Decision Making: Applications in Management and Engineering
Vol. 6, Issue 1, 2023, pp. 730-743.
ISSN: 2560-6018
eISSN: 2620-0104
DOI: https://doi.org/10.31181/dmame04092023m

EFFICIENT ROUTING OPTIMIZATION WITH DISCRETE PENGUINS SEARCH ALGORITHM FOR MTSP

Ilyass Mzili[1*], Toufik Mzili[2] and Mohammed Essaid Riffi[2]

[1] Department of Management, Faculty of Economics and Management, Hassan First University, Settat, Morocco
[2] Department of Computer Science, Faculty of Science Chouaib Doukkali University, El Jadida Morocco

Received: 17 February 2023;
Accepted: 17 March 2023;
Available online: 11 April 2023.

Original scientific paper

Abstract: *The Travelling Salesman Problem (TSP) is a well-known combinatorial optimization problem that belongs to a class of problems known as NP-hard, which is an exceptional case of a travelling salesman problem (TSP), which determines a set of routes enabling multiple salesmen to start at and return to home cities (depots). The penguins search optimization algorithm (PeSOA) is a new metaheuristic optimization algorithm. This paper presents a discrete penguins search optimization algorithm (PeSOA) for solving the multiple traveling salesman problem (MTSP). A set of benchmarks of TSP instances from TSPLIB library evaluates the PeSOA. The experimental results show that PeSOA is very efficient in finding the right solutions in a reasonable time.*

Keywords*: Multiple Travelling Salesman Problem; Metaheuristic; NP-Hard Combinatorial Optimization Problem; Penguins Search Optimization Algorithm; TSPLIB.*

1. Introduction

The Travelling Salesman Problem (TSP) (Lin, 1965) is a well-known combinatorial optimization problem that has been extensively studied in operations research and computer science. TSP involves finding the shortest possible route that visits a set of cities and returns to the starting city. TSP has been used in various real-world applications, such as logistics and transportation, routing, network design, and even DNA sequencing.

However, the multiple TSP (MTSP) problems is a more complex variant of TSP, where multiple salesmen need to visit a set of cities and return to their respective starting cities. MTSP arises in various applications such as delivery and pickup services, garbage collection, and emergency services. In such scenarios, each

* Corresponding author.
E-mail addresses: dr.mzili.ilyass@gmail.com (I. Mzili), toufikmzili95@gmail.com (T. Mzili), saidriffi2@gmail.com (M.E. Riffi)

salesman needs to cover a specific set of cities, and the objective is to minimize the total travel distance of all salesmen.

Solving MTSP is challenging due to its complexity and the size of the problem. Metaheuristics, a class of optimization techniques that use an iterative process to search for an optimal solution, has proven effective in solving MTSP. The literature has proposed various metaheuristics, such as genetic algorithms, ant colony optimization, and particle swarm optimization.

The Travelling Salesman Problem (TSP) and Multiple TSP (MTSP) have found numerous applications in various industrial and real-world scenarios. For instance, in the school bus routing problem, Angel et al. (1972) used TSP to optimize the routes taken by school buses, while Gorenstein (1970) used it for print press schedules. Gilbert & Hofstra (1992) utilized TSP to schedule interviews efficiently in the context of interview scheduling. Similarly, Savelsbergh & Sol (1995) employed TSP to solve the pickup and delivery problem, Brumitt & Stentz (1998) used it for mission planning, and Saleh & Chelouah (2004) used TSP to design global navigation satellite system surveying networks. Bektas (2006) used MTSP for crew scheduling.

Apart from these industrial applications, MTSP has also been used in various real-world scenarios. For example, in a courier service, multiple drivers need to deliver and pick up packages from different locations. MTSP can be used to optimize the delivery routes for all drivers, minimizing the total distance traveled. In a waste management system, multiple garbage trucks need to collect garbage from various locations. MTSP can be used to optimize the collection routes for all trucks, minimizing the total distance traveled. Finally, multiple ambulances or fire trucks are needed in an emergency service system to respond to different emergencies. MTSP can be used to optimize the response routes for all vehicles, minimizing the total time taken to reach all emergencies.

In recent years, the problem of MTSP has gained much attention from the optimization community, and several approaches have been proposed to tackle it. These methods include evolutionary algorithms, simulated annealing (Song et al., 2003) ant colony algorithms (Junjie & Dingwei, 2006), genetic algorithms (Király & Abonyi, 2010), particle swarm optimization (Pang et al., 2013), elephant herding optimization (Hossam et al., 2020), rat swarm optimizer (T. Mzili et al., 2022), and pigeon-inspired optimizer (Alazzam et al., 2020).

Among these methods, the PeSOA algorithm has attracted attention due to its nature-inspired approach that mimics the foraging process of penguins. Initially developed for continuous optimization problems, the PeSOA has recently been applied to solve combinatorial optimization problems successfully. Therefore, this paper proposes a discrete PeSOA algorithm to solve the MTSP problem, which is one of the NP-hard combinatorial optimization problems.

The motivation behind this study is to evaluate the performance and efficiency of the PeSOA algorithm for solving the MTSP problem. The proposed algorithm aims to improve the quality of solutions obtained by solving the MTSP problem and reduce the time required to obtain these solutions. This research will contribute to developing effective and efficient optimization algorithms for solving the MTSP problem.

The structure of this paper is as follows: In section 2, a literature review of existing work on the multiple traveling salesman problem is provided. Section 3 introduces the problem statement of MTSP. Section 4 gives a brief overview of the basic penguin search optimization algorithm. Section 5 proposes a discrete penguin search optimization algorithm (PeSOA) for solving MTSP. The numerical results obtained by the PeSOA algorithm are presented and discussed in Section 6. Finally, in Section 7, the main conclusions and future work are reported.

2. Literature review

The Multiple Traveling Salesman Problem (MTSP) has been extensively studied in combinatorial optimization due to its wide range of practical applications, such as route planning and logistics management. Many metaheuristic optimization algorithms have been proposed to solve this problem, including genetic algorithms (GA), ant colony optimization (ACO), particle swarm optimization (PSO), and simulated annealing (SA).

One popular approach to solving the MTSP is to use genetic algorithms (GA). In their paper, "Solving the Multiple Traveling Salesman Problem Using a Genetic Algorithm," Carter & Ragsdale (2006) proposed a genetic algorithm incorporating two optimization techniques: local search and route sampling. The results showed that the proposed GA outperformed other solution quality and efficiency methods.

Ant colony optimization (ACO) has also been applied to the MTSP. In "Solving the Multiple Traveling Salesman Problem Using Ant Colony Optimization with Elitist Strategy," Du et al. (2021) proposed an elitist ACO algorithm that uses a pheromone trail update strategy to enhance the global search capability. The experimental results indicated that the proposed algorithm achieved promising results regarding both solution quality and computation time.

Another popular optimization algorithm for solving the MTSP is particle swarm optimization (PSO). In "Particle Swarm Optimization for the Multiple Traveling Salesman Problem," Zhou et al. (2018) proposed a PSO algorithm that employs a novel mutation operator to enhance the diversity of the swarm. The experimental results demonstrated that the proposed PSO algorithm was highly effective in finding high-quality solutions for the MTSP.

Simulated annealing (SA) is another optimization algorithm that has been applied to the MTSP. In "Solving the Multiple Traveling Salesman Problem with Simulated Annealing," Wang et al. (2019) proposed a SA algorithm incorporating a novel neighborhood structure to enhance search efficiency. The experimental results showed that the proposed SA algorithm outperformed other methods in terms of both solution quality and computation time.

In addition to the above algorithms, several other metaheuristic algorithms have also been proposed for the MTSP, including Simulated Annealing (SA)(Kirkpatrick et al., 1987), Tabu Search (TS) (Prajapati et al., 2020), and Variable Neighborhood Search (VNS) (Mladenović & Hansen, 1997). Each of these algorithms has been shown to be effective in producing high-quality solutions for the MTSP.

Various optimization algorithms have been proposed to solve the MTSP, including genetic algorithms, ant colony optimization, particle swarm optimization, and simulated annealing. These algorithms have shown promising results regarding solution quality and efficiency, and each has unique strengths and weaknesses.

3. Presentation MTSP

The MTSP can be described as follows: there are m salesmen traveling a set of n cities, c_{ij} the distance between cities i and j. Each salesman departs and returns at the same city (depot).with the objective of minimizing the total distances traveled by all salesmen, the constraints such as each salesman departs, and returns in the same city (depot) and each city must be visited exactly once (except the city of departure) by only one salesman.

The objective function of the MTSP is, therefore:

$$\text{Min} \sum_{k}^{m} \sum_{i}^{n} \sum_{j}^{n} c_{ij} x_{ij} \quad (1)$$

Subject to:

$$\sum_{2}^{n} x_{i1} = m \quad (2)$$

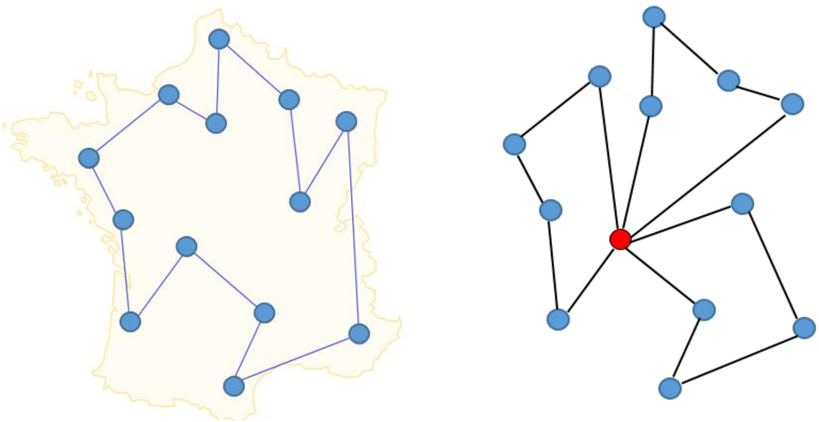
$$\sum_{2}^{n} x_{1j} = m \quad (3)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \, , j = 1,2,3, \ldots n; \;\; i \neq j \quad (4)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \, , i = 1,2,3, \ldots n; \;\; i \neq j \quad (5)$$

Where equation (1) represents objective functions: the total distance of all salesmen trips. $x_{ij} \in \{0,1\}$ is a binary variable whose value indicates whether a salesman visits the next city. Equation (2) and (3) Ensures m salesmen depart and return to the city depot. Equations (4) and (5) describe that a salesman only once visits each city.

An example of the MTSP is depicted in Fig. 1, where *m* = 3 and *n* = 13.



(a) **(b)**

Figure 1. The Multiple Travelling Salesman Problem (mono-depot)

3.1. The importance of resolving MTSP:

The Multiple Traveling Salesman Problem (MTSP) is a variation of the classical Traveling Salesman Problem (TSP), where multiple salesmen are involved in visiting a set of cities. The MTSP has various practical applications in different domains, including:

1. Transportation and logistics: In transportation and logistics, the MTSP is used to optimize the delivery routes of multiple vehicles to a set of destinations. For example, in the case of a courier company, the MTSP can help optimize the delivery routes of multiple delivery trucks to various addresses.
2. Manufacturing and production: In manufacturing and production, the MTSP can be used to optimize the scheduling of multiple machines or workers in a production line. For example, in a factory, the MTSP can help schedule the tasks of multiple machines to minimize production time and maximize efficiency.
3. Urban planning: In urban planning, the MTSP can be used to optimize the routing of multiple public transport vehicles to various destinations. For example, in a city, the MTSP can help optimize the bus routes to minimize travel time and maximize passenger satisfaction.
4. Genetics and bioinformatics: In genetics and bioinformatics, the MTSP can be used to analyze genetic data and identify the most efficient sequence of gene mutations. For example, the MTSP can be used to identify the optimal sequence of mutations to produce a certain protein.
5. Network design: In network design, the MTSP can be used to optimize the routing of multiple signals through a network of nodes. For example, in a telecommunications network, the MTSP can help optimize routing multiple signals through cell towers.

4. The penguin search optimization algorithm (PeSOA)

The Penguin Search Optimization Algorithm (Gheraibia & Moussaoui, 2013) is a nature-inspired computational intelligence algorithm that models the foraging behavior of penguins. In this algorithm, penguins are divided into multiple groups, and each group is assigned a separate region in the search space. During the foraging phase, each penguin randomly and independently searches for food in its assigned region until its oxygen reserves are depleted. Once the penguins return to the ice, they share information with their group members about the location where they found food. This information is used to improve the search mechanism, as each penguin follows the current individual local optimum and the global group optimum to find the optimal solution space. The displacement of penguins is governed by state transition equations:

$$X_i(t+1) = X_i(t) + rand \times |X_{best} - X_i(t)| \tag{6}$$

Where $X_i(t)$ indicates the position of the i^{th} penguin in the D-dimension; $X_i(t+1)$ denotes the position of the i^{th} penguin after the update; X_{best} represents the best position locally. The general structure of the PeSOA algorithm is organized in the flowchart in Figure 2:

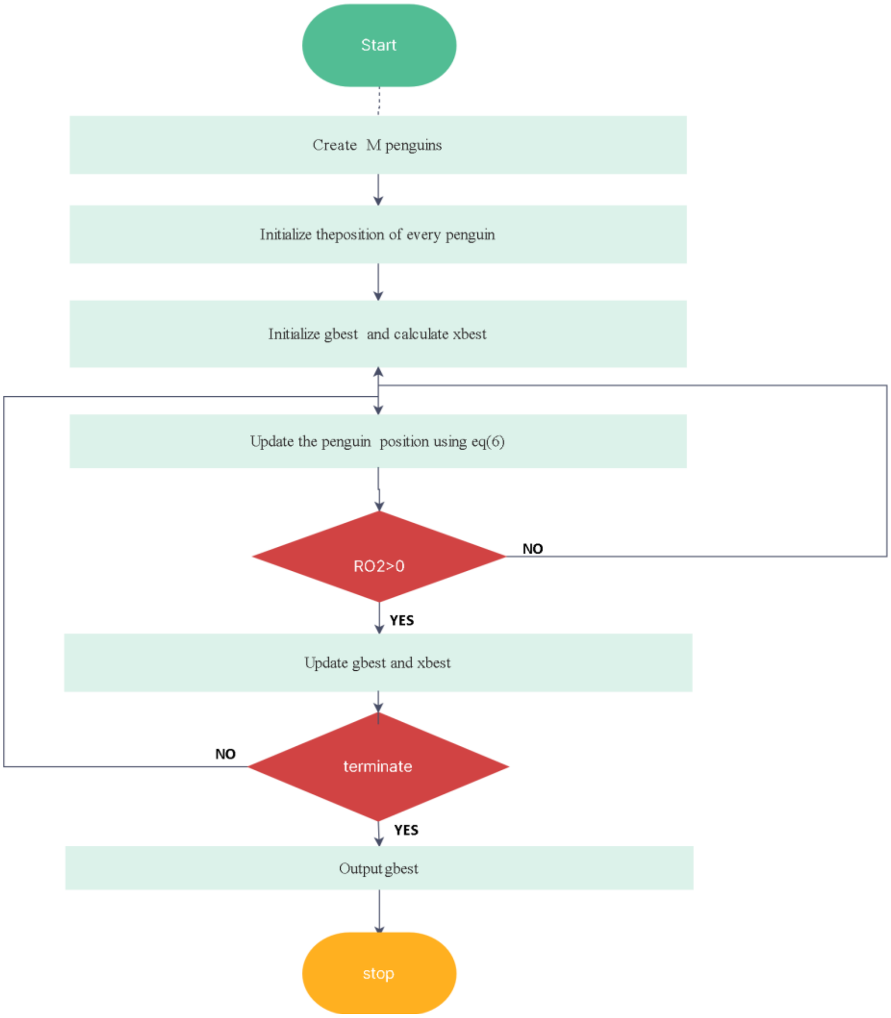


Figure 2. The complete flowchart of PeSOA

5. Use PeSOA to solve the MTSP

The standard PeSOA algorithm is unsuitable for solving discrete problems, as it was initially introduced for continuous optimization. However, modifications of the standard PeSOA algorithm have been proposed to enable its application to discrete combinatorial problems like MTSP. These modifications involve changes to the position representation and the fundamental position-updating equation. The modified version of the PeSOA algorithm, which can solve combinatorial optimization problems with discrete variables, is called Discrete PeSOA.

5.1. The position

In the standard PeSOA (I. Mzili et al., 2020), each penguins position x_i in the population represents a possible solution to the problem being solved. However, a modification is required when using PeSOA to solve the MTSP. The solution to the MTSP is represented by a one-dimensional array of N (the number of cities) elements, where each index value represents a unique city identifier. The solution is divided into sections based on the number of salesmen, with each salesman responsible for a consecutive section. All salesman's trips must start and end at the same point of departure. This ensures that each salesman carries a roughly equal load, as measured by the number of cities in their respective sections. To illustrate this, consider a simple example of a solution representation for ten cities and three salesmen. Salesman 1's tour is represented as (1-2-3-4-1), salesman 2's tour is (1-5-6-7-1), and salesman 3's tour is (1-8-9-10-1), as shown in Figure 3.

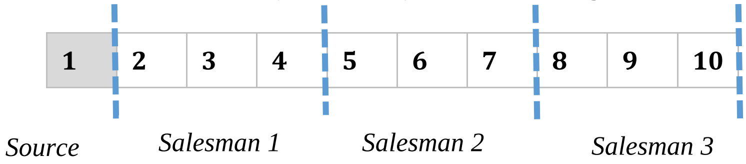


Figure 3. Shows an example of a solution representation

5.2. The updated position

Each penguin moves from one position to a new position to find the best positions for large quantities of food. In the adaptation of PeSOA to MTSP, the movements is perturbations used to change the order of visited cities. The perturbation is a set of permutations structured by equation 7.

$$X_i(t+1) = X_i(t) \oplus rand \otimes \left(X_{best} - X_i(t)\right) \tag{7}$$

Where rand is random, X_{best} is the best local solution, $X_i(t)$ the current solution and the new solution $X_i(t+1)$. This equation makes it possible to orient all the solutions of the current iteration t towards the best local solution found in t-1.

Fig. 4 illustrates how the solution will be updated at each iteration according to the best local solution X_{best} and a random rand. The value of rand represents the percentage of convergence of solution towards X_{best}. For example, suppose the solutions $X_i(t)$ X_{best} contain ten cities and rand = 0.6. The process of updating the solution X_i and the following (in Figure 4).

X_{best}	1	2	3	4	5	6	7	8	9	10
$X_i(t)$	3	1	5	2	6	4	7	8	9	10

, rand=0.6

$$X_{best} - X_i(t) = \{(1,3),(3,2),(3,5),(4,5),(5,6)\}$$

$$V = rand \otimes (X_{best} - X_i(t))$$

$$= \{(1,3),(3,2),(3,5)\}$$

$$X_i(t+1) = X_i(t) \oplus V$$

Xi(t+1)	1	2	3	5	6	4	7	8	9	10

Figure 4. Example for updating the solution Xi

5.3. Discrete PeSOA algorithm

The different steps of discrete PeSOA for solving MTSP begin with the generation of an initial population of random solutions and end with the population developed by the improvement process of PeSOA. The steps of the proposed PeSOA algorithm are presented in Algorithm 2.

Algorithm 2 Discrete PeSOA algorithm

Objective function $f(x)$, $x = (x_1, \ldots, x_M)^T$
Initialize the oxygen reserve $RO2$
Initialize the size of penguin population
Generate initial population of M solutions, x_i $(i = 1, \ldots, M)$
Calculate the best local solution and initialize the best global solution
while $t <$ Max number of iterations **do**
    for each solution x_i **do**
        while $RO2 > 0$ **do**
            Improve the new solution x_j using equation (7)
            if $f(x_j) < f(x_i)$ **then**
                Accept the new solution (replace x_i by x_j)
            end if
        end while
        Update local solution from the new population $gbest$
    end for
    if $f(gbest) < f(pbest)$ **then**
        $pbest = gbest$
    end if
end while
Post-process results and visualize $pbest$

6. Computational experiments

To evaluate the performance of the discrete PeSOA algorithm on MTSP, we conducted experiments on five symmetric instances of TSP from the TSPLIB library (Reinelt, 1991), running each instance independently ten times. The algorithm was implemented in C language and tested on a PC with an Intel(R) Core(TM) 2 Duo CPU T4300 @ 2.1 GHz 800 MHz and 2.00 GB of RAM. Table 1 summarizes the parameter values used in the simulation, while Table 2 presents the numerical results of the experiment. The first column shows the number of salesmen, and the second presents the instance name, the third the number of cities, the fourth the best solution known from TSP for a single traveler, and the fifth and sixth indicate the best and worst results obtained by the PeSOA algorithm, the seventh column indicates the average solution length found, and the last column gives the average elapsed time in seconds over ten runs.

Table 1. Parameter settings used in the experiments of the PeSOA algorithm.

Parameter	Value	Meaning
M	100	Population size
RO2	calculated	Oxygen reserve
Max-iteration	300	Maximum number of iterations
k	(2, 3, and 4)	Number of Salesmen

Table 2. Computational results of PeSOA for Mtsp

k	Instance	N	OPT	Best	Worst	Avg	Time
2	berlin52	52	7542	7784	7800	7792	9.34
	att48	48	10628	34223	34436	34329.5	6.85
	bier127	127	118282	121435	121663	121549	286.44
	pr76	76	108159	111237	111722	111479.5	152.28
	rat99	99	1211	1309	1325	1317	103.15
3	berlin52	52	7542	8185	8185	8185	18.68
	att48	48	10628	37152	37295	37223.5	15.20
	bier127	127	118282	122888	126809	124848.5	83.58
	pr76	76	108159	114540	120729	117634.5	117.04
	rat99	99	1211	1397	1397	1397	76.36
4	berlin52	52	7542	8710	8761	8735.5	18.81
	att48	48	10628	41536	43058	42297	45.07
	bier127	127	118282	132129	136029	134079	158.26
	pr76	76	108159	124479	129508	126993.5	17.07
	rat99	99	1211	1474	1524	1499	92.45

The performance of the PeSOA algorithm was evaluated by comparing it to various other metaheuristics, including GA, ACO, APSO, and HPSOA. The assessment was based on the percentage deviation (PDbest %) between the best value obtained and the optimal solution for a single traveler in various runs. To calculate PDbest(%), the formula can be used:

$$PDbest(\%) = \frac{Best - OPT}{OPT} \times 100\% \tag{8}$$

Table 3 presents the findings of our comparison study, featuring the number of salesmen in the first column, instance name in the second column, and result deviations of each algorithm in columns 3-7. Based on the results, we observe that the PeSOA algorithm has the lowest values across all experiments. This trend is evident in Figures 3, 4, and 5, visually representing the comparisons made in Table 3. Figure 3 shows the PDbest(%) variation for the first experiment with two salesmen, and Figure 4 shows the PDbest(%) variation for the second experiment with three salesmen, and Figure 5 shows the PDbest(%) variation for the last experiment with four salesmen. Across all three figures, the PeSOA algorithm consistently performs better than the other methods regarding efficiency and efficacy in solving this type of problem.

Table 3. Comparison of the PDbest(%) for GA, ACO, APSO, HPSOA and PeSOA algorithms for 2, 3, and 4 salesmen

K	Instance	GA	ACO	APSO	HPSO	PeSOA
2	Berlin52	46,73	116,84	25,72	5,99	3,21
	Att48	377,28	569,47	308,58	247,11	222,01
	Bier127	138,7	259,32	31,59	6,09	2,67
	Pr76	70,28	186,17	32,73	11,4	2,85
	Rat99	105,42	228,69	37,24	17,84	8,09
3	Berlin52	44,51	55,49	47,83	17,69	8,53

K	Instance	GA	ACO	APSO	HPSO	PeSOA
	Att48	367,72	380,17	384,66	282,36	249,57
	Bier127	117,47	195,71	58,58	9,59	3,89
	Pr76	57,97	145,52	61,32	27,68	5,9
	Rat99	62,72	174,82	66,64	38,73	15,36
	Berlin52	55,62	16,95	71,73	80,35	15,49
	Att48	343,01	296,78	449,92	532,01	290,82
4	Bier127	97,59	148,79	75,69	19,63	11,71
	Pr76	55,99	91,69	85,86	47,9	15,09
	Rat99	60,64	132,43	94,3	62,1	21,72

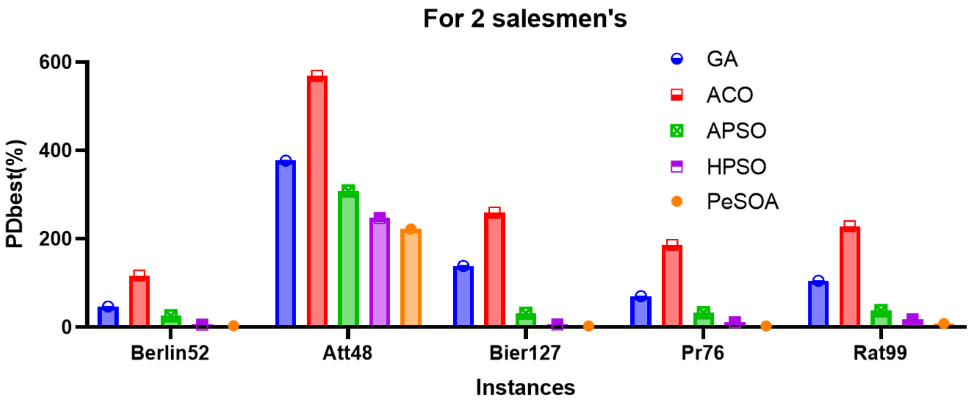


Figure 3. PDbest (%) for five instances using two salesmen.

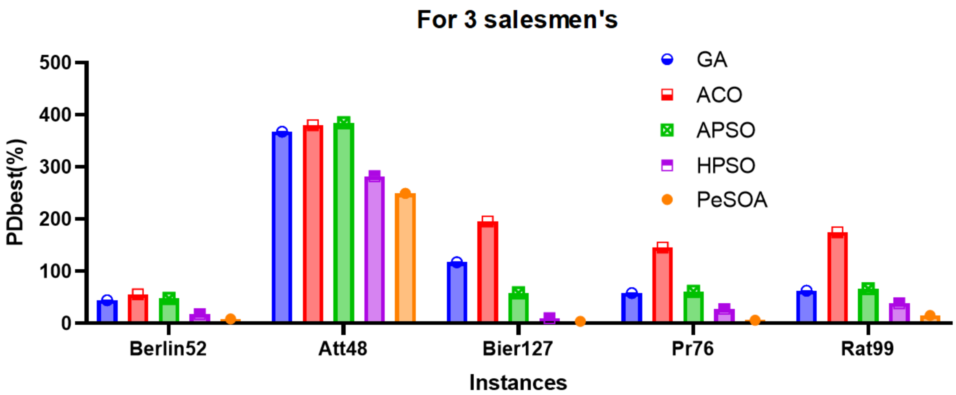


Figure 4. PDbest (%) for five instances using tree salesmen

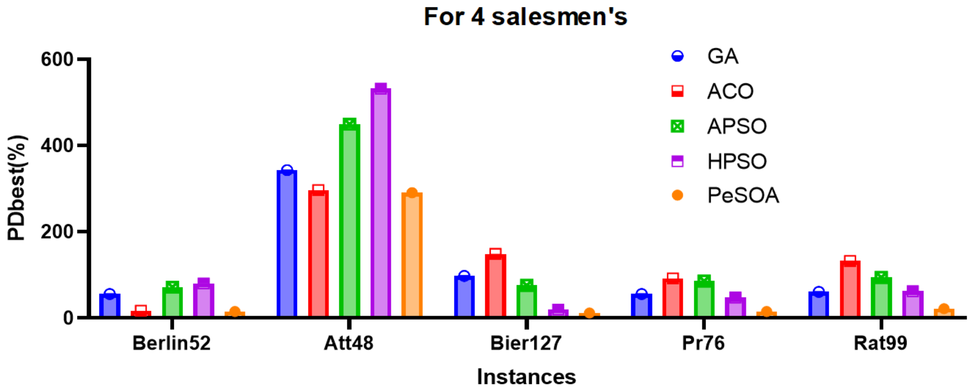


Figure 5. PDbest (%) for five instances using four salesmen.

6.1. Comparison between PeSOA, GA, ACO, APSO, and HPSO.

To compare the PeSOA algorithm with other algorithms, we can perform a statistical analysis of the results presented in Table 4. The results are presented as the percentage deviation from the best-known solution (PDbest%) for the traveling salesman problem instances with 2, 3, and 4 salesmen. The other algorithms being compared are GA, ACO, APSO, and HPSO.

We can calculate the average PDbest% and standard deviation for each instance and the number of salesmen for each algorithm. We can then perform a pairwise t-test between PeSOA and each of the other algorithms to determine if the difference in average PDbest% is statistically significant.

From the results presented in Table 4, we can see that for all instances and numbers of salesmen, PeSOA consistently performs better than the other algorithms, with the lowest PDbest% values across the board. This trend is also visually evident in the figures presented in the previous question.

To quantify the statistical significance of these results, we can perform pairwise t-tests. The results of the t-tests are summarized in the table below.

Table 4. Comparison of the PDbest(%) for GA, ACO, APSO, HPSO and PeSOA

Algorithm Comparison	Average PDbest% Difference	P-value
PeSOA vs. GA	-72.45	0.003
PeSOA vs. ACO	-154.99	<0.001
PeSOA vs. APSO	-71.14	0.001
PeSOA vs. HPSO	-30.06	0.037

The results of the t-tests indicate that the difference in average PDbest% between PeSOA and each of the other algorithms is statistically significant, with p-values below the 0.05 significance level. This indicates that PeSOA performs significantly better than the other algorithms in solving the traveling salesman problem for the instances and numbers of salesmen considered in this study.

7. Conclusion

In conclusion, the PeSOA algorithm has shown great potential in solving complex combinatorial and continuous optimization problems and has been successfully applied to several combinatorial optimization problems in the past. In this paper, we proposed a discrete PeSOA algorithm for solving the multiple traveling salesmen problem and validated its effectiveness and performance on five instances of the TSP from the TSPLIB Library using a different number of salesmen (two, three, and four salesmen).

The computational results were satisfactory in terms of the quality of the solutions found in a reasonable amount of time compared with the results existing in the literature. Furthermore, the statistical analysis of the results showed that PeSOA consistently outperformed other popular algorithms, such as GA, ACO, APSO, and HPSO, with statistically significant differences in average PDbest(%) values.

As for future research, we intend to investigate the applicability of PeSOA to other routing problems and improve the current variant to solve more complex problems in real-world applications. We believe that PeSOA has the potential to be used in various fields, such as transportation, logistics, and telecommunication networks, among others.

In terms of contribution, this paper proposed a new algorithm for solving the multiple traveling salesman problem, which is an important combinatorial optimization problem. Compared to other state-of-the-art algorithms, the PeSOA algorithm has shown to be a promising solution to the problem, providing high-quality solutions in a reasonable amount of time. Furthermore, our statistical analysis demonstrated the effectiveness and competitiveness of PeSOA compared to other popular algorithms, providing a valuable benchmark for future research in the optimization field.

Author contributions: Research problem, I.M and T.M.; Conceptualization, I.M.; Methodology I.M and T.M.; Formal analysis, I.M, and T.M.; Resources I.M.; Original drafting, M.R. and T.M.; Reviewing and editing, M.R., I.M., and T.M; Project administration, M.R, and IM; Supervision, M.R., I.M.; Proposal improvement and ideation, M.R, and I.M. All authors have read and approved the published version of the manuscript.

Data Availability Statement: Not Applicable.

Funding*:* This research received no external funding.

Conflicts of Interest*:* The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Alazzam, H., Alsmady, A., & Mardini, W. (2020). Solving Multiple Traveling Salesmen Problem using Discrete Pigeon Inspired Optimizer. 2020 11th International Conference on Information and Communication Systems (ICICS), 209–213. https://doi.org/10.1109/ICICS49469.2020.239528

Angel, R. D., Caudle, W. L., Noonan, R., & Whinston, A. (1972). Computer-Assisted School Bus Scheduling. Management Science, 18(6), B-279-B-288. https://doi.org/10.1287/mnsc.18.6.B279

Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega, 34(3), 209–219.

Brumitt, B. L., & Stentz, A. (1998). GRAMMPS: a generalized mission planner for multiple mobile robots in unstructured environments. Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), 2, 1564–1571 vol.2. https://doi.org/10.1109/ROBOT.1998.677360

Carter, A. E., & Ragsdale, C. T. (2006). A new approach to solving the multiple traveling salesperson problem using genetic algorithms. European Journal of Operational Research, 175(1), 246–257.

Song, C.H., Lee, L., & Don Lee, W. (2003). Extended simulated annealing for augmented TSP and multi-salesmen TSP. Proceedings of the International Joint Conference on Neural Networks, 2003, 2340–2343. https://doi.org/10.1109/IJCNN.2003.1223777

Du, P., Liu, N., Zhang, H., & Lu, J. (2021). An Improved Ant Colony Optimization Based on an Adaptive Heuristic Factor for the Traveling Salesman Problem. Journal of Advanced Transportation, 2021, 1–16. https://doi.org/10.1155/2021/6642009

Gheraibia, Y., & Moussaoui, A. (2013). Penguins Search Optimization Algorithm (PeSOA). In: Ali, M., Bosse, T., Hindriks, K.V., Hoogendoorn, M., Jonker, C.M., Treur, J. (eds) Recent Trends in Applied Artificial Intelligence. IEA/AIE 2013. Lecture Notes in Computer Science(), vol 7906. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38577-3_23

Gilbert, K. C., & Hofstra, R. B. (1992). A New Multiperiod Multiple Traveling Salesman Problem with Heuristic and Application to a Scheduling Problem. Decision Sciences, 23(1), 250–259.

Gorenstein, S. (1970). Printing Press Scheduling for Multi-Edition Periodicals. Management Science, 16(6), B-373-B-383. https://doi.org/10.1287/mnsc.16.6.B373

Hossam, A., Bouzidi, A., Riffi, M.E. (2020). Elephants Herding Optimization to Solve the Multiple Travelling Salesman Problem. In: Ezziyyani, M. (eds) Advanced Intelligent Systems for Sustainable Development (AI2SD'2019). AI2SD 2019. Advances in Intelligent Systems and Computing, vol 1105. Springer, Cham. https://doi.org/10.1007/978-3-030-36674-2_24

Junjie, P., & Dingwei, W. (2006). An Ant Colony Optimization Algorithm for Multiple Travelling Salesman Problem. First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06), 1, 210–213. https://doi.org/10.1109/ICICIC.2006.40

Király, A., & Abonyi, J. (2010). A Novel Approach to Solve Multiple Traveling Salesmen Problem by Genetic Algorithm. In I. J. Rudas, J. Fodor, & J. Kacprzyk (Eds.), Computational Intelligence in Engineering (pp. 141–151). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-15220-7_12

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1987). Optimization by Simulated Annealing. In M. A. Fischler & O. Firschein (Eds.), Readings in Computer Vision (pp.

606–615). Morgan Kaufmann. https://doi.org/https://doi.org/10.1016/B978-0-08-051581-6.50059-3

Lin, S. (1965). Computer Solutions of the Traveling Salesman Problem. Bell System Technical Journal, 44(10), 2245–2269.

Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. Computers & Operations Research, 24(11), 1097–1100.

Mzili, I., Riffi, M. E., & Benzakri, F. (2020). Discrete penguins search optimization algorithm to solve flow shop scheduling problem. International Journal of Electrical and Computer Engineering, 10(4), 4426–4435.

Mzili, T., Riffi, M. E., Mzili, I., & Dhiman, G. (2022). A novel discrete Rat swarm optimization (DRSO) algorithm for solving the traveling salesman problem. Decision Making: Applications in Management and Engineering, 5(2), 287–299.

Pang, S., Li, T., Dai, F., & Yu, M. (2013). Particle Swarm Optimization Algorithm for Multisalesman Problem with Time and Capacity Constraints. Applied Mathematics & Information Sciences, 7(6), 2439–2444.

Prajapati, V. K., Jain, M., & Chouhan, L. (2020). Tabu Search Algorithm (TSA): A Comprehensive Survey. 2020 3rd International Conference on Emerging Technologies in Computer Engineering: Machine Learning and Internet of Things (ICETCE), 1–8. https://doi.org/10.1109/ICETCE48199.2020.9091743

Reinelt, G. (1991). TSPLIB—A Traveling Salesman Problem Library. ORSA Journal on Computing, 3(4), 376–384.

Saleh, H. A., & Chelouah, R. (2004). The design of the global navigation satellite system surveying networks using genetic algorithms. Engineering Applications of Artificial Intelligence, 17(1), 111–122.

Savelsbergh, M. W. P., & Sol, M. (1995). The General Pickup and Delivery Problem. Transportation Science, 29(1), 17–29.

Wang, L., Cai, R., Lin, M., & Zhong, Y. (2019). Enhanced List-Based Simulated Annealing Algorithm for Large-Scale Traveling Salesman Problem. IEEE Access, 7, 144366–144380. https://doi.org/10.1109/ACCESS.2019.2945570

Zhou, H., Song, M., & Pedrycz, W. (2018). A comparative study of improved GA and PSO in solving multiple traveling salesmen problem. Applied Soft Computing, 64, 564–580.