MATHEMATICAL MODELING AND COMPUTING, Vol. 11, No. 1, pp. 290–299 (2024)

A combined ant colony optimization with Levy flight mechanism for the probabilistic traveling salesman problem with deadlines

El Asri F., Tajani C., Fakhouri H.

SMAD Team, Polydisciplinary Faculty of Larache, Abdelmalek Essaadi University, Tetouan, Morocco

(Received 26 June 2023; Accepted 9 March 2024)

In this paper, we are interested in the Probabilistic Traveling Salesman Problem with Deadlines (PTSPD) where clients must be contacted, in addition to their random availability before a set deadline. The main objective is to find an optimal route that covers a random subset of visitors in the same order as they appear on the tour, attempting to keep the path as short as possible. This problem is regarded as being ♯P-hard. Ant Colony Optimization (ACO) has been frequently employed to resolve this challenging optimization problem. However, we suggest an enhanced ACO employing the Levy flight algorithm in this study. This allows some ants to take longer jumps based on the Levy distribution, helping them escape from local optima situations. Our computational experiments using standard benchmark datasets demonstrate that the proposed algorithm is more efficient and accurate than traditional ACO.

Keywords: *stochastic combinatorial optimization; probabilistic traveling salesman problem; deadlines; metaheuristics, ant colony optimization; Levy flight.*

2010 MSC: 90C15, 90C27, 90-08, 65K05 **DOI:** 10.23939/mmc2024.01.290

1. Introduction

Stochastic Combinatorial Optimization (SCOP) is an important field of study that utilizes probabilistic methods to optimize complex systems. The objective is to identify the optimal outcome among a given range of possible solutions, where the quality of each solution is determined by a stochastic function. SCOP has gained significant attention in recent times and has made a significant impact in various fields, such as Stochastic Integer Programming, Markov Decision Processes and Simulation Optimization. It is useful in a variety of practical contexts, including vehicle routing problems, routing on information networks, finance, organization and location problems.

The Probabilistic Traveling Salesman Problems with Deadlines (PTSPD), introduced by Campbell and Thomas in 2008, is an important class of SCOP problems [1]. Weyland et al. later presented a variant of this problem called the Probabilistic Traveler Problem, which takes into account temporal constraints related to deadlines [2,3]. The PTSPD is an advanced version of the traditional Probabilistic Traveling Salesman Problem (PTSP) that considers the time limitations and uncertain availability of customers. The challenge is to determine a route that starts and finishes at a depot, visits each customer exactly once, ignores any customers who do not need to be visited, and takes temporal dependencies into account in terms of deadlines. This problem has applications in many fields where it is necessary to plan tours with deadlines and uncertainties, such as delivery, collection, maintenance, distribution or visit.

By using the PTSPD, one can minimize the cost of travel, avoid delays or non-deliveries, manage stocks or expiration dates. There are various approaches to tackle the PTSPD, including the chance constrained model and two different recourse models. Our research mainly centers on the recourse model that involves two distinct phases. Firstly, we determine a preliminary solution that determines the sequence in which customers will be served. Secondly, we execute the stochastic variables and initiate a recourse action based on the initial solution.

The PTSPD is considered to be ♯P-hard for Euclidean examples [3]. Numerous objective function approximations have been used in practical applications [4]. To solve difficult probabilistic combinatorial optimization issues, metaheuristics have been created. One of the most popular methods among them is Ant Colony Optimization (ACO), which was developed by Dorigo and Caro in [5]. The idea behind ACO was influenced by the pheromone-based trail-laying and trail-following behavior of actual ants. Since its inception in the 1990s, the number of researchers and ACO algorithms has grown rapidly, with all ACO algorithms sharing the same basic idea. However, ACO still has limitations such as long search times and local extremal. To overcome these limitations, some studies propose combining ACO with other techniques such as Levy flight [6].

The Levy flight (LF) mechanism has gained popularity as a search strategy in many metaheuristics optimization algorithms, due to its ability to help escape from being stuck in local optima. LF is named after French mathematician Paul Pierre Levy, who was the first to study Levy motion. Many animals' foraging movements have LF features, the majority of their feeding time is spent near a current food source, with occasional long-distance migration required to locate the next food source efficiently. It means that in ACO with LF, some ants will do long jumps in accordance with a Levy distribution to avoid becoming trapped in local optima. Several studies have proposed to combine ACO with LF to solve different combinatorial optimization problems, such as the traveling salesman problem [7], the heating route design [8]. Our goal is to suggest an ant colony optimization (ACO) metaheuristic that solve the PTSPD using the Levy flying method.

The rest of this paper is organized as follows: in Section 2, we present the mathematical formulation of the PTSPD; the ant colony optimization is presented and discussed in Section 3; Section 4 is devoted to ACO with Levy flight for the PTSPD. Computational experiments are presented in Section 5. Finally, in Section 6, we draw conclusions based on the results obtained from our study.

2. Model formulation

The PTSPD consists to find a tour that starts and ends at depot, visits each customer exactly once, skipping those that do not need to be visited, and respects the delivery dates and minimizes the expected length of the tour. To formalize this goal mathematically, let $N = \{1, \dots, n\}$ be a set of customers, and let 0 be a special element representing the depot. For each customer $i \in N$, we have a distance d_{ij} to any other customer $j \in N \cup \{0\}$ such that d_{ij} is the Euclidean distance formula, is defined as $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, where (x_i, y_i) and (x_j, y_j) are the coordinates of nodes i and j, respectively, and $d_{ij} = d_{ji}$. We consider p_i is a probability of needing service, a deadline l_i, and a penalty λ_i for missing the deadline. An a priori solution is then represented by a permutation $\xi\colon \{0, 1, \dots, n\} \to N \cup \{0\}$ such that $\xi(0) = 0$ and $\xi(i) \neq 0$ for all $i > 0$. The objective is to minimize the expected cost of the route, taking into account the uncertainties that affect the customers and the deadlines, which can be written as:

$$\mathbb{E}(\text{travel time for } \xi) + \mathbb{E}(\text{penalties for } \xi), \tag{1}$$

where $\mathbb{E}$ denotes the expected value operator.

The expected travel time can be computed in the same way as it is for the well-known PTSP mentioned in [1]:

$$\mathbb{E}(\text{travel time for } \xi) = \sum_{j=1}^{n} p_j d_{0j} \prod_{k=1}^{j-1}(1-p_k) + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} p_i p_j d_{ij} \prod_{k=i+1}^{j-1}(1-p_k) + \sum_{i=1}^{n} p_i d_{i0} \prod_{k=i+1}^{n}(1-p_k) \tag{2}$$

The equation (2) has three terms that correspond to the travel time from the depot to the first customer, from one customer to another, and from the last customer to the depot. Each there is weighted by the probability of visiting or skipping the customers.

We design a delivery service with fixed penalties for late arrivals. These penalties are applied when we have to reimburse the customers due to missed deadlines. To estimate the penalties, we use the expected value approximation method, which uses only the times of expected arrival if a visit is necessary, this method is suitable for fixed penalties, as Campbell's study proved [9].

The formula for expected penalties is simplified as we can find in [2]:

$$\mathbb{E}(\text{penalties for } \xi) = \sum_{i=2}^{n} p_i \lambda_i V_i. \tag{3}$$

Here, the indicator variable V_i takes a value of 1 if the expected arrival time at customer i is later than the target date and 0 otherwise, and λ_i is a fixed penalty value for violating the deadline.

3. Ant colony optimization for the PTSPD

3.1. Overview of ant colony optimization

Ant colony optimization (ACO) is a popular metaheuristic algorithm that draws its inspiration from the foraging habits of actual ants. In ACO, a team of artificial ants solves optimization problems by traversing a network or graph of interconnected nodes. To share information regarding the quality of their solutions, the algorithm utilizes pheromone trails, similar to the communication method employed by real ants. Since the introduction of the first ACO algorithm in the early 1990s, ACO algorithms and the number of researchers have been growing rapidly. All ACO algorithms share the same fundamental concept, Table 1 shows the list of ACO algorithms with their authors and years of publication.

Table 1. List of ACO algorithms.

Algorithm	Authors	Year
ACO	Deneubourg el al. [10]	1990
ANT SYSTEM (AS)	Dorigo et al. [11]	1991
ELITIST	Dorigo et al. [12]	1992
ANT-Q	Gambardella and Dorigo [13]	1995
ANT COLONY SYSTEM	Dorigo and Gambardella [14]	1996
MAX-MIN AS	Stützle and Hoos [15]	1996
ANT-CLASS	Monmarche et al. [16]	1999
BWAS	Cordon et al. [17]	2000
ANT-CLUST	Labroche [18]	2001
ANT-TREE	Azzag [19]	2004

3.2. Construction of ACO process for the PTSPD

The PTSPD Euclidean problem involves finding a closed tour that covers each node exactly once (excluding those that do not require a visit), begins and concludes at the depot, and satisfies a known deadline while minimizing the expected cost. Each node represents a customer, and each edge represents the distance between two nodes. The ACO process for the PTSPD consists of a population of m artificial ants. The number of ants at node i at time t is denoted by $b_i(t)$, where $i = 1, \dots, n$, and the total number of ants is calculated as $m = \sum_{i=1}^{n} b_i(t)$.

Each ant is an agent that carries out the following tasks: moving from node i to node j, leaving a pheromone trail one edge (i, j), and determining the next node to visit based on a probability influenced by the node's distance and the quantity of pheromone on the connected edge. Importantly, an ant avoids revisiting previously explored nodes until completing full tour to ensure the validity of the solutions.

The intensity of the pheromone trail at time t on edge (i, j) is represented by $\tau_{i,j}(t)$, the trail intensity undergoes an iterative update using a formula that includes a coefficient ρ representing trail disappearance and the change in pheromone $\Delta\tau_{i,j}(t, t+1)$, which is the amount of pheromone deposited by ants on edge (i, j) between times t and $t + 1$.

The ACO process facilitates the collaborative search of ants to discover high-quality solutions by strategically selecting paths that balance distance and pheromone information. Over iterations, the process adapts pheromone intensities to improve exploration and exploitation of potential solutions.

Tour construction. At the start of the construction process in ACO for the PTSPD, a set of m ants are placed at the depot. During each iteration, the ants will select a new city to visit based on a decision that takes into account both the local heuristic information η_{ij} and the pheromone trail strength $\tau_{ij}(t)$. Typically, the arc length is used to define the heuristic information η_{ij}, such as $\eta_{ij} = \frac{1}{d_{ij}}$. The following equation, can be used to calculate the probability that an ant k in city i will move to city j [20]:

$$p^k_{i,j}(t) = \begin{cases} \dfrac{[\tau_{i,j}(t)]^\alpha \eta^\beta_{i,j}}{\sum_{l \in N^k_i} [\tau_{i,l}(t)]^\alpha \eta^\beta_{i,l}} & \text{if} \quad j \in N^k_i, \\ 0 \quad \text{else.} \end{cases} \tag{4}$$

The significance of trail intensity, $\tau_{i,j}(t)$, and visibility, $\eta_{i,j}$, are determined by two parameters, α and β, respectively. The set of cities that the ant has not yet visited is represented by the feasible neighborhood of ant k, denoted by N^k_i. To generate a valid Hamiltonian cycle. The ants are permitted to return to their route after finishing the trip. This allows them to spread pheromone along the arcs they traveled.

Pheromone update. The pheromone values are modified at each iteration depending on the solutions generated by m ants. The following equation updates the pheromone associated with the edge between cities i and j, indicated by $\tau_{i,j}$ [21]:

$$\tau_{i,j} \longleftarrow (1-\rho)\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau^k_{i,j}. \tag{5}$$

The evaporation rate is ρ in this case, and the quantity of pheromone that ant k lay on the edge (i, j) is represented by $\Delta\tau^k_{i,j}$.

Ant-density, Ant-quantity and Ant-cycle are three alternative instances of the ant algorithm that result in three different possibilities for how to compute $\Delta\tau^k_{i,j}(t, t+1)$ and when to update $\tau_{i,j}(t)$. In the case of Ant-cycle, $\Delta\tau^k_{i,j}$ is computed only after a complete tour and is given by equation (6). The expression for $\Delta\tau^k_{i,j}$ given in [21] is defined as follows:

$$\Delta\tau^k_{i,j} = \begin{cases} \dfrac{Q}{\mathbb{E}(\text{travel time}) + \mathbb{E}(\text{penalties})} & \text{if} \quad (i,j) \in T^k(t), \\ 0 \quad \text{if else.} \end{cases} \tag{6}$$

Here, $T^k(t)$ is the path chosen by ant k at time t, Q is a fixed parameter and $\mathbb{E}(\text{travel time}) + \mathbb{E}(\text{penalties})$ represents the expected path length.

The value of evaporation rate coefficient ρ is set to a value less than 1 (typically between 0 and 1) to prevent excessive buildup of trail [11]. In our studies, a specific initial value is selected for the trail intensity $\tau_{i,j}(0)$ on each edge (i, j). Figure 1 presents the process to solve PTSPD by ACO algorithm.

4. Ant colony optimization with Levy flight for the PTSPD

ACO is a metaheuristic method used to solve stochastic combinatorial optimization problems such as the PTSPD. This method is inspired by the foraging behavior of ants when searching for food. The fundamental approach of ACO involves employing probabilistic path selection rules that are based on the quantity of pheromone present on the graph's edges, to guide the ants in their search for optimal solutions. However, this algorithm may be limited in its capacity to efficiently explore the search space, which can lead to getting trapped in suboptimal solutions.

To address these limitations, ACO with Levy Flight (LFACO) has been proposed as an extension of the basic ACO algorithm. This technique uses a foraging strategy inspired by Levy flight, to guide ants in the search for optimal solutions for the PTSPD. By using this strategy, ants can avoid becoming trapped in local suboptimal areas and search for more promising solutions in the search space.

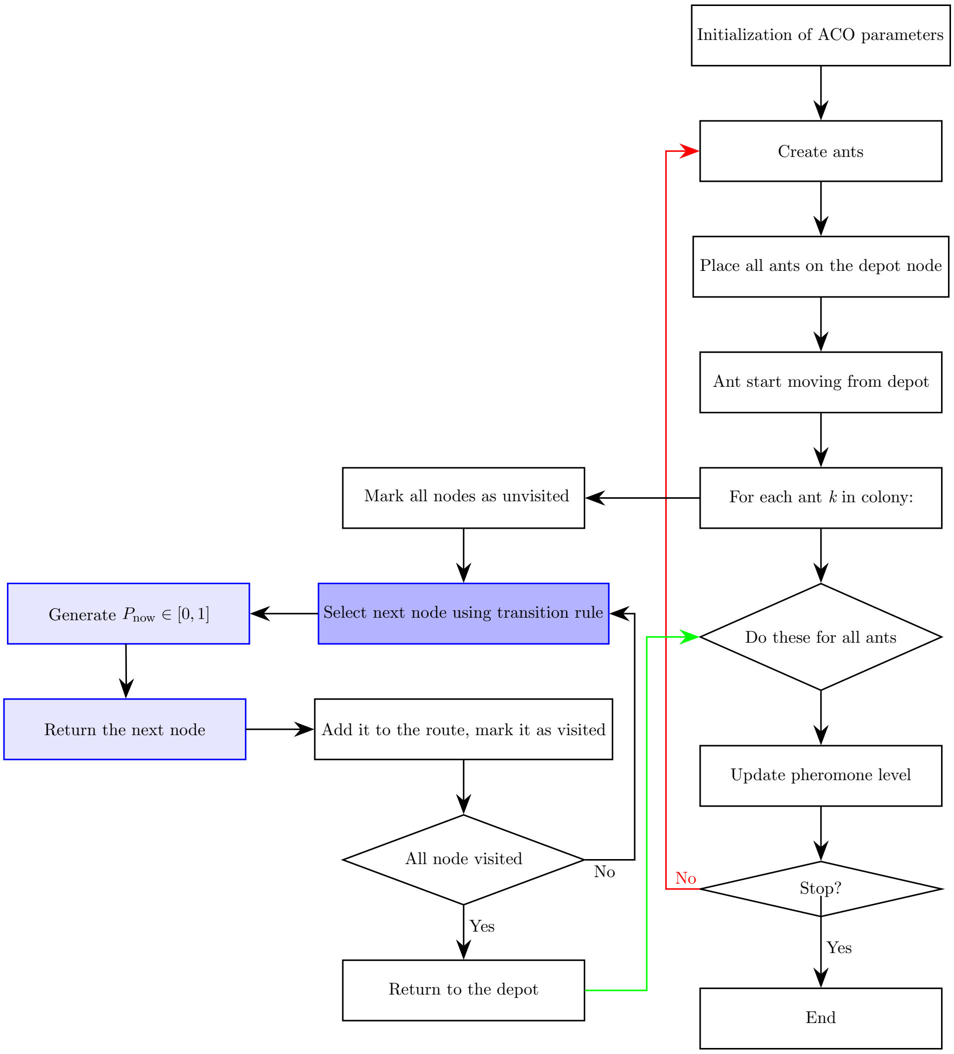

Fig. 1. Diagram illustrating the different steps of applying the ACO algorithm to solve the PTSPD.

4.1. ACO algorithm with Levy's flight

The steps of a random walk known as the Levy Flight follow a distribution known as the Levy distribution. In this distribution, the mean value is a specific type and the power law has an infinite variance [22]:

$$\text{Levy}(\beta) \sim u = t^{-\beta}, \quad 1 < \beta \leqslant 3. \tag{7}$$

There are multiple ways to generate random numbers using the Levy Flight method, but one of the easiest and most efficient is to use Mantegna's formulas to calculate the S step [23]:

$$S = \frac{u}{|v|^{\frac{1}{\beta}}}, \tag{8}$$

where u and v denote Gaussian centered distribution, and thus

$$u = N(0, \sigma_u^2), \quad v = N(0, \sigma_v^2) \tag{9}$$

with

$$\sigma_u^2 = \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma((1+\beta)/2)\,\beta\,2^{(\beta-1)/2}}, \quad \sigma_v^2 = 1, \tag{10}$$

where $\Gamma(z)$ is the Gamma function:

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt \tag{11}$$

calculating the Levy flight using formulas (7) to (11) can be complex and time-consuming for ACO algorithm. A proposed approach, Levy Flight-ACO (LFACO), introduces a conversion formula with formula (14) and modified version of formula (8) (now formula (12)) to address this issue [24]:

$$S_{\text{new}} = \begin{cases} \dfrac{1}{A} \cdot \dfrac{1 - P_{\text{threshold}}}{1 - P_{\text{Levy}}}, & \text{if} \quad S_{\text{new}} \geqslant 1; \\ 1, & \text{else}, \end{cases} \tag{12}$$

$$1 - P_{\text{new}} = \frac{1}{S_{\text{new}}} \cdot (1 - P_{\text{now}}), \tag{13}$$

$$P_{\text{new}} = \begin{cases} 1 - A \cdot \dfrac{1 - P_{\text{Levy}}}{1 - P_{\text{threshold}}} \cdot (1 - P_{\text{now}}), & \text{if} \quad P_{\text{Levy}} \geqslant P_{\text{threshold}}; \\ P_{\text{now}}, & \text{else}. \end{cases} \tag{14}$$

In this approach, S_{new} for Levy's flight is fixed with A (Levy flight Altering ratio) and $P_{\text{threshold}}$ (Levy flight threshold) as parameters. The turning on/off of Levy flight altering is represented by P_{Levy} and the original selection probability is represented by P_{now}. The resulting selection probability after Levy flight altering is P_{new}.

The Levy flight conversion process transforms step lengths higher than 1 into values between 0 and 1 using formulas (12) to (14). The Levy flight threshold and amplification ration are two predefined parameters required for the conversion. Formula (14) introduces a new selection probability to encourage exploration of a wider range of solutions and facilitate escape from local optima.

4.2. ACO with Levy flight for the PTSPD

This paper introduces a new ACO algorithm designed for the PTSPD described bellow.

Algorithm 1 Ant Colony Optimization with Levy flight for the PTSPD.

initialization: $t \leftarrow 0$; for each arc (i, j), initialize the intensity of the trail to $\tau_{ij}(0) \leftarrow \tau_0$;
Step 2: Starting node
for each ant k
  Place the ant k on the starting node (the station node) and store this information in Tabu_k;
Step 3: Build a tour for each ant
for i from 1 to n
  for k from 1 to m
    Choose the next node j, $j \notin \text{Tabu}_k$, where j is selected based on probability:
    $p^k_{i,j}(t) = \frac{[\tau_{i,j}(t)]^\alpha [\eta_{i,j}]^\beta}{\sum_{l \notin \text{Tabu}_k} [\tau_{i,l}(t)]^\alpha [\eta_{i,l}]^\beta}$
    Generate a uniform random $P_{\text{now}} \in [0, 1]$, $P_{\text{Levy}} \in [0, 1]$;
    if $P_{\text{Levy}} \geqslant P_{\text{threshold}}$ **then**
      $P_{\text{new}} = 1 - A \cdot \frac{1 - P_{\text{Levy}}}{1 - P_{\text{threshold}}} \cdot P_{\text{now}}$;
    else
      $P_{\text{new}} = P_{\text{now}}$;
    The next node will be selected using P_{new} from the candidate list;
    Store the chosen node in Tabu_k;
Step 4: Global update of the trail
Calculate the expected value of the tour, $E(\text{travel time}) + E(\text{penalties})$, for each ant k;
Apply the local improvement method to the routes of all k ants and recalculate $E(\text{travel time}) + E(\text{penalties})$;
for each edge $(i, j) \in \text{Cycle}^\star$
  Update the trail according to: $\tau_{i,j}(t + 1) = (1 - \rho)\tau_{i,j}(t) + \sum_{k=1}^{m} \Delta\tau^k_{i,j}(t)$
  with $\Delta\tau^k_{i,j}(t) = Q/(E(L_{\text{Cycle}^\star}) + E(\text{penalties}))$ if the ant k uses the (i, j) arc in its tour, and $\Delta\tau^k_{i,j}(t) = 0$
  otherwise;
$t \leftarrow t + 1$
Step 5: Conditions of termination
Memorize the shortest circuit found up to this point;
if not (end-test) **then**
  Empty all Tabu_k and go to **Step 2**;
else
  Stop;

The algorithm consists of two main components that guide the ants in finding the most optimal tours: pheromone intensity τ_0 and a tabu list tabu_k. The pheromone intensity determines the attractiveness of each edge, whereas the tabu list prevents ants from revisiting the same node. The algorithm updates the pheromone values based on the expected tour value. This process is repeated until the stopping condition is met. The Levy ACO algorithm modifies the candidate selection by utilizing the Levy flight step length to adjust the random number and generate diverse solutions. It incorporates parameters such as $P_{\text{threshold}}$ and A to enhance the algorithm's efficiency.

5. Computation experiment

In this section, we describe a computational study that aims to determine the efficacy of our newly developed Levy ACO algorithm in comparison with the conventional ACO algorithm. In our experiment, we consider the same instances that are given in [1, 25]. We used two sets of instances: the "Range" and "Mixed" data sets. The former have uniformly distributed probabilities for each customer, while the latter have random probabilities of 0.1 or 1 to represent scenarios where businesses of different sizes are served by the same delivery person. In both cases, we set a penalty of 5, to represent the costs related of failing to meet customer deadlines. The deadline of a task depends on its time window. If the time window has a non-zero opening time, the deadline is defined by that specific time. Otherwise, the deadline is equal to the closing time of the window.

We measure the computation time required to run each algorithm and evaluate the quality of the obtained solutions. The experimental setup is conducted in Python on a machine equipped with an Intel Core i5-7200U processor at 2.50 GHz, 8 GB of RAM, and a 64-bit operating system with an x64 processor.

Table 2. Comparison of ACO and LFACO based on expected values for the PTSPD.

Probability	Range		Mixed	
Data set	LFACO	ACO	LFACO	ACO
5	5.29	5.27	2.751	2.753
22	275.69	304.87	94.97	102.91
42	252.63	387.43	222.90	247.90
62	430.23	455.35	323.10	323.81
102	673.71	677.78	629.06	685.36

We use the same parameter settings for both algorithms: $\tau_0 = 0.01$, $m = 7$, $\alpha = 1$, $\beta = 2$, $\rho = 0.5$, $P_{\text{threshold}} = 0.8$, $A = 1$, and $Q = 1$. These values are determined based on the previous literature and empirical studies.

Table 2, show that LFACO outperforms ACO in terms of expected values for both Range and Mixed data sets, except for the case of data set 62 in the Mixed data set, where the difference between the two algorithms is negligible.

The difference in expected values between two algorithms is relatively small for small instances (e.g., data set 5), but it becomes more significant as the size of the problem increases (e.g., data sets 42, 62, and 102).

It is interesting to note that LFACO performs better than ACO even for data sets where the probabilities are uniformly distributed (i.e., Range data set), which suggests that the Levy Flight mechanism is effective in improving the algorithm's search capability beyond the effect of the probability distribution of the problem.

Table 3. Comparison of ACO and LFACO based on CPU values for the PTSPD.

Probability	Range		Mixed	
Data set	LFACO	ACO	LFACO	ACO
5	0	0	0	0
22	1	1	1	1
42	4	6	3	6
62	21	31	20	23
102	95	206	94	273

The performance difference between two algorithms is more pronounced for the Mixed data set, where the probabilities are randomly assigned with either a small or large value. This suggests that the ability of LFACO to explore the search space more efficiently is particularly useful for problems with heterogeneous probability distributions.

According to Table 3, LFACO requires less CPU time (in seconds) than ACO for all data sets and both Range and Mixed data sets. The difference is especially significant for larger instances (e.g., data sets 42, 62, and 102).

The difference in CPU time, between two algorithms is more pronounced for the Mixed data set, which suggests that the Levy Flight mechanism allows LFACO to find better solutions more efficiently in problems with a more complex probability distribution.

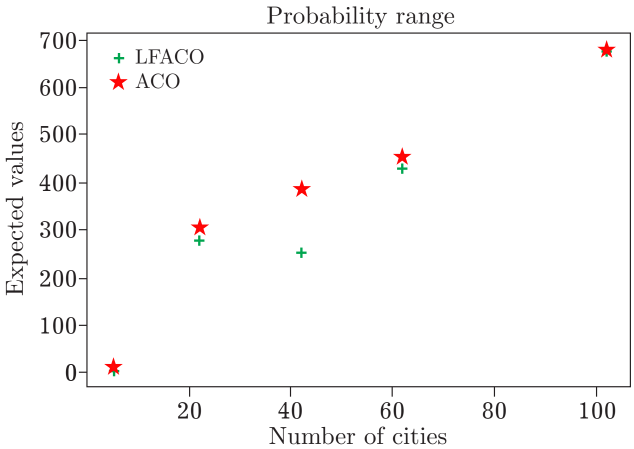

Fig. 2. Comparing ACO and LFACO for the PTSPD Using Expected Values and Probability Range Analysis.

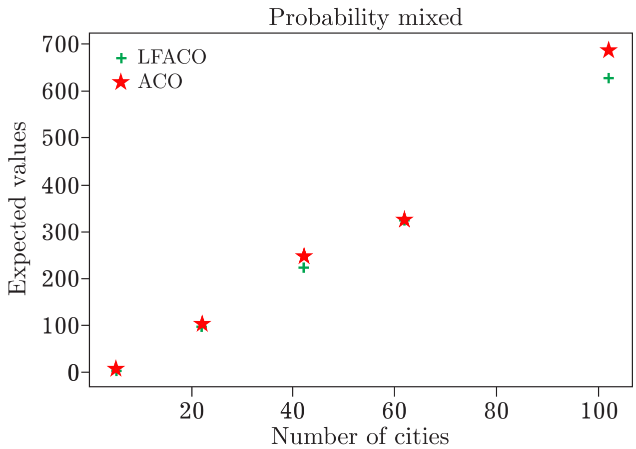

Fig. 3. Comparing ACO and LFACO for the PTSPD Using Expected Values and Probability Mixed Analysis.

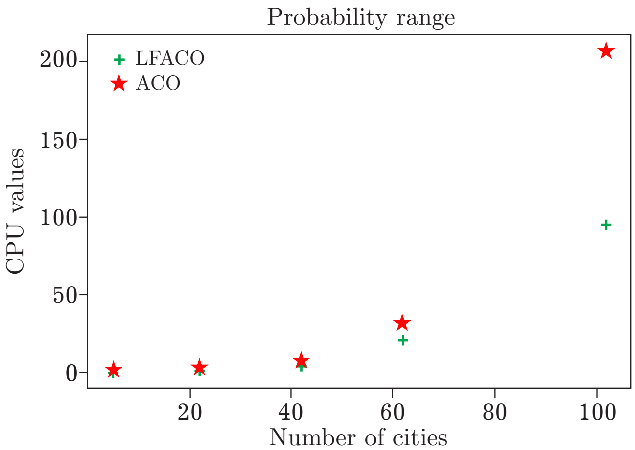

Fig. 4. Comparing ACO and LFACO for the PTSPD Using CPU Values and Probability Range Analysis.

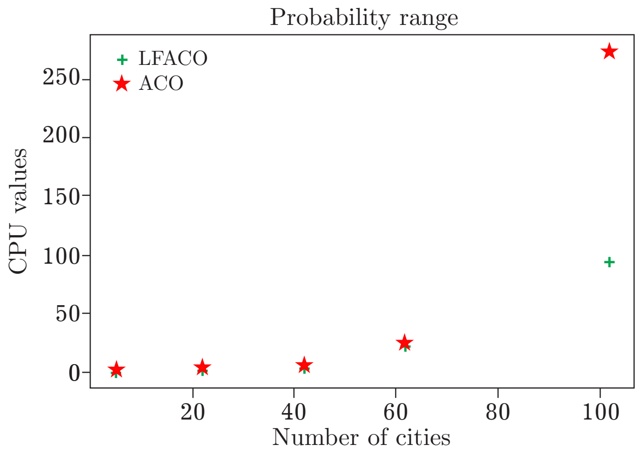

Fig. 5. Comparing ACO and LFACO for the PTSPD Using CPU Values and Probability Mixed Analysis.

Figures 2, 3 show that our LFACO algorithm outperforms the ACO algorithm in terms of expected values, for the Range and Mixed datasets, except for instance 62 of the Mixed dataset where the difference is negligible. This improvement is particularly significant, for larger instances.

Figures 4, 5 show the efficiency of the LFACO algorithm is enhanced for problems with heterogeneous probability distributions. Additionally, our algorithm requires less CPU time than the ACO algorithm for all instances and the Range and Mixed datasets, with this difference being particularly significant for larger instances of the Mixed dataset.

The obtained results corroborate previous studies that highlighted the difficulty of efficiently solving the probabilistic traveling salesman problem with deadlines, including the papers [1,25]. However, this current study provides a novel contribution by proposing an enhancement of ant colony optimization algorithm with Levy's flight mechanism to improve its performance beyond the effect of the probability distribution of the problem.

6. Conclusion

In this paper, we are interested in solving the probabilistic traveling salesman problem with deadlines (PTSPD). Through the course of our computational experiments, we have concluded that the Ant Colony Optimization with Levy Flight (LFACO) algorithm is a promising approach for PTSPD. Our results show that LFACO outperforms ACO in terms of expected values for both Range and Mixed datasets with a relatively small difference for small problem instances and a more significant difference for larger and more complex instances. Moreover, LFACO requires less CPU time than ACO for all datasets, especially for larger instances. These findings suggest that LFACO is an efficient and accurate algorithm for finding optimal routes for a salesman to visit a set of locations with a probability of being visited while meeting the deadlines imposed on each customer. Furthermore, the use of Levy flight in the ACO algorithm appears to be an effective way to enhance the search capability of the algorithm and enhance its performance beyond the effect of the probability distribution of the problem.

[1] Campbell A. M., Thomas B. W. Probabilistic traveling salesman problem with deadlines. Transportation Science. **42** (1), 1–21 (2008).

[2] Weyland D. On the computational complexity of the Probabilistic Traveling Salesman Problem with Deadlines. Theoretical Computer Science. **540**, 156–168 (2014).

[3] Weyland D., Montemanni R., Gambardella L. M. Hardness results for the probabilistic traveling salesman problem with deadlines. International Symposium on Combinatorial Optimization. 392–403 (2012).

[4] Weyland D., Montemanni R., Gambardella L. M. Heuristics for the probabilistic traveling salesman problem with deadlines based on quasi-parallel Monte Carlo sampling. Computers & Operations Research. **40** (7), 1661–1670 (2013).

[5] Dorigo M., Di Caro G. Ant colony optimization: a new meta-heuristic. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406). **2**, 1470–1477 (1999).

[6] Liu Y., Cao B. Improving ant colony optimization algorithm with Levy flight. MISTA 2017 Conference (2017).

[7] Liu Y., Cao B. A Novel Ant Colony Optimization Algorithm With Levy Flight. IEEE Access. **8**, 67205–67213 (2020).

[8] Zhang Y., Zhao H., Cao Y., Liu Q., Shen Z., Wang J., Hu M. A hybrid ant colony and cuckoo search algorithm for route optimization of heating engineering. Energies. **11** (10), 2675 (2018).

[9] Campbell A. M., Thomas B. W. Runtime reduction techniques for the probabilistic traveling salesman problem with deadlines. Computers & Operations Research. **36** (4), 1231–1248 (2009).

[10] Deneubourg J. L., Aron S., Goss S., Pasteels J. M. The self-organizing exploratory pattern of the Argentine ant. Journal of Insect Behavior. **3** (2), 159–168 (1990).

[11] Dorigo M., Maniezzo V., Colorni A. The ant system: An autocatalytic optimizing process. Technical Report (1991).

[12] Dorigo M. Optimization, Learning and Natural algorithms. PhD thesis, Politecnico di Milano (1992).

[13] Gambardella L. M., Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem. Machine Learning Proceedings 1995. 252–260 (1995).

[14] Gambardella L. M., Dorigo M. Solving symmetric and asymmetric TSPs by ant colonies. Proceedings of IEEE International Conference on Evolutionary Computation. 622–627 (1996).

[15] Stützle T., Hoos H. H. Improving the Ant System: A detailed report on the MAX–MIN Ant System. FG Intellektik, FB Informatik, TU Darmstadt, Germany, Tech. Rep. (1996).

[16] Monmarché N. On data clustering with artificial ants. AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions. 23–26 (1999).

[17] Cordon O., de Viana I. F., Herrera F., Moreno L. A new ACO model integrating evolutionary computation concepts: The best-worst Ant System. Proc. ANTS. 22–29 (2000).

[18] Labroche N., Monmarché N., Venturini G. A new clustering algorithm based on the chemical recognition system of ants. ECAI. **77**, 345–349 (2002).

[19] Azzag H., Venturini G. A clustering model using artificial ants. Universite Francois-Rabelais (2004), (in France).

[20] Stützle T., Hoos H. H. MAX–MIN ant system. Future Generation Computer Systems. **16** (8), 889–914 (2000).

[21] Dorigo M. Ant colony optimization. Scholarpedia. **2** (3), 1461 (2007).

[22] Feyel P. Optimisation des correcteurs par les métaheuristiques. Application à la stabilisation inertielle de ligne de visée. PhD thesis, CentraleSupélec (2015).

[23] Zhou Y., Ouyang X., Xie J. A discrete cuckoo search algorithm for travelling salesman problem. International Journal of Collaborative Intelligence. **1** (1), 68–84 (2014).

[24] Liu Y., Cao B., Li H. Improving ant colony optimization algorithm with epsilon greedy and Levy flight. Complex & Intelligent Systems. **7** (4), 1711–1722 (2021).

[25] Weyland D., Montemanni R., Gambardella L. M. A metaheuristic framework for stochastic combinatorial optimization problems based on GPGPU with a case study on the probabilistic traveling salesman problem with deadlines. Journal of Parallel and Distributed Computing. **73** (1), 74–85 (2013).

Комбінована оптимізація мурашиної колонії з механізмом польоту Леві для імовірнісної задачі комівояжера з термінами виконання

Ель Асрі Ф., Таяні К., Фахурі Х.

Команда SMAD, Полідисциплінарний факультет Лараша, Університет Абдельмалека Ессааді, Тетуан, Марокко

У цій статті нас цікавить ймовірнісна задача комівояжера з термінами виконання (PTSPD), де необхідно зв'язатися з клієнтами, на додаток до їх випадкової доступності, до встановленого кінцевого терміну. Основна мета — знайти оптимальний маршрут, який охоплює випадкову підмножину відвідувачів у тому самому порядку, в якому вони з'являються під час туру, намагаючись зробити шлях якомога коротшим. Ця проблема вважається ♯P-складною. Оптимізація мурашиної колонії (ACO) часто використовується для вирішення цієї складної залачі оптимізації. Однак у цьому дослідженні пропонуємо розширений ACO, використовуючи алгоритм польоту Леві. Це дозволяє деяким мурахам робити довші стрибки на основі розподілу Леві, допомагаючи їм уникнути локальних оптимальних ситуацій. Наші обчислювальні експерименти з використанням стандартних контрольних наборів даних демонструють, що запропонований алгоритм ефективніший і точніший, ніж традиційний ACO.

Ключові слова: *стохастична комбінаторна оптимізація; імовірнісна задача комівояжера; терміни виконання; метаевристика, оптимізація мурашиних колоній; політ Леві.*