BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 69 (73), Numărul 3, 2023
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ
DOI:10.2478/bipie-2023-0017

COMPARISON BETWEEN AN EXACT AND A HEURISTIC-BASED TRAVELING SALESMAN PROBLEM WITH TIME WINDOW CONSTRAINTS

BY

MIHAELA-ALEXANDRA BARB-CIORBEA*

“Gheorghe Asachi” Technical University of Iași,
Faculty of Automatic Control and Computer Engineering, Iași, Romania

Received: June 30, 2024
Accepted for publication: August 30, 2024

Abstract. This work aims to compare two distinct approaches for solving a Travelling Salesman Problem with time window constraints. Given an environment with a fixed number of cities (points of interest), a robot must determine a route such that each city is visited in an imposed time interval. Both of the examined techniques have the objective of identifying the path with the lowest cost in terms of the distance traveled.

The initial approach employs an exact method by defining the requirements as a mixed integer linear programming (MILP) optimization problem.

The second method involves a meta-heuristic approach, using an ant colony procedure to solve the optimization problem.

Besides qualitative information, the performed quantitative comparison relies on multiple numerical simulations performed in a MATLAB environment. We thus highlight the advantages and disadvantages of both methods, by taking into consideration criteria as the simulation time and the relative difference between the obtained costs versus the number of cities.

Keywords: Travelling Salesman, optimization, ant colony, graph, time windows.

---

*Corresponding author; *e-mail*: mihaela-alexandra.barb-ciorbea@academic.tuiasi.ro

1. Introduction

The Travelling Salesman Problem (TSP) is a well-known problem that is classified as NP-hard. It has been extensively studied in operations research and computer science. The TSP has practical applications in various industries, including logistics and transportation, manufacturing and production planning, network design and optimization, as well as robotics and automation, (Larni-Fooeik *et al*., 2024; Chandra and Natalia, 2023).

In the context of logistics and transportation, organizations aim to optimize their delivery routes to minimize expenses. For example, a delivery company seeks to discover the most efficient route that visits all of its clients while consuming the least amount of gasoline or taking the shortest delivery time. In the manufacturing industry, it is common for plants to seek the most efficient path to visit all the machines requiring maintenance in order to reduce operational interruptions and optimize efficiency. When it comes to network design and optimization, users must choose the most efficient way to connect their network nodes in order to decrease signal loss and enhance network performance.

In the classical Travelling Salesman Problem (TSP), a salesman is required to visit each city in a given set exactly once and then return to the beginning city, while minimizing the total distance travelled. Applying the Travelling Salesman Problem (TSP) to real-world applications is not always feasible due to various constraints. Additionally, there may be a need for distinct variants of TSP in these real-life circumstances. As a result, other versions of the Travelling Salesman Problem (TSP) have been suggested to address various limitations in different applications, including Profit-based (Khanra *et al*., 2015), Time Windows-based (Dumas *et al*.,1995; Fontaine *et al*., 2023), Maximal Based (Aguayo *et al*., 2018), and Selective Travelling Salesman Problem (Laporte and Martello, 1990). The goal of profit-based variations of the Travelling Salesman Problem (TSP) is to find the most efficient route that maximizes profit while minimizing travel costs. It is not necessary to visit every vertex, and each vertex has a predetermined profit value. The purpose of the Selective Travelling Salesman Problem is to discover a trip that maximizes the profit collected while ensuring that the travel cost does not exceed a certain value.

Several techniques for addressing this type of problem and its variations have been popular due to their ability to quickly observe the computation results. However, classical algorithms are primarily well-suited for global path planning in deterministic environments. They can also be utilized in uncertain areas when combined with optimization approaches, although their effectiveness may vary. It is important to realize that the research indicates the possibility of encountering high computational costs and low computational efficiency. Some of these techniques are presented in Chapter 2.

2. Related work

When solving the Traveling Salesman Problem with time windows variant, the literature has identified the following main categories: precise procedures and heuristic methods, (Cheikhrouhou, 2021).

In (Dantzig *et al*., 1954), the author proposed an early deterministic solution to TSPTW by using linear programming (LP) relaxation. This approach involves solving the integer formulation and continuously adding a well-selected linear inequality to the list of constraints. The TSPTW problem can be solved exactly using a dynamic programming formulation proposed in (Held and Karp, 1962). However, this approach is highly challenging. The algorithm, which utilizes the branch and bound technique as outlined in (Little *et al*., 1963), partitions the set of all potential tours after calculating the minimum possible length for each subset. Ultimately, it discovers a solitary excursion within a subset where all the distances are either less than or equal to a specified minimum value. There are two further precise solutions to the problem from the same category that are worth noting: branch-and-bound (Applegate, 2006) and branch-and-cut (Padberg and Rinaldi, 1991; Clímaco *et al*., 2021). However, in recent years, heuristics have gained significant attention in research circles due to their increasing intricacy. In (Carlton and Barnes, 1996), the authors expanded the objective function by incorporating a fixed penalty for exceeding time constraints in certain locations, aiming to tackle the problem of impossibility. In (López-Ibáñez *et al*., 2013) the authors proposed a constructive heuristic approach. This approach begins by solving an ad hoc assignment problem and then utilizes an insertion strategy to generate a full solution. Finally, the answer is further improved through the use of local search. The Lin-Kernighan heuristic, introduced in (Lin and Kernighan, 1973) is widely regarded as one of the most efficient techniques for producing optimal or near-optimal solutions for the symmetric Travelling salesman problem. The closest neighbour algorithm, utilized in (Mladenović, 2016), is another widely recognized approach. The tour commences by visiting a city selected at random initially, and concludes by visiting the nearest unknown city. The algorithm terminates once all cities have been visited on the tour. Furthermore, it is noteworthy to mention the insertion algorithms presented in (Gendreau *et al*., 1998). These algorithms begin with a tour that contains a small number of randomly selected places and then select a new location at each step that has not previously been included in the tour. The most minimal insertion cost is attained by integrating this location into the pre-existing itinerary that links two consecutive cities. In addition to these, nature-inspired heuristic optimization algorithms have been widely utilized as a means of addressing the Travelling salesman problem and its variations: (Cacchiani *et al*., 2023) uses a modified genetic algorithm to solve a TSP variation with three optimization objectives: fuel consumption, the energy required by speed variations and the energy used to curry the curb weight and the load on the vehicle

(Zhang *et al*., 2023; Hamza *et al*., 2023) uses a local search approach inspired by bees heuristic approach (Zhong *et al*., 2023) uses a collaborative neurodynamic optimization with neural networks.

Among these, ant colony optimization has garnered interest (Glabowski *et al*., 2012; Skinderowicz, 2022; Wu *et al*., 2023; Comert and Yazgan, 2023; Tong *et al*., 2023).

The Ant Colony Algorithm, which is based on the behaviour of ant colonies, is a highly popular heuristic algorithm due to its versatility in solving various sorts of problems, while maintaining a straightforward functionality. This approach has served as a source of inspiration for a wide range of strategies and methodologies employed in various sorts of minimization problems. These techniques, collectively referred to as Ant Colony Optimization (ACO), have been derived from the principles of ant colonies.

ACO is derived from the examination of certain ant species that employ pheromone emissions to mark the routes taken by individuals in their quest for sources of sustenance. The other individuals receive these quantities of pheromone, and eventually, the quickest route to the food source is established. In (Grassé, 1946), the author elucidates how he observed the response of certain ant species to particular stimuli. He observed that the responses of these ants to stimuli alter the behaviour of both the receiving ant and the ant producing the stimulus.

The propagation of information is determined by the quantity of pheromone released by individuals, as well as the overall amount emitted by the group as a whole. The information regarding the quantities of pheromones is discovered along the paths taken between the food supply and the den of the colony. This information helps to optimize the route and travel time, while also resulting in an increase in the number of individuals involved in food transportation. A different and intriguing method is suggested in (Pamosoaji and Raflesia, 2020) to address the Collision-Free Multiple Traveling Salesman Problem. In order to accomplish this task, a collective of mobile robots is required to traverse every node in the graph, commencing from distinct initial positions, while ensuring there are no collisions between them. This problem is resolved using an innovative method, in which every vehicle is represented by a distinct species of ants. It is considered that the collision-free condition occurs when the arrival time in the same node is at or above the minimum permissible value.

This study focuses on the topic of the Travelling Salesmen Problem with time windows (TSPTW) considering a robot that must reach a set of points of interest in a given time frame for each point. The issue involves the comparison of two solving methods: an exact method based on solving the TSPTW using mixed integer linear programming, and a heuristic way using Ant Colony Optimization (ACO).

3. Problem formulation

Given a 2D environment, a set of points of interest, an omnidirectional mobile robot (agent) must find a route such that each point is visited in an imposed time interval. The robot can move with a constant speed $v \in \mathbb{R}_+$ or it can pause its motion.

The points of interest are denoted by $P = \{p_1, p_2, \dots p_{|P|}\}$ and they represent the location that the robot must visit (cities that the Travelling salesman visits, e.g. Fig. 1). The robot must reach each p_i in a time interval (e_i, l_i) where $e_i \in \mathbb{R}_+$ represents the minimum entry time and $l_i \in \mathbb{R}_+$ the maximum departing time from the current point of interest. Fig. 1 also shows the time windows (TW), meaning the time interval (e_i, l_i) for each point. The robot starts at time 0 from the depot (the point of interest noted with s that does not have a time window) and after visiting all points it returns to the depot. It is assumed that the tasks in the points of interest are served instantaneously and the robot leaves each point as soon as reaching it. The robot can also wait in a certain point if its time window it is not yet opened upon arrival.

The environment with the points of interest is modelled as a graph which has the nodes $P = \{s, p_1, p_2, \dots p_{|P|}\}$ and the arcs are weighted with the Travelling distance between two points.

The first method used to find a solution involves solving a mixed linear programming problem using problem-based optimization solver in MATLAB. The optimization toolbox in MATLAB has two approaches to solving optimization problems: problem-based and solver-based, however, as mentioned here problem-based is used.

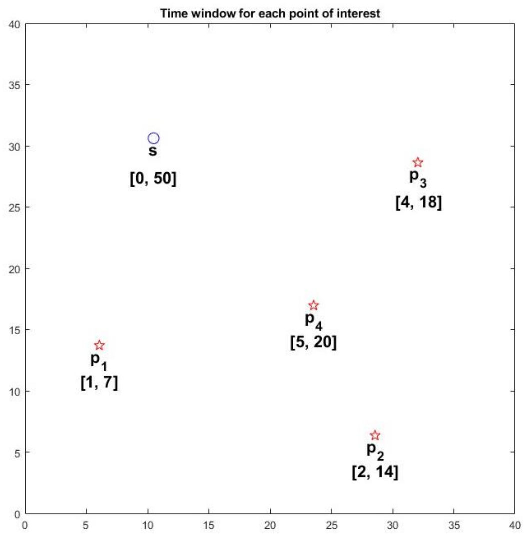

Fig. 1 – TSP. Cities time-windows.

4. Mixed linear integer programming approach

The TSPTW formulation is described and solved in this chapter. The problem is described using the format in Eq. (1) and later solved using Optimization Toolbox in MATLAB.

$$\min_{x} f^t x \ \ subject\ to \begin{cases} x \in \mathbb{Z}_+ \\ Ax \le b \\ A_{eq}x = b_{eq} \\ lb \le x \le ub \end{cases} \tag{1}$$

The TSPTW is modelled as a graph *G = (N, A),* with the following meaning:

• $N = \{n_0, n_1, \dots n_{|P|}\}$-nodes refer to the starting city(depot) and points of interest(cities). n_0 corresponds to starting point s and n_i corresponds to the rest of the cities c_i, $i = 1,$ $|P|$. An example for this can be seen in Fig.1 where there is a configuration of 4 cities and a depot

• $A = \{(n_i, n_j)/(n_i, n_j)\} \in N \times N$, with $n_i \ne n_j\}$, meaning any nodes of G are connected, i.e., G is a complete graph.

• c: N × N→ R+ associates the cost $c(n_i, n_j)$ equal with the time for following trajectory. It will be referred as c_{ij} throughout the paper

To obtain the solution, the following optimization problem is used:

$$\sum_{(i,j)\in A} c_{ij}\, x_{ij} \tag{2}$$

$$\sum_{j\in A} x_{ij} = 1 \tag{3}$$

$$\sum_{j\in A} x_{ij} - \sum_{j\in A} x_{ji} = 0 \tag{4}$$

$$x_{ij} = 1 => D_i + t_{ij} \le D_j \tag{5a}$$

$$D_i + t_{ij} - D_j \le (1 - x_{ij})M \tag{5b}$$

$$e_i \le D_j \le l_i \tag{6}$$

$$x_{ij} \in \{0,1\} \tag{7}$$

$$D_i - D_j + |P| \cdot x_{ij} \le |P| - 1 \tag{8}$$

The above formulation has the following meaning:

- The objective function Eq. (2) represents the total distance traveled by the robot. The variable x_{ij} takes the value 1 if there is a route between nodes n_i and n_j in the graph and 0 otherwise.
- Constraints Eq. (3) and Eq. (4) impose that each graph node is left once and entered once, respectively.
- Constraints Eq. (5a) and Eq. (5b) ensure that the robot respects the time windows defined in Section 2, for points of interest. Variable t_{ij} represents the time needed to travel between nodes n_i and n_j and D_i represents the departing time from the node n_j. Because Eq. (5a) is a deviation from the standard MILP this is transformed into linear inequalities by using the big-M method in Eq.(5b).
- A solution of the above MILP is easily transformed into a robot motion plan and the calculated route can be seen in Fig. 2. The robot departs from the depot and then visits $p_{1,}$ $p_{2,}$ $p_{4,}$ p_3 respecting thus, all time intervals.

The departure times from each point of interest show if the robot has to pause its movement in some positions.

- Eq. (8) assures the so-called subtour elimination, a necessary restriction in all TSP-derived problems.

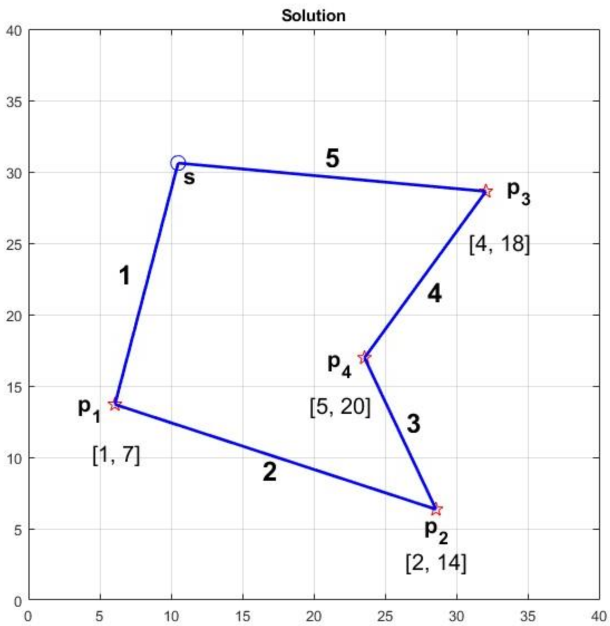

Fig. 2 – Solution given by MILP for example in Fig. 1.

The solution is determined using problem-based optimization with an *intlinprog* solver. In problem-based optimization, the user creates optimization variables, that represent the objective and constraints or that represent equations, and solves the problem using the *solve* function.

The optimization problem object is created by using the *optimproblem* function. A problem object is a container that defines an objective expression and constraints. The optimization problem object defines the problem and any bounds that exist in the problem. The variables are created with *optimvar*. An optimization variable is a symbolic variable that used to describe the problem objective and constraints. The default solver is changed to *intlinprog*.

5. Ant colony optimisation approach

In the literature, the schemes for solving the TSPTW usually define two distinct objectives. The first objective is to minimize the total travel time along the path, without considering the waiting time at customer sites. The second objective is to minimize the arrival time at the depot after completing the tour, which is also known as the overall tour time (Gendreau *et al*., 1998). For example, in (Cheng and Mao, 2007) the latter is adopted. Although this paper follows the algorithm from (Cheng and Mao, 2007) the objective is the former, so the method suffers some alteration.

Typically, all ACO algorithms for the TSPTW adhere to a same algorithmic framework.

- At the start, artificial ants, referred to as agents, are positioned at specific nodes within the network and the pheromone trails and parameters are initialized. Within the primary iteration, the ants initially create viable routes, which are subsequently enhanced through the implementation of local search techniques. This is also the step where all the necessary parameters are defined.

  m - represents the number of ants that will be used at each iteration

  α - parameter that determines the influence of pheromone quantities in choosing the road

  β - the parameter that describes the influence of the cost between 2 nodes in the choice of the road.

  w - the amount of pheromone on the arc

  ρ - pheromone evaporation rate, universal rate within the graph

  w_0- the initial amount of pheromone on the arcs

  p_{ij}- probability of choosing an arc

  c_{ij}- the cost attached to an arc (which is the same as the one defined in Chapter 3)

  Q - heuristic coefficient of the environment

- Each agent executes a sequence of random movements between adjacent nodes, following a predetermined transition probability (Eq. (9)) or rule. Typically, the likelihood of transitioning along a specific arc is contingent upon the level of pheromone concentration determined with Eq. (8) and the length of the arc. Using parameters α and β will set the influences of pheromones and arc length.

$$w_{ij} = w_{ij} + \frac{Q}{J} \tag{9}$$

$$p_{ij} = \frac{w_{ij}^{\alpha}}{c_{ij}^{\beta}} \tag{10}$$

- Once an agent has traversed all the nodes in the network, the tour's quality is assessed. The levels of pheromones on the paths taken during the tour are then adjusted using predetermined rules that depend on the tour's quality.
- Upon completing their tours, agents store the tour with the highest quality achieved up to the present time point. The agents then continue to repeat the aforementioned operations until the stop conditions are met.

Given the absence of pheromones on the paths, the initial selection of the first ant is entirely arbitrary. The ant will cover its chosen route, emitting pheromones that will be memorized by the surroundings as it moves. The pheromone levels will affect the following ants, and this influence will intensify as the number of journeys increases. Furthermore, as ants move continuously, the pheromone, which serves as an information signal, also evaporates. This evaporation process has a stronger impact when the branch on which it occurs is less frequently traveled, resulting in a smaller number of ants that have traversed the branch so far. The impact of pheromone deposition is contingent upon the length of the road. Following numerous individual movements, the pheromone concentrations on the trails will vary, so affecting the ants' selection of routes when foraging and returning. In this scenario, when the roads are identical and the selection of the road is purely random, over a significant duration, one of the two routes will inevitably exhibit a greater level. During each iteration, every ant in the given set is required to create a solution by traversing the complete graph, moving from one node to another, with the constraint of visiting each node only once. The selection of the next node from the present node is determined by a stochastic procedure that takes into account the pheromone levels on the related arcs. For instance, starting from node n_i, the subsequent node n_j is selected probabilistically, until the selected node has not been visited, according to the probabilities determined by the pheromone level. An ant will not choose a path that leads to a node already visited or to an already formed path. Additionally, if an edge is longer, it will receive less preference, since it implies higher travel times. Once the full route has been completed, if the optimization criterion of a program is met, the pheromone level is updated on the arcs that form the solution. Following each generation, the pheromone level is modified in accordance with the process of evaporation. Essentially, the technique follows the subsequent steps: Set parameters, initialize pheromone trails, Construct Solutions, Update Trails.

6. Comparison between the methods

The algorithms are applied to the Solomon Benchmark dataset R101 taking into consideration only the cities coordinates and time window constraints. Regarding the application of the methods for the datasets belonging to the example from Chapter 3, the numerical simulation for both of the methods is performed in a similar amount of time reaching up to 3 seconds regarding of the method used. This time is achieved taking into consideration, for the MILP a number of 30 variables and for the ACO 10 ants and 100 iterations. Simulation times for other configurations in terms of the number of points of interest can be found in Table 1. However, the timings for the R101 change considerably between the two methods.

With 100 cities, the dataset creates over 1000 variables for the MILP, however the maximum number of ants used was 15. For this simulation the runtime for ACO is around 100 seconds but for the MILP exceeds 500.

The exact method is advantageous because it gives us the optimum solution but in cases with a large number of cities, the run time is very high. The heuristic method offers a solution much faster for the cases mentioned above, however, it is not assured that the solution found is the optimal one. In the example, both algorithms offer the optimum solution.

Another advantage of the heuristic is that it aims to reduce the waiting time at nodes in the event that waiting cannot be avoided.

The Ant Colony Optimization algorithm had a good overall performance for the given constraints of this project. While its probabilistic side can bring a lot of complexity to the code, the easy implementation and possible improvements make it a great tool for estimating complex problems.

In other extensions for the algorithms, multiple ant colonies can be introduced, where each colony is responsible for a part of the problem or operates with different parameters. This approach increases diversity and helps in exploring a larger solution space.

Table 1
Simulation time comparison for different number of points of interest

No of points of interest $\lvert P\rvert$	No of ants m	Simulation time (MILP)	Simulation time (ACO)
5	10	1.49 s	1.02 s
8	10	2.54 s	1.3 s
15	10	3.48 s	2.07 s
20	15	30.16 s	16.3 s
30	15	Over 120 s	22.27 s
R101	15	500 s	100 s

7. Conclusions

This paper provides a quantitative comparison of two strategies employed to solve the Travelling Salesman problem with time windows.

The first approach uses an exact method by formulating the requirements as a mixed integer linear programming (MILP) optimization problem. The MILP guarantees the optimality of the solution, but it belongs to the NP-hard complexity class. For quantitatively investigating the results, we implement the MILP problem in MATLAB by using a problem-based optimization method and solve it with the *intlinprog* solver from the Optimization Toolbox.

The second solution employs a metaheuristic technique, utilizing an ant colony algorithm to address the optimization problem. Each ant begins its journey from a randomly chosen city (vertex of the graph) and progresses along the edges of the graph, remembering its path at each step of the construction process. In the following stages, the ant selects the edges that do not lead to cities that have already been visited, based on the levels of pheromone and the heuristic value. After the ant has traversed all the vertices in the graph, it has successfully constructed a solution. The pheromone on the edges is modified according to the quality of the solutions for each individual ant.

The performed comparison relies on multiple numerical simulations performed in a MATLAB environment. We highlight the advantages and disadvantages of both studied methods, by taking into consideration criteria as the simulation time. We acknowledge that both strategies can be selected to address the problem with the exact one offering the optimal solution and the heuristic one being a feasible substitute in case the exact methods need a large simulation time.

Future work will consider adding constraints to the problem and also other heuristics which have proven feasible with this kind of problems.

REFERENCES

Aguayo M.M., Sarin S.C., Sherali H.D., *Solving the single and multiple asymmetric traveling salesmen problems by generating subtour elimination constraints from integer solutions*, IISE Transactions, 2018 Jan 2, 50(1):45-53.

Applegate D.L., *The Travelling salesman problem: a computational study*, Princeton University Press, 2006.

Cacchiani V., Contreras-Bolton C., Escobar-Falcón L.M., Toth P., *A matheuristic algorithm for the pollution and energy minimization traveling salesman problems*, International Transactions in Operational Research, 2023, 30, 655-687, doi:10.1111/itor.12991.

Carlton W.B., Barnes J.W., *Solving the Travelling-salesman problem with time windows using tabu search*, IIE transactions, 1996 Aug 1, 28(8):617-629.

Chandra A., Natalia C., *Application of multiple traveling salesman problem on zone picking*, Academic Journal of Manufacturing Engineering, 2023 Jan 1, 21(1):51-58.

Cheikhrouhou O., Khoufi I., *A comprehensive survey on the Multiple Traveling Salesman Problem: Applications, approaches and taxonomy*, Computer Science Review, 40, p.100369, (2021).

Cheng C.B., Mao C.P., *A modified ant colony system for solving the travelling salesman problem with time windows*, Mathematical and Computer Modelling, 2007 Nov 1, 46(9-10):1225-35.

Clímaco G., Simonetti L., Rosseti I., *A branch-and-cut and MIP-based heuristics for the prize-collecting travelling salesman problem*, RAIRO - Operations Research 55, S719–S726, 2021, doi:10.1051/ro/2020002.

Comert S.E., Yazgan H.R., *A new approach based on hybrid ant colony optimization-artificial bee colony algorithm for multi-objective electric vehicle routing problems*, Engineering Applications of Artificial Intelligence, 2023 Aug 1, 123:106375.

Dantzig G., Fulkerson R., Johnson S., *Solution of a large-scale Travelling-salesman problem*, Journal of the operations research society of America, 1954 Nov, 2(4):393-410.

Dumas Y., Desrosiers J., Gelinas E., Solomon M.M., *An optimal algorithm for the traveling salesman problem with time windows*, Operations research, 1995 Apr, 43(2):367-371.

Fontaine R., Dibangoye J., Solnon C., *Exact and anytime approach for solving the time-dependent traveling salesman problem with time windows*, European Journal of Operational Research, 2023 Dec 16, 311(3):833-844.

Gendreau M., Hertz A., Laporte G., Stan M., *A generalized insertion heuristic for the Travelling salesman problem with time windows*, Operations Research, 1998 Jun, 46(3):330-335.

Glabowski M., Musznicki B., Nowak P., Zwierzykowski P., *Shortest Path Problem Solving Based on Ant Colony Optimization Metaheuristic. Image Processing & Communications*, (2012).

Grassé P.P., *Les Insectes Dans Leur Univers*, Ed. du Palais de la d́ecouverte, Paris, France, 1946.

Hamza A., Haj Darwish A., Rihawi O., *A new local search for the bees algorithm to optimize multiple traveling salesman problem*, Intelligent Systems with Applications, 2023, doi:10.1016/j.iswa.2023.200242.

Held M., Karp R.M., *A dynamic programming approach to sequencing problems*, Journal of the Society for Industrial and Applied mathematics, 1962 Mar, 10(1):196-210.

Khanra A., Maiti M.K., Maiti M., *Profit maximization of TSP through a hybrid algorithm*, Computers & Industrial Engineering, 2015 Oct 1, 88:229-236.

Larni-Fooeik A., Ghasemi N., Mohammadi E., *Insights into the application of the traveling salesman problem to logistics without considering financial risk: A bibliometric study*, Management Science Letters, 2024, 14(3):189-200.

Laporte G., Martello S., *The selective travelling salesman problem*, Discrete applied mathematics, 1990 Mar 1, 26(2-3):193-207.

Little J.D., Murty K.G., Sweeney D.W., Karel C., *An algorithm for the Travelling salesman problem*, Operations research, 1963 Dec, 11(6):972-989.

Lin S., Kernighan B.W., *An effective heuristic algorithm for the Travelling-salesman problem*, Operations research, 1973 Apr, 21(2):498-516.

López-Ibáñez M., Blum C., Ohlmann J.W., Thomas B.W., *The travelling salesman problem with time windows: Adapting algorithms from travel-time to makespan optimization*, Applied Soft Computing, 2013 Sep 1, 13(9):3806-3815.

Mladenović N., *An efficient general variable neighborhood search for large travelling salesman problem with time windows*, Yugoslav Journal of Operations Research, 2016 Oct 11, 23(1).

Padberg M., Rinaldi G., *A branch-and-cut algorithm for the resolution of large-scale symmetric Travelling salesman problems*, SIAM review, 1991 Mar, 33(1):60-100.

Pamosoaji A.K., Raflesia S.P., *Ant Colony Optimization-Based Multiple-AGV Route-and-Velocity Planning for Warehouse Operations*, In iMEC-APCOMS 2019: Proceedings of the 4th International Manufacturing Engineering Conference and the 5th Asia Pacific Conference on Manufacturing Systems (pp. 224-229), Springer Singapore, (2020).

Skinderowicz R., *Improving Ant Colony Optimization efficiency for solving large TSP instances*, Applied Soft Computing, 2022 May 1, 120:108653.

Tong S., Qu H., Xue J., *K-DSA for the Multiple Traveling Salesman Problem*, Journal of Systems Engineering and Electronics 34, 1614–1625, 2023, doi:10.23919/JSEE.2023.000023.

Wu L., Huang X., Cui J., Liu C., Xiao W., *Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot*, Expert Systems with Applications, 2023 Apr 1, 215:119410.

Zhang Z., Liu H., Zhou M., Wang J., *Solving Dynamic Traveling Salesman Problems with Deep Reinforcement Learning*, IEEE Transactions on Neural Networks and Learning Systems, 2023, 34, 2119-2132, doi:10.1109/TNNLS.2021.3105905.

Zhong J., Feng Y., Tang S., Xiong J., Dai X., Zhang N., *A collaborative neurodynamic optimization algorithm to traveling salesman problem*, Complex and Intelligent Systems 9, 1809-1821, 2023, doi:10.1007/s40747-022-00884-6.

COMPARAȚIE ÎNTRE O METODĂ EXACTĂ ȘI O METODĂ EURISTICĂ PENTRU REZOLVAREA UNEI PROBLEME DE PLANIFICARE A RUTEI UNUI VEHICUL CU CONSTRÂNGERI DE TIMP

(Rezumat)

Această lucrare prezintă o comparație între două metode diferite pentru rezolvarea probleme de rutare a unui vehicul de tipul "problema comisului voiajor" cu constrângeri de tip "fereastră de timp". Având în vedere un mediu cu un număr fix de orașe, un vehicul trebuie să găsească o rută astfel încât fiecare oraș să fie vizitat într-un interval de timp impus. Ambele metode investigate urmăresc să găsească calea costului minim în ceea ce privește distanța parcursă.

Prima abordare folosește o metodă exactă prin formularea cerințelor ca o problemă de optimizare de programare liniară cu numere întregi mixte (MILP).

A doua metodă implică o abordare metaeuristică, folosind procedura de colonie de furnici pentru a rezolva problema de optimizare.

Pe lângă informațiile calitative, comparația efectuată se bazează pe multiple simulări numerice efectuate în mediul MATLAB. Evidențiem astfel avantajele și dezavantajele ambelor metode, luând în considerare criterii precum timpul de simulare și diferența relativă dintre costurile obținute față de numărul de orașe.