directly define the problem being solved. They define a model of the problem and the structure of the solution space for it.

III. METHODOLOGY

The proposed methodology was implemented with four delivery routes where for each delivery point the latitude and longitude coordinates of each of the points were obtained with the help of Google maps, these routes belong to the days in which the packing plant distributes its products. Figure 1 shows the routes marked in Google Maps, each of the locations is marked with a specific color that belongs to the specific day in which the products are delivered.

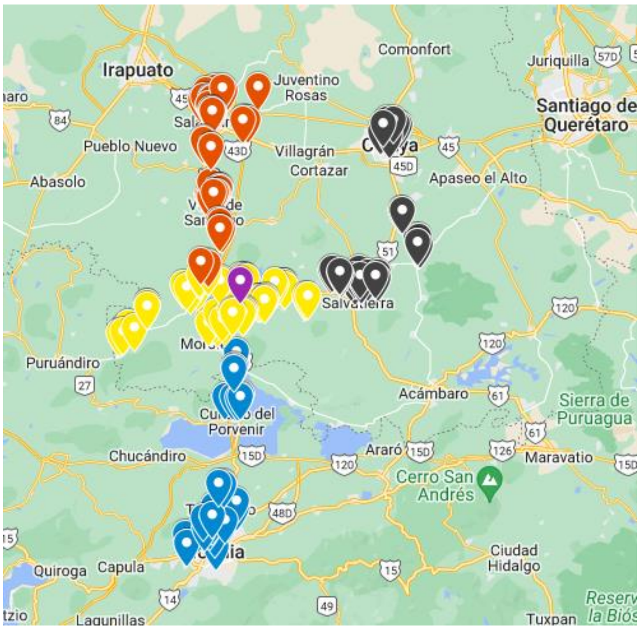

Fig. 1. *Delivery Route Locations.*

Figure 2 shows the distribution graph of these points, and shows the initial routes that the packer has, the routes are also identified with their respective color, depending on the day of delivery.

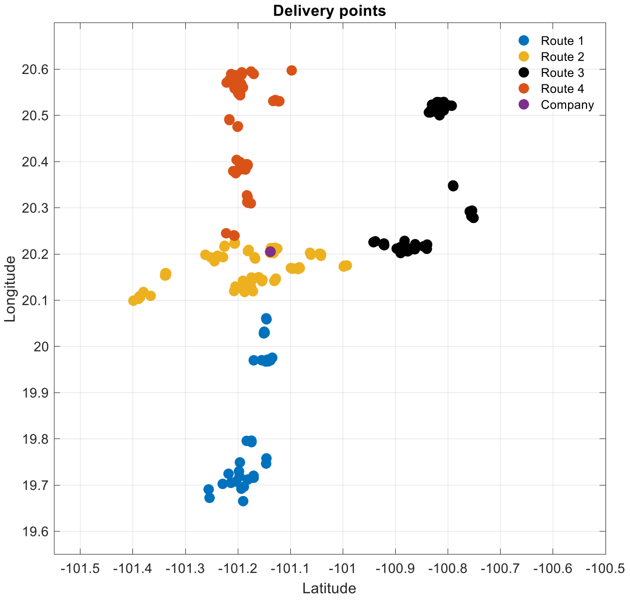

Fig. 2. Initial Delivery Routes (MATLAB).

Once the initial routes were available, the algorithms were implemented one by one, starting with the Greedy algorithm, followed by the SRM, then the ILS algorithm and finally the RS algorithm. Adapting the algorithms to solve other optimization problems involves making modifications to several key aspects of the algorithm to fit the characteristics and constraints of the new problem.

Some of the areas that need to be modified for a specific problem are:

- Objective Function and Solution Evaluation.
- Solution Representation.
- Neighborhood.
- Stopping Criteria.
- Adaptation and Exploration Function.
- Validation and Testing.

Adapting a metaheuristic algorithm to solve a new problem may require considerable design and fine-tuning effort, as each problem has its own unique characteristics. In the case of this work, we used the distance parameter of the routes calculated by the Haversine formula, which is used to calculate the total distance between two points knowing their longitude and latitude [16], the formula used is shown below:

$$d = R * \left[2atan2\left(\sqrt{a}, \sqrt{(1-a)}\right)\right] \quad (1)$$

Where,

$$a = sin^2\left(\frac{lat2 - lat1}{2}\right) + cos(lat1) * cos(lat2) * sin^2\left(\frac{lon2 - lon1}{2}\right),$$

R is the radius of the Earth which is $6371\ km$, $lat1$ and $lon1$ are latitude and longitude in radians of point 1 y $lat2$ y $lon2$ are the latitude and longitude in radians of point 2.

It is worth mentioning that each of the algorithms were implemented in MATLAB, and for each problem each of the algorithms was run ten times in order to observe the variability in the exploration (local search) and exploitation (global search) of the algorithms, as well as to improve the quality of the solutions obtained, reduce the influence of randomness and obtain a more complete understanding of the behavior of the algorithm in different instances of the problem. This allows more informed decisions to be made about the quality of the final solution and the consistency of the algorithm's performance.

IV. Results

Each of the algorithms were implemented route by route as mentioned above, the initial routes shown in Figure 2 were separated and evaluated. The results obtained for each of the routes are shown below.

In figures 3-6, the results of each of the routes can be seen, the initial distance and 3 of the algorithms implemented in each route are presented, however, the results obtained with these algorithms (ILS, SRM and SA) failed to find an optimal route to solve this problem, the 3 algorithms present routes with greater distance than the initial one, so it is not considered to optimize the route.

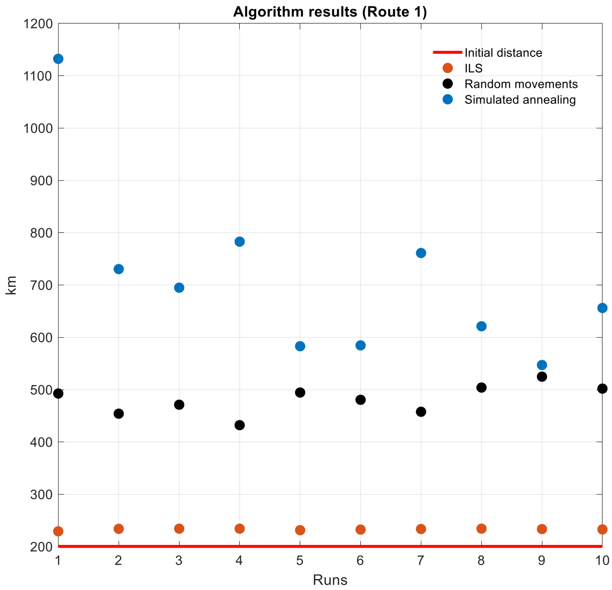

Fig. 3. Algorithms implemented Route 1.

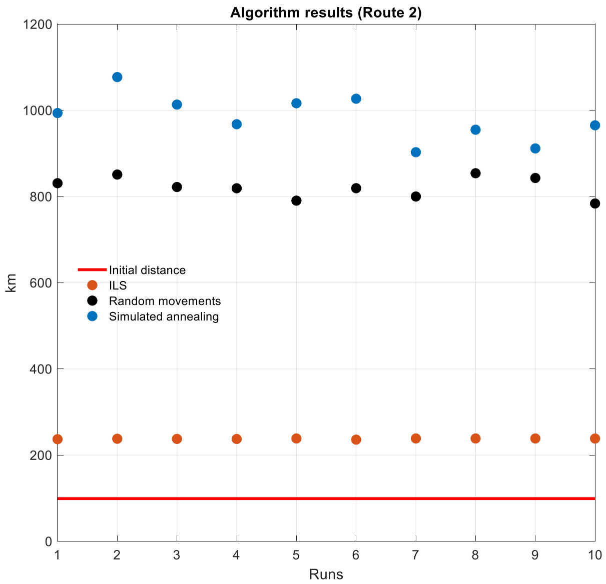

Fig. 4. Algorithms implemented Route 2.

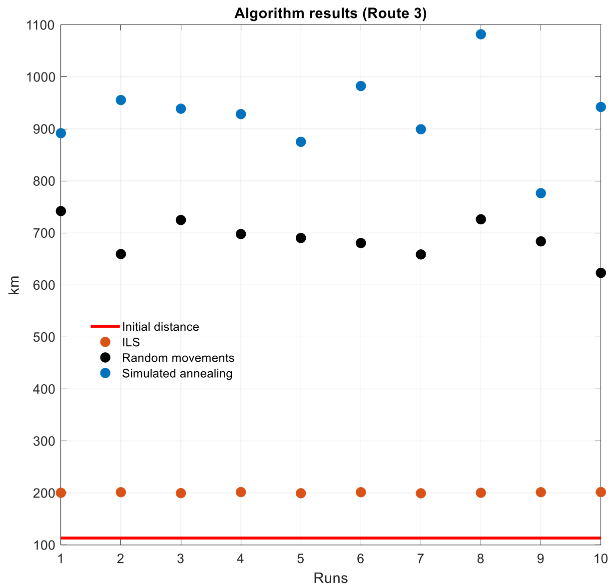

Fig. 5. Algorithms implemented Route 3.

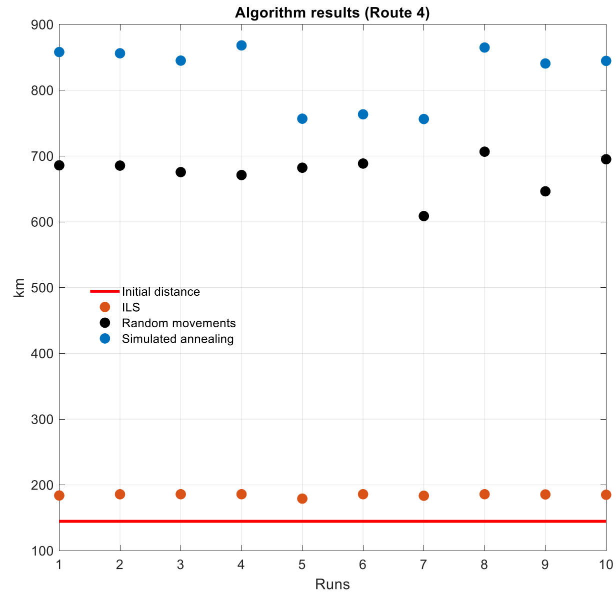

Fig. 6. Algorithms implemented Route 4.

The following table shows the best results obtained for the 3 algorithms (ILS, SRM and SA) compared to the initial routes that were already pre-established by the company.

TABLE I. RESULTS OF THE ALGORITHMS ILS, SRM AND SA.

Algorithm results (km)				
Routes	**Initial**	**ILS**	**SRM**	**SA**
1	200.51	229.38	432.25	547.23
2	99.23	237.08	783.93	902.87
3	113.35	201.47	623.22	776.41
4	144.72	179.22	608.72	756.26
Total	**557.81**	**847.15**	**2448.12**	**2982.77**

The results of the Greedy algorithm, unlike those shown above, do not change each time they are used due to its operation. The results obtained with the Greedy algorithm compared to the initial routes that were already pre-established by the company are shown below.

TABLE II. GREEDY ALGORITHM RESULTS.

Greedy algorithm results (km)		
Routes	**Initial**	**Greedy**
1	200.51	128.8
2	99.23	64.19
3	113.35	90.21
4	144.72	110.20
Total	**557.81**	**393.4**

According to the results obtained from Table 1 and 2, the best algorithm for this type of problem posed in this work is the Greedy algorithm, since with this algorithm the routes with the shortest possible distances were obtained.

Finally, after selecting the algorithm with the best result, the optimal route to be followed to achieve a total savings of **164.41 km** (calculated from the data in Table II) was determined for the company's four delivery routes. The following figure shows these routes obtained by applying the greedy algorithm.

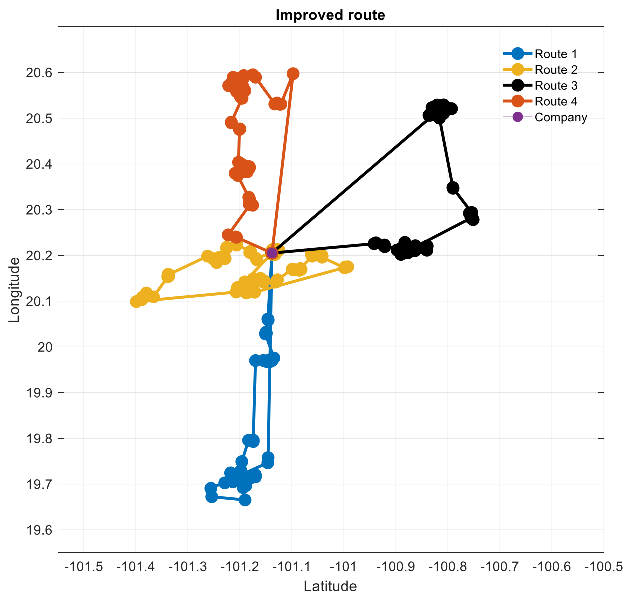

Fig. 7. Improved route.

V. CONCLUSIONS

The use of path algorithms can help with the optimization of some problems, however, although they help to solve these problems, it may not be the expected result, that is why several algorithms have to be tested to find the one that performs best for solving the specific problem, as an example we have the present research work, in which 3 of the 4 algorithms failed to find the optimal solution (fig. 3-6 and table I), however for other optimization problems, they could work adequately.

On the other hand, the application of metaheuristic algorithms in route optimization can be a powerful tool for entrepreneurs seeking to improve operational efficiency, reduce costs and offer more competitive and sustainable services.

These algorithms can not only be used for delivery route optimization, they can be implemented in several areas within a company, it is a matter of testing which algorithms work best for each specific problem.

Finally, according to the results in Table II, it is shown that the greedy algorithm for this specific problem works adequately, since it manages to significantly reduce the number of kilometers that must be traveled to achieve all the deliveries that are scheduled by the company. Thanks to the result of the greedy algorithm, a route map (fig. 7) could be made for each of the routes and the delivery points that are contemplated in each of them.

References

[1] E. A. B. Muñoz and J. H. C. J. E. h. y. l. m. Sotero, "Diseño de un modelo de optimización de rutas de transporte," no. 32, pp. 52-67, 2009.

[2] M. Chaveinte García, "Caso de estudio de aplicación de algoritmos genéticos para la optimización de rutas marítimas," 2023.

[3] M. I. Minga Quezada and Y. F. Zhiminaycela León, "Optimización de las rutas de recolección de los residuos sólidos urbanos del centro cantonal sígsig," 2019.

[4] N. R. J. I. d. Morales, "Modelo de optimización de programación de rutas para una empresa logística peruana usando herramientas FSMVRPTW," vol. 19, no. 2, pp. 118-123, 2016.

[5] J. C. Molina Gómez, "Diseño y aplicación de una herramienta para la optimización de rutas de vehículos con aspectos medioambientales," 2016.

[6] J. A. C. Benavides, "Aplicación de técnicas metaheurísticas a un problema de diseño de rutas de reparto en Galicia," in *II International Conference on Industrial Engineering and Industrial Management*, 2008, pp. 1425-1434.

[7] R. Baltazar, J. E. Vázquez, A. Rada, and C. Díaz, "Desarrollo de un sistema capaz de optimizar rutas de entrega utilizando algoritmos genéticos," ed: SINNCO, 2010.

[8] A. R. Zárate Avendaño, "Sistema de inventario para abastecimiento de medicamentos y su optimización con redes neuronales artificiales recurrentes aplicado a una Entidad Privada de Salud," 2011.

[9] C. R. Alcívar Molina, "Optimización de rutas para la gestión de pedidos y entregas aplicado a las empresas distribuidoras de productos," Quevedo: UTEQ, 2016.

[10] D. Fuertes Coiras, C. R. del-Blanco Adán, J. J. Navarro Corcuera, F. Jaureguizar Núñez, and N. García Santos, "Planificación de rutas para múltiples drones usando redes neuronales profundas Transformer."

[11] F. J. C. Glover and o. research, "Future paths for integer programming and links to artificial intelligence," vol. 13, no. 5, pp. 533-549, 1986.

[12] R. A. DeVore and V. N. J. A. i. c. M. Temlyakov, "Some remarks on greedy algorithms," vol. 5, pp. 173-187, 1996.

[13] C. Villate, C. A. Peña Cortes, and O. E. J. I. C. Gualdron Guerrero, "Algoritmo estocástico para la generación automática de trayectorias de un robot humanoide," vol. 14, no. 1, 2018.

[14] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search," in *Handbook of metaheuristics*: Springer, 2003, pp. 320-353.

[15] P. J. Van Laarhoven, E. H. Aarts, P. J. van Laarhoven, and E. H. Aarts, *Simulated annealing*. Springer, 1987.

[16] C. C. J. T. A. M. M. Robusto, "The cosine-haversine formula," vol. 64, no. 1, pp. 38-40, 1957.