Webology (ISSN: 1735-188X)
Volume 19, Number 6, 2022

An Understanding Of The Traveling Salesman Problem With Heuristic Techniques In Postgraduate Students

Javier E. Sierra[1], Alejandro S. Guerrero Hernández[1] and María F. Sierra[2]

[1] Faculty of Engineering, Universidad de Sucre, Sincelejo, Sucre- Colombia

[2] Education Faculty, Universidad de Sucre, Sincelejo, Sucre- Colombia

* Correspondence: Author: Alejandro S. Guerrero Hernández

Abstract
The problem of a traveling salesman remains of great importance in using exact optimization techniques and heuristics. Therefore, this paper discusses some heuristic algorithms in MATLAB to select the best candidate and obtain results, such as the number of iterations, time, and target function. In addition, it serves as a methodology basis for graduate students of the University of Sucre.

Keywords: Traveling Salesman, Genetic algorithm, Ant Colony, Annealing Simulated.

I. INTRODUCTION

Optimization aims to find the best solution that maximizes or minimizes the target function of a process, design, or system. These algorithms can be classified into deterministic algorithms and heuristic algorithms. Deterministic algorithms are based on classical mathematical models and nature-inspired heuristic algorithms (Yang, 2020).

The Traveling Salesman's problem has been modified recently to reduce CO2 pollution and energy consumption on the vehicular route. The proposed solutions are more recently based on heuristic algorithms such as the work done in (Albayrak & Allahverdi, 2011; Contreras-Bolton & Parada, 2015; Deep & Mebrahtu, 2011). In the following sections, we focus the work on three heuristics: Genetic algorithm, Annealing Simulated, and Ant Colony on the classic routing problem and determine the best solution as a clear example of logistics in optimization disciplines and operation research in undergraduate and graduate classrooms**.** In section II, we will discuss the objective function and constraints of the traveling salesman. Section III shows the results obtained with each heuristic developed in the MATLAB software and, finally, the conclusions.

II. PROPOSED SYSTEM

Traveling Salesman's Problem

Suppose a clerk has to visit different cities, starting and ending his journey in the first city. The order of visits to the cities does not matter; each can go directly to any other. In short, the traveling salesman's problem is discovering the route that minimizes the total trip (Gutin & Punnen, 2007).

$$\min Z = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \quad (1)$$

Subject to restrictions

$$x_i = 0, \; i = 0,1,2,3, \ldots, n \; (2)$$
$$x_{ij} \epsilon \{0,1\}, \quad i, j = 1,2,3, \ldots, n \; (3)$$

$$\sum_{i=1}^{n} x_{ij} = 1, \; j = 1, \ldots, n \; (4)$$

$$\sum_{j=1}^{n} x_{ij} = 1, \; i = 1, \ldots, n \; (5)$$
$$\sum_{i \epsilon S} \sum_{i \epsilon S} x_{ij} \leq |S| - 1, S \subset V: S \neq \phi \; (6)$$

Equation (1) represents the objective function of the problem. It consists of minimizing the cost for the traveling salesman to pass through all the n distinct points once and return to the initial starting point. The problem's constraints are given by equations (2) up to (5). Equation (2) restricts that it is impossible to go from a city to itself. In contrast, equation (3) represents a binary variable x_{ij} of displacement confirmation, such that if x_{ij}=1, the clerk goes directly from city j to city i on the presented route—otherwise, x_{ij}=0. Equations (4) and (5) determine that the clerk must pass once at each point, either on arrival or at departure, and to the next point until the entire route is carried out and return to the starting point. Equation 6 prevents cycling smaller than n on the route, i.e., only Hamiltonian cycles are produced.

III. RESULTS

Annealing Simulated

In this algorithm, the first phase of the search, the tendency to improve the result is low. However, this behavior is modified, along with the temperature, causing the result to converge at the end of the algorithm processing to a minimum place. The choice of a worse state depends on two variables: the cost difference between the current state and the next and the current temperature. The more significant the difference between the current state and the next, the lower the probability it will be selected. The higher the temperature, the greater the probability that a worse state will be chosen (Sindhuja et al., 2018).

The parameters used were.

- Starting Temperature:8000
- Temperature reduction rate: 0.98

Table 1 shows that the higher the maximum number of iterations, the objective function (Cost) decreases.

Table 1. Simulated Annealing result

Maximum iterations	**Objective Function**	**Final Temperature**	**Time (Seg)**
6000	568	2,27E-33	2
7000	526	3,82E-42	2
8000	494	6,44E-51	2
10000	493	1,82E-67	3
20000	443	3,32E-156	6
30000	448	6,06E-244	8
40000	431	1,2e-322	11
50000	430	1,2e-322	13
60000	444	1,2e-322	15
70000	414	1,2e-322	17
80000	434	1,2e-322	22

Genetic algorithm.

We can define genetic algorithms as research systems based on selection mechanisms and natural genetics. They consider the principles of survival of the fittest and compatibility between string structures, which contain random information, subject to processes of exchange, reproduction, and mutation (Golberg, 1989).

The goal of the Genetic Algorithm is to maximize the return of candidate solutions in the population against a cost function of the problem domain. Genetic algorithms can be described: the first step of initializing a population of individuals; is the second stage of assessment and selection according to the characteristics of each individual and their aptitude for specific evaluation parameters, reproduction to generate new offspring, and random mutation of new offspring (Jones, 2005).

The parameters used were.

- Population size: 101 individuals.

- Iterations: 100 - 500.

Table 2 shows that the higher the maximum number of iterations, the objective function (Cost) decreases.

Table 2. Genetic algorithm results

Generation	Time (seg)	Objective Function
100	28	723
150	51,6	474
200	59,88	237
250	72,3	166
300	98,13	162
350	93,1	172
400	110,6	168
500	147,6	182

Ant Colony.

Ant colony optimization (ACO) algorithms, as the name implies, were inspired by ants, mainly in their behavior when looking for food, but also regarding the organization of work and cooperation. The strategy aims to exploit historical and heuristic information to build candidate solutions and duplicate the information learned from building solutions in history. Solutions are constructed one discrete piece at a time, probabilistically, step by step. The probability of selecting a component is determined by the heuristic contribution of the component to the total cost of the solution and the quality of the solutions historically known to have included the component. The history is updated proportionally to the quality of the most well-known solution and is proportional to the use of discrete components of the solution(Dorigo & Blum, 2005). Table 3 shows that the higher the maximum number of iterations, the more objective function.

The parameters used were.

- The number of ants: 10.
- Pheromone reduction factor: 0.9.
- The number of iterations: 50-600.

Table 3. Results Ant colonies

ITERATIONS	TIME (seg)	Objective Function
50	15	430
100	31,0	430
200	60,0	420
300	98	410
400	122	395
500	162,0	432
600	330,0	416

Metaheuristics are compared, and the following analyses are obtained:

- Figure 1 shows the smallest objective function value obtained with the genetic algorithm.
- Figure 2 shows the shortest time of achievement with Annealing Simulated.
- Figure 3 shows that the genetic algorithm reached the maximum number of iterations.

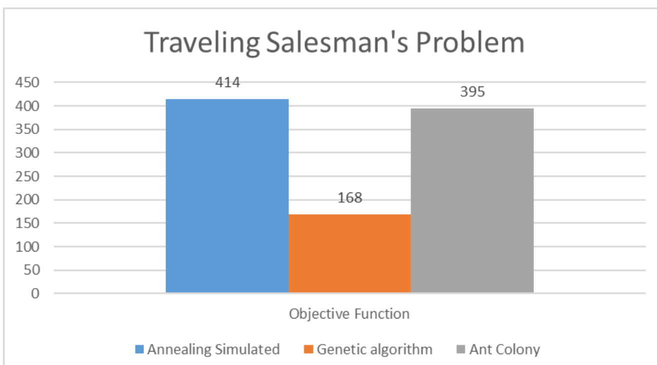

Figure 1. Objective function

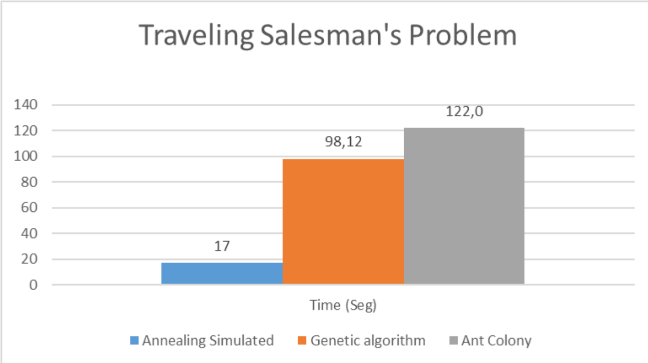

Figure 2. Time heuristic.

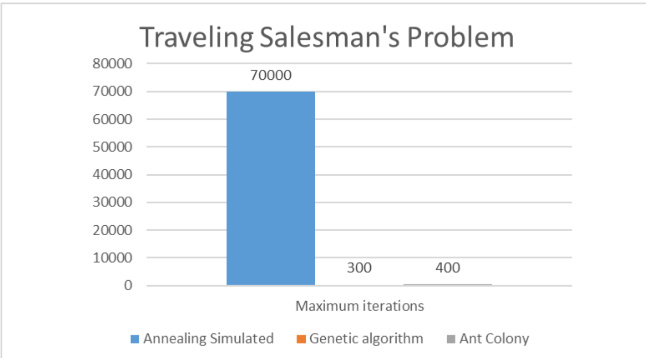

Figure 3. Maximum iteration

IV. CONCLUSIONS

With the application of metaheuristic techniques to optimize the street vendor cost, we can obtain the following conclusions:

The heuristic (Simulated Annealing) is designed to find reasonable approximate solutions to complex combinatorial problems that available optimization algorithms cannot solve. The advantage of heuristics is that they usually determine (reasonable) solutions quickly, using simple solution rules. The downside is that the quality of the solution (relative to the optimum) is often unknown.

The metaheuristic (Genetic Algorithm, Ant Colony) is primarily designed to escape entrapment in the local optimum, allowing lower moves if necessary. The flexibility added to the search is expected to lead to a better solution.

The computational cost of simulators is one of the disadvantages when applying them, especially in optimization problems. However, it presents acceptable results with the traveling salesman.

REFERENCES

Albayrak, M., & Allahverdi, N. (2011). Develop a new mutation operator to solve the traveling salesman problem using genetic algorithms. Expert Systems with Applications, 38(3). https://doi.org/10.1016/j.eswa.2010.07.006

Contreras-Bolton, C., & Parada, V. (2015). Automatic combination of operators in a genetic algorithm to solve the traveling salesman problem. PLoS ONE, 10(9). https://doi.org/10.1371/journal.pone.0137724

Deep, K., & Mebrahtu, H. (2011). Hadush Combined Mutation Operators of Genetic Algorithm for the Travelling Salesman problem. México © International Journal of Combinatorial Optimization Problems and Informatics, 2(3).

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. Theoretical Computer Science, 344(2–3). https://doi.org/10.1016/j.tcs.2005.05.020

Golberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Choice Reviews Online, 27(02), 36. https://doi.org/10.5860/choice.27-0936

Gutin, G., & Punnen, A. (2007). The Traveling Salesman Problem and its Variations. … of Combinatorial Optimization: Problems and ….

Jones, T. M. (2005). AI Application Programming (Charles River Media Programming). Charles River Media.

Sindhuja, P., Ramamoorthy, P., & Suresh Kumar, M. (2018). A Brief Survey on Nature Inspired Algorithms: Clever Algorithms for Optimization. Asian Journal of Computer Science and Technology, 7(1). https://doi.org/10.51983/ajcst-2018.7.1.1835

Yang, X. (2020). Nature-inspired optimization algorithms. Academic Press.