UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Tecnura

https://revistas.udistrital.edu.co/ojs/index.php/Tecnura/issue/view/1275
DOI: https://doi.org/10.14483/22487638.21052

INVESTIGACIÓN

Solución al Problema del Agente Viajero usando un algoritmo de colonia de abejas

Solution to the TSP Problem Using an Algorithm Bee Colony

Jairo Alberto Villegas Flórez [1], Angel Augusto Agudelo Zapata [2] y John Andrés Muñoz Guevara [3]

Fecha de Recepción: 18 de julio de 2023 **Fecha de Aceptación**: 10 de abril de 2025

Cómo citar: J. A. Villegas Florez, A. A. Agudelo Zapata y J. A. Muñoz Guevara, ”Solución al Problema del Agente Viajero Usando un Algoritmo de Colonia de Abejas”, *Tecnura*, 29(84), 12-24. https://doi.org/10.14483/22487638.21052

Resumen

El Problema del Agente Viajero, o TSP por sus siglas en inglés (*Travelman Salesman Problem*), es un problema de optimización que consiste en hallar la ruta mínima para recorrer *n* ciudades, partiendo desde la ciudad de origen o nodo cero, visitando todas las ciudades y retornando nuevamente al punto o ciudad de origen. Este artículo expone una solución al problemautilizando la heurística exacta 2-opt, en combinación con una metaheurística de Colonia de Abejas, con el fin de obtener soluciones de buena calidad.

Objetivo: Usar la heurística 2-opt y la metaheurística algoritmo Colonia de Abejas para encontrar soluciones cercanas al óptimo, o incluso iguales, para el problema del agente viajero.

Métodología: Se emplearon las heurísticas MTZ y 2-opt para obtener soluciones iniciales de buena calidad, de tal forma que, al combinarlas con el algoritmo de Colonia de Abejas, estas estas pudieran mejorarse y tener valores cercanos o mejores que el óptimo reportado en la literatura.

Resultados: Entre los resultados obtenidos para las instancias de TSPLIB, donde se agrupan casos conocido conocidas como att48, berlin52, dantzig42, datasahara, oliver30, entre otras, la aplicación la metaheurística de colonia de abejas se llegó a resultados por encima del 10 por ciento del óptimo, valor que no era adecuado, posteriormente al combinarse con la heurística 2-opt se mejoró dicha solución alcanzándose valores cercanos al óptimo en instancias como oliver30, dantzig42 y datashara.

Conclusiones: Aunque el modelo MTZ es una heurística que, por su modelo matemático, ayuda a romper *subtours* y a encontrar buenas soluciones iniciales, presenta el inconveniente de que, a medida que se aumenta el número de ciudades, su complejidad polinómica hace difícil que se tenga un buen punto de partida para la solución final. Por

[1]Docente Titular. Facultad de Ingenierías. Universidad Tecnológica de Pereira.
Email: javi@utp.edu.co
[2]Docente Asistente. Facultad de Ingenierías. Universidad Tecnológica de Pereira.
Email: a3udeloz@utp.edu.co
[3]Docente Asociado. Facultad de Ciencias Empresariales. Universidad Tecnológica de Pereira.
Email: johandmunoz@utp.edu.co

esta razón, optamos por usar una heurística 2-opt que hace cambios en la misma ruta, logrando romper los *subtours* y encontrando soluciones de buena calidad mejoradas con la metaheurística Colonia de Abejas.

Palabras clave: 2-opt, Colonia de Abejas, Heurística, Metaheurística, Problema del Agente Viajero, TSP

Abstract

The Traveling Salesman Problem (TSP) is an optimization problem that seeks the shortest possible route for a tour of *n* cities, starting from an origin city (node zero), visiting each city, and returning to the city of departure. This article present a solution to the Traveling Salesman Problem is presented using an exact heuristic called 2-opt combined with a bee colony metaheuristic. This combination allows for finding high-quality solutions.

Objective: To use the 2-opt heuristic and the Bee Colony metaheuristic algorithm to find solutions close to or equal to the optimum for the Traveling Salesman Problem.

Methodology: The MTZ and 2-opt heuristics were employed to generate good-quality initial solutions, which were subsequently refined through the Bee Colony algorithm to improve performance and approach optimal results.

Results: Using the MTZ and 2-opt heuristics, find good-quality initial solutions so that, when combined with the Bee Colony algorithm, they can be further improved and yield values close to or better than the optimum reported in the literature.

Conclusions: Among the results obtained for the instances from TSPLIB, which include some well-known ones such as att48, berlin52, dantzig42, datasahara, oliver30, among others, when applying the Bee Colony metaheuristic, results above 10 percent of the optimum were obtained, which was not satisfactory. However, by combining it with the 2-opt heuristic, the solution was improved, reaching values close to the optimum in instances such as oliver30, dantzig42, and datashara

Keywords: 2-opt, Bee Colony, Heuristic, Metaheuristic, Traveling Salesman Problem

Introduction

The Traveling Salesman Problem (TSP) is a well-known combinatorial optimization problem in operations research literature, particularly in logistics applications. It involves finding the shortest possible route for a tour that visits a set of *n* cities, starting from a designated origin point, visiting each city exactly once, and returning to the starting point. The objective can be to minimize the distance or time required to complete the tour. The problem is often modeled as a directed graph, with cities as vertices and paths as edges, where each edge has a weight representing the distance or time between two cities.

There are various approaches to solving the TSP, including linear programming, exact techniques, heuristics, and metaheuristic methods for solving combinatorial problems. In this research study, the problem was addressed as follows: instances from the TSPLIB dataset were selected, including att48, berlin52, dantzig42, datasahara, oliver30, among others. Initially, the Bee Colony metaheuristic was applied, yielding results that were above 10 % of the optimum, which was deemed inadequate. To improve these solutions, the 2-opt heuristic was integrated, resulting in values close to the optimum for instances such as oliver30, dantzig42, and data-sahara.

The article is structured into several sections: Section 2 provides an explanation of the models used, Section 3 presents the results, Section 4 contains the conclusions, and a final section provides recommendations.

Methodology

The following methodology was developed to meet the objectives of this article:
The Traveling Salesman Problem (TSP) consist of a set of *n* cities, referred to as nodes, and a set of distances between each pair of cities, referred to as arcs. Consider the case where an agent needs to travel from city 1 to city 2; the distance will be denoted as d_{12}, which may not necessarily be equal to d_{21}. This distinction gives rise to two variants of the problem: symmetric TSP and asymmetric TSP (ATSP). The primary objective is to determine the shortest possible route that visits each city exactly once and then returns to the starting point (1) and (2).

Although the TSP can be mathematically formulated in a relatively straightforward way, its complexity grows with the number of cities. To express the TSP as a linear programming problem, we introduce a variable X_{ij}, which equals 1 if the arc from i to j is used, and 0 otherwise.

The mathematical formulation of the problem is presented below:

Classic TSP Mathematical Model:

$$X_{ij} = \begin{cases} 1 : \text{If city } j \text{ is visited after city } i \\ 0 : \text{In another case} \end{cases}$$

$$C_{ij} : \text{It is the cost associated with visiting city } j \text{ after visiting city } i$$

$$MinZ = \sum_{i=0} \sum_{j=0} C_{ij} X_{ij} \tag{1}$$

Equation (5) represents the objective function, which aims to minimize the distance traveled by the traveling agent. In addition to this objective function, constraints regarding the entry and exit of each city (4) should be formulated.

$$\sum_{i=0 i \neq j} X_{ij} = 1 \tag{2}$$

Equation (2) ensures that the agent arrives at one city at a time

$$\sum_{j=0 j\neq i} X_{ij} = 1 \tag{3}$$

Equation (3) ensures that the agent leaves a city each time.

However, since constraints (2) and (3) are not sufficient to optimize the problem, as sub-tours can still be created between sets of cities, the following constraint is introduced:

$$\sum X_{ij} >= 1 \forall i \in L, \forall j \in N \tag{4}$$

Linear programming problems like the Traveling Salesman Problem (TSP) are considered difficult to solve due to their combinatorial explosion, which makes it challenging to find the global optimum using exact techniques. Combinatorial problems can be divided into two types: P-type or polynomial problems, for which finding an optimal solution is relatively easy, and NP-type or non-polynomial problems, for which finding the global optimum is not as straightforward (1), (5) and (6). The Traveling Salesman Problem falls into the category of non-polynomial problems, making finding the global optimum a difficult task, especially as the number of cities increases.

The Application of the MTZ (Miller-Tucker-Zemlin)

The Traveling Salesman Problem can be solved using the MTZ model as described in (4) or as suggested by (14), as long as the number of cities is small. The formulation of the model is as follows:

Variables:

$$X_{ij} = \begin{cases} 1: \text{If go from city } i \text{ to city } j \\ 0: \text{In other case} \end{cases}$$

$$\mu_i : \text{cities number visited until city } i$$

Objective function

$$MinZ = \sum_{i=0} \sum_{j=0} C_{ij} X_{ij} \tag{5}$$

Constraints:

$$\sum_{i=0} X_{ij} = 1 : \text{Visit each city once} \tag{6}$$

$$\sum_{j=0} X_{ij} = 1 : \text{Leave each city once} \tag{7}$$

$$\mu_i + 1 <= \mu_j + 1 : \text{Breaks the subtours} \tag{8}$$

The constraint (8) break the sub-tours for all values of *i* from 2 to *n* and for all values of *j* from 2 to *n*, with X_{ij} belonging to the interval $[0, 1]$ and μ_{ij} greater than or equal to 1.

One of the main challenges of the Traveling Salesman Problem (TSP) is that, as the number of cities increases, finding an optimal solution becomes increasingly difficult. To address this issue, heuristics or metaheuristics can be implemented. In this study, we focus on the Bee Colony metaheuristic and the 2-opt heuristic, which are commonly used to solve the TSP.

The Bee Colony metaheuristic is inspired by the foraging behavior of bees. It involves creating a population of artificial bees that explore the search space and share information about promising solutions. Each bee performs a local search around its current position, while a global communication mechanism facilitates the exchange of information, enabling efficient exploration and exploitation of the search space. This metaheuristic aims to find good solutions by exploiting the search space efficiently.

On the other hand, the 2-opt heuristic is a local search algorithm that iteratively improves an initial solution. It works by iteratively swapping pairs of edges in the tour to create a new tour with a shorter total distance, as shown in (12). This process continues until no further improvements can be made. The 2-opt heuristic is known for its simplicity and effectiveness in improving the quality of solutions.

By combining these metaheuristic and heuristic approaches, it is possible to tackle the TSP and find high-quality solutions, even when dealing with a large number of cities.

The Bee Colony (BC) Metaheuristic

The Bee Colony algorithm belongs to a class of evolutionary algorithms inspired by the the foraging behavior of bees in their search for nectar sources around their hive (7). This class of metaheuristics has only recently begun to receive significant attention, as previously another metaheuristic with the same food-search principle—the Ant Colony algorithm, or Ant Colony Optimization—was more widely applied (6,9). Consequently, different variants of bee-inspired algorithms with various names have been proposed to solve combinatorial problems. However, in all of them, some common search strategies are applied, meaning that complete or partial solutions are considered food sources, and groups of bees attempt to exploit these food sources with the hope of finding high-quality nectar (i.e., high-quality solutions) for the hive. Communication among bees is modeled through the waggle dance, which conveys information about the search space and food sources.

In the algorithm, bees are divided into three types: employed bees, onlookers, and scouts. Employed bees are responsible for exploiting the available food sources and gathering the required information. They also share the information with the onlookers, and the onlookers select the existing food sources to be further explored. As described in (11), when an employed bee abandons a food source, it becomes a scout and starts searching for a new food source near the hive. Abandonment occurs when the quality of the food source fails to improve after performing a maximum allowed number of iterations.

The ABC algorithm is iterative in nature. It begins by generating random solutions, treated as food sources, and assigning each employed bee to a food source. Then, during each iteration, each employed bee finds a new food source near its originally assigned (or old) food source (using a neighborhood operator). The amount of nectar (fitness) of the new food source is then evaluated. If the new food source has greater fitness than the previous one, it replaces the old source. Once all employed bees have finished with the previous exploitation process, they share the nectar information of the food sources with the onlookers. Each onlooker selects a food source according to the traditional roulette wheel selection method. After that, each onlooker finds a food source near its selected food source (using a local operator) and calculates the amount of nectar of the neighboring food source. Then, for each previous food source, the best food source among all the food sources near the previous one is determined. The employed bee associated with the previous food source is assigned to the best food source and abandons the old one if the best food source is better than the previous food source. As noted in (10), an employed bee may also abandon a food source if its quality does not improve for a predetermined number of successive iterations (a predefined number). That employed bee becomes a scout and searches for a new random food source. After the scout finds a new food source, it becomes an employed bee again. After each employed bee is assigned to a food source, another iteration of the ABC algorithm begins. The entire process is repeated until a termination condition is met.

The steps of the bee colony algorithm are as follows:

1. Randomly generate a set of solutions as initial food sources $x_i, i = 1, ..., \tau$. Assign each employed bee to a food source

2. Evaluate the fitness $f(x_i)$ of each of the food sources $x_i, i = 1, ..., \tau$
3. Set $v = 0$ and $l_1 = l_2 = ... = l\tau$
4. Repeat

   a. For each food source x_i

      i. Apply a neighborhood operator to x, resulting in x_i=x'

      ii. If $f(x') > f(x_i)$ then replace x_i with x' and $l_i = 0$ else $l_i = l_i + 1$

   b. Set $G_i = 0$, $i = 0, 1, 2, ..., \tau$. Where G_i it is the set of neighborhood solutions of food source i.

   c. For each observer

      i. Select a food source x_i using the roulette wheel selection method

      ii. Apply a neighborhood operator to x_i in x'_i

      iii. $G_i = G_i U x'_i$

   d. For each food source

      i. Set $x'i \in argmax_{\sigma \in G_i}$

      ii. If $f(x_i) < f(x')$ then replace x_i with x' y $l_i = 0$, else $l_i = l_i + 1$

   e. For each food source x_i

      i. if $l_i = limit$ then replace x_i with a random solution

   f. $v = v + 1$

   g Until $v = maxiterations$

2-Opt Heuristic

For many years, various solution methods have been proposed for NP-Hard problems that arise from real-life situations. Due to the nature of these problems, it is impossible to list all possible solutions due to the combinatorial explosion, and exact techniques are unable to solve them (4).

But what is an heuristic? One definition, provided in (13), is as follows: .A simple procedure are show in (13), that is an often based on common sense, that is supposed to offer a good solution (though not necessarily the optimal one) to highly complex problems in an easy and fast manner" (5). Heuristics are used when exact solution methods do not exist or when an exact method exists but consumes a lot of time and memory. The problem addressed in this article belongs to this category, as no comprehensive solution for all the flaws is feasible, necessitating the search for an alternative solution method. Therefore, heuristic and metaheuristic techniques are employed (6). One of these techniques is the 2-opt operator (2), which involves swapping two customers in the route to eliminate overlaps or crossings that occur in the initial solution, thereby improving the solution and finding better results to reduce the gap. The following is a pseudo-code representation of the implemented heuristic:

- Within the tour, select a city Y_i
- Compare arcs in the route X_{ij} in order to exchange nodes while satisfying constraints (2) and (3)
- The exchange is only possible if, upon comparing the arcs, the route improves in terms of the actual travel, meaning that the distance traveled is minimized

As mentioned before, vehicle routing problems fall into the category of NP-Hard (5) and (6), which means that there is no optimal solution in polynomial time. Therefore, it is necessary to resort to heuristic and metaheuristic solution techniques, such as the ones mentioned in this article

Results

The implementation and execution of this algorithm were performed on a computer equipped with an Intel(R) Core(TM) i7-6700 CPU @3.40 GHz, running a 64-bit Windows 7 operating system and 8 GB of RAM.

The software environment used was Python version 2.7. With this program, an algorithm combining a swap heuristic called 2-opt and an ABC (Ant Bee Colony) metaheuristic was executed.

As a working framework, the algorithm was tested using instances from the TSPLIB library. Table 1 shows the nine instances run when applying the Ant Bee Colony algorithm combined with the 2-opt heuristic and his best known result.

Table 2 show us the result obtained using only the Ant Bee Colony and we can observe how the 2-opt heuristic help to improvement the result in some cases found by the ABC metaheuristic.

Tables 3 and 4 show the results obtained when running the ant bee colony combine with the algorithm with the 2-opt heuristic. It can be observed that, in the Oliver 30 instance (Table 3), the executed algorithm was able to reach the minimum reported in the literature.

Table 1. Instances Run

Number	Instances	Total of Nodes	Best Result
1	Oliver 30	30	424
2	Dantzig 42	42	699
3	Datasahara 29	29	27603
4	Berlin 52	52	7542
5	Att 48	48	10628
6	Eil 51	51	426
7	St 70	70	675
8	Kroc 100	100	20749
9	Pr 144	144	58537

Table 2. Bee Colony Results

Bee Colony Metaheuristic				
Instance	**Results With Bee Colony Metaheuristic**			
	Best Result	**Iteración 1**	**Iteración 2**	**Iteración 3**
Oliver 30	424	447 (5,42 %)	460,8 (8,68 %)	451,5 (6,48 %)
Dantzig 42	699	748,7 (7,11 %)	705,8 (0,97 %)	743,0 (6,29 %)
Datasahara 29	27603	27748,7 (0,53 %)	27748,7(0,53 %)	27601,2 (-0,01 %)
Berlin 52	7542	8362,1 (10,87 %)	8040,7(6,61 %)	8332,6 (10,48 %)
Att 48	10628	35963,8 (238,39 %)	35806,8 (236,91 %)	37686,5 (254,6 %)
Eil 51	426	477,4 (12,06 %)	471,3 (10,63 %)	459,6 (7,88 %)
St 70	675	781,8 (15,83 %)	812,7 (20,40 %)	823,7 (22,03 %)
Kroc 100	20749	28011,3 (35 %)	28233,5 (36,07 %)	26798,3 (29,15 %)
Pr 144	58537	106441,8 (81,84 %)	100654,2 (71,95 %)	116573,0 (99,14 %)

In addition, in Tables 3 and 4 , we can observe how the dantzig 42 and datasahara 29 instances achieved values below the established minimum. In the case of dantzig 42, the value is -1.431 %, and in the case of datasahara, the value is -0.007 %, in the other instances like the St 70 and Pr 144 the combination of the both algorithms show good results nearly to the 3.5 % in the GAP

To conclude, it can be observed that the algorithm 2-0pt Heuristic performs well for instances with more than 50 nodes. In the table 4, specifically for the Eil 51, Att 48 and Pr 144 instances, respectively, the algorithm achieved values around 1 % of the solution found in the literature. More precisely, it reached 0,313 % for Att 48, 1.643 % for Eil 51 and 1.095 % for Pr144.

Table 3. Instances Result (A)

INSTANCES RESULTS			
Instance	**Statistical**		
Oliver30	Minimum	Maximum	Average
	424,000	432,838	432,200
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	0,00 %	6,604 %	1,934 %
Dantzig42	Minimum	Maximum	Average
	689,000	715,000	703,625
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	-1,431 %	2,289 %	0,662 %
Datasahara	Minimum	Maximum	Average
	27601,170	27748,710	27630,680
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	-0,007 %	0,528 %	0,100 %
Berlin52	Minimum	Maximum	Average
	7934,000	8445,000	8116,000
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	5,198 %	11,973 %	7,611 %

Source: Authors.

Conclusions and Recommendations

- Although the MTZ model is a heuristic that, with its mathematical model, helps break sub-tours and find good initial solutions, it has the disadvantage that as the number of cities increases, its polynomial complexity makes it difficult to have a good starting point for the final solution. Because of this, we chose to use a 2-opt heuristic that makes changes within the same route, effectively breaking sub-tours and finding improved high-quality solutions with the bee colony metaheuristic.

- The combination of heuristics with metaheuristics enhances the ability to reach solutions much closer to the optimum, as was the case for us, where we were able to reach the best known optimal solution in some cases.

- The application of the 2-opt heuristic as a starting point is also a very effective tool, since it combines multiple options to deliver high-quality solutions. This provides a better search space for the Bee Colony metaheuristic, leading us to find good-quality solutions and, in some cases, solutions close to the optimum.

- As future work, it is suggested to combine a different heuristic—such as the savings algorithm, a 3-opt, a granular search, etc.—with the proposed ABC model to potentially achieve optimal solutions. Comparing these results with the findings of this research would be beneficial.

Table 4. Instances Result (B)

INSTANCES RESULTS			
Instance	**Statistical**		
Att48	Minimum	Maximum	Average
	33628,000	35589,000	34734,330
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	0,313 %	6,163 %	3,613 %
Eil51	Minimum	Maximum	Average
	433,000	451,000	441,286
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	1,643 %	5,869 %	3,588 %
St70	Minimum	Maximum	Average
	696,000	734,000	721,000
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	3,111 %	8,741 %	6,815 %
Kroc100	Minimum	Maximum	Average
	21473,000	24307,000	22652,170
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	3,479 %	17,137 %	9,162 %
Pr144	Minimum	Maximum	Average
	59178,000	60781,000	59769,860
	GAP (%)		
	GAP (Minimum)	Gap (Maximum)	GAP (Average)
	1,095 %	3,833 %	2,106 %

Source Authors.

- By incorporating alternative heuristics, we can explore different approaches to solving the problem and potentially improve the quality of the solutions obtained. This comparative analysis would provide insights into the effectiveness of different heuristic combinations and their synergy with the ABC algorithm.
- Additionally, it would be valuable to assess the performance of these hybrid approaches on various benchmark problems and real-world instances to evaluate their scalability, robustness, and efficiency. This comprehensive evaluation would contribute to a better understanding of the strengths and limitations of different heuristic combinations in conjunction with the Bee Colony algorithm, aiding in the development of more powerful optimization techniques for addressing large-scale combinatorial problems
- Another recommended work is to combine the same 2-opt heuristic with different exhaustive search algorithms, such as the ant colony algorithm and the bee colony algorithm. It is also suggested to perform comparisons between evolutionary algorithms and search algorithms.
- Combining the 2-opt heuristic with various exhaustive search algorithms would provide an opportunity to explore different approaches and techniques for solving the problem. For instance, one could investigate strategies that incorporate the 2-opt heuristic as a local improvement phase within the ant colony or ant bee colony algorithms. This integration could leverage the strengths of both approaches to obtain high-quality solutions

References

[1] Hincapie, Isaza,. Ricardo., Rios, Porras, Carlos.,& Gallego, Ramon. Técnicas Heurísticas Aplicadas al Cartero Viajante (TSP). *Revista Scientia et Technica, Nro 24, Marzo de 2004*. https://revistas.utp.edu.co/index.php/revistaciencia/article/view/7279

[2] Heiko, Röglin., Vöcking, Berthold. (2014). Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP. *Algorithmica*, Nro 68. Enero 1 de 2014. Paginas 190 - 264. https://doi.org/10.1007/s00453-013-9801-4

[3] Velasquez, Escobar, J. (2013). *Sistemas de Transporte: Modelado y Optimización*.

[4] Velasquez, Escobar, J., Linfati, Rodrigo., & Adarme, Jaimes, Wilson. (2015). Problema de Localización y Ruteo con Restricciones de Capacidad: Revisión de la Literatura. *Revista Facultad de Ingeniería*, Vol 24. Nro 39). https://www.redalyc.org/articulo.oa?id=413940776008

[5] Rocha, L. González, C., & Orjuela, J. (2011). Una revisión al estado del arte del problema de ruteo de vehículos: Evolución histórica y Métodos de Solución. *Revista Facultad de Ingeniería*. Volumen 16. Número 2. Paginas 35 - 55. https://doi.org/10.1007/s10898-007-9149-x

[6] Karaboga, Dervis & Barhiye, Basturk.(2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization, Volumen 39*, pp 459-471. https://doi.org/10.1007/s10898-007-9149-x

[7] Villegas, Jairo A., Grisales, Carlos J., & Gatica, Gustavo. (2017). Una aplicación del método MTZ a la solución del problema del agente viajero. *Revista Scientia Et Technica, Volumen 22, Número 4, Diciembre 2017*, pag 341 - 344. https://doi.org/10.22517/23447214.12751

[8] Pavez, Alejandro., Almonacid, Héctor. (2002). Un Algoritmo ACS con Selección Dinámica de Movimiento y Operador 2-Opt. *Ingeniería Informática, Número 8*. http://inf.udec.cl/~revista/ediciones/edicion8/Acs.pdf

[9] Cisneros Perez, M.S., Sanchez Solis, J. P., Rivera Zarate, G., Garcia Jimenez, V. & Florencia Juarez, R. (2020).Algoritmo de colonia de hormigas para abordar el problema de order picking. *Memorias del Congreso Internacional de Investigación Academia Journals Hidalgo 2020, Vol. 12, No. 7, 2020*, 387-392. http://cathi.uacj.mx/bitstream/handle/20.500.11961/16411/Memoria%20ACO.pdf?sequence=1&isAllowed=y

[10] Yeh Wei-Chang, Su Jack C.P, Hsieh Sung-Jung, Chih Mingchang & Liu Sin-Long (2011).Approximate Reliability Function Based on Wavelet Latin Hypercube Sampling and Bee Recurrent Neural Network. *IEEE Transactions on Reliability 2011, Vol. 60, No.2*, 404 - 414. https://doi.org/10.1109/TR.2011.2134190

[11] Augusto, Luiz C. Andrade G & Cunha Claudio B. (2015). An ABC heuristic for optimizing moveable ambulance station location and vehicle repositioning for the city of São Paulo. *International Transactions In Operational Research 2015, Vol. 22, No.3*, 473 - 501. https://doi.org/10.1111/itor.12160

[12] Ihsan, Kilic Kemal & Mostarda, Leonado. (2021). Novel Concave Hull-Based Heuristic Algorithm For TSP. *Operations Research Forum 2022, Vol. 3, No.25*, https://doi.org/10.1007/s43069-022-00137-9

[13] Borges, Ana Carolina. Padilha, Reinaldo. Rangel, Arthur & Iano, Yuzo. (2022). The fundamentals and potential of heuristics and metaheuristics for multiobjective combinatorial optimization problems and solution methods. *Academic Press*, 9 - 29. https://doi.org/10.1016/B978-0-12-823799-1.00002-4

[14] Moroyoqui Olan, Marcos Alberto and Orozco Rosas, Ulises & Picos, Kenia. Implementación de un Algoritmo Genético Modificado para la solución al Problema del Agente Viajero. Volumen 8. Número 17. Año 2022. https://repositorio.cetys.mx/handle/60000/1426