



HAL
open science

A Hybrid Algorithm Based on Multi-colony Ant Optimization and Lin-Kernighan for solving the Traveling Salesman Problem

Mathurin Soh, Nguimeya Tsofack, Tayou Clémentin

► **To cite this version:**

Mathurin Soh, Nguimeya Tsofack, Tayou Clémentin. A Hybrid Algorithm Based on Multi-colony Ant Optimization and Lin-Kernighan for solving the Traveling Salesman Problem. *Revue Africaine de Recherche en Informatique et Mathématiques Appliquées*, In press. hal-03646847v1

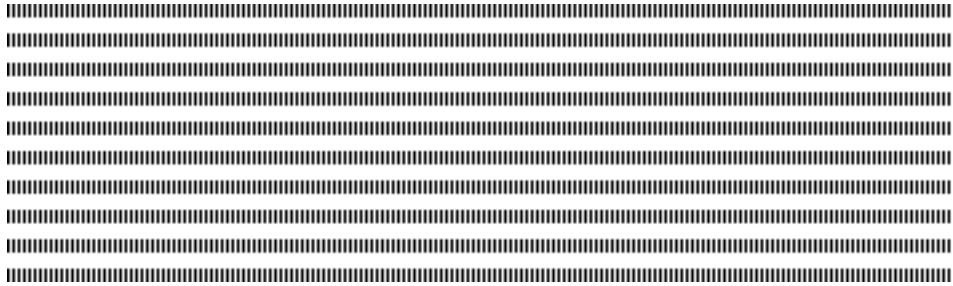
HAL Id: hal-03646847

<https://hal.science/hal-03646847v1>

Submitted on 30 Oct 2021 (v1), last revised 30 Jul 2023 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



A Hybrid Algorithm Based on Multi-colony Ant Optimization and Lin-Kernighan for solving the Traveling Salesman Problem

Mathurin Soh, Baudoin Nguimeya Tsofack, Clémentin Tayou Djamegni

Research Unit in Fundamental Informatics, Engineering and Applications
University of Dschang
P.O. Box 67 Dschang, Cameroon
mathurinsoh@gmail.com* ,
nguimeyabaudoin@yahoo.fr,
dtayou@gmail.com



RÉSUMÉ.

ABSTRACT. In this article, a hybrid heuristic algorithm is proposed to solve the traveling salesman problem (TSP). This algorithm combines two main metaheuristics: optimization of multi-colony ant colonies (MACO) and Lin-Kernighan-Helsgaun (LKH). The proposed hybrid approach (MACO-LKH) is a so-called insertion and relay hybridization. It brings two major innovations: The first consists in replacing the static visibility function used in the MACO heuristic by the dynamic visibility function used in LKH. This has the consequence of avoiding long paths and favoring the choice of the shortest paths more quickly. Hence the term insertion hybridization. The second innovation consists in modifying the pheromone update strategy of MACO by that of the dynamic λ -opt mechanisms of LKH in order to optimize the solutions generated and save in execution time. Hence the relay hybridization. The significance of the hybridization, is examined and validated on benchmark instances including small, medium, and large instance problems taken from the TSP library. The results are compared to four other state-of-the-art metaheuristic approaches. It results in that they are significantly outperformed by the proposed algorithm in terms of the quality of solutions obtained and execution time.

MOTS-CLÉS : Colonie, Fourmis, Heuristique Multi-colonie, Problème du Voyageur de Commerce

KEYWORDS : Ants, Colony, Heuristic, Multi-colony, Traveling Salesman Problem



1. Introduction

The field of distribution or collection of goods, logistics face optimization problems like the famous Traveling Salesman Problem (TSP). The TSP is a problem in combinatorial optimization studied in operational research, computational mathematics, and artificial intelligence. The Traveling Salesman Problem particularly attracts the attention of many researchers in recent years. Indeed, the search for exact or optimal solutions to TSP remains a challenge for scientific community. To date, we note that several works have already been subject of intense research related to the TSP. Among these works, we note the efficiency of heuristics and metaheuristics for the resolution of this combinatorial optimization problem and other NP-difficult problems. In literature, we find heuristics and metaheuristics such as Ants Colony Optimization (ACO) which are capable of solving small instances of TSP. Despite these advances, the TSP remains difficult to solve when the size of the instances increases. It is a trend to combine ACO with other algorithms to solve very large scale of the traveling salesman problem. With the concept of hybridization that we adopt for this work, the exploitation of several resolution techniques is a new opportunity offered to researchers in the field.

In this paper, we present a new hybrid model of ACO with multiple colonies by LKH. Our aim is to use the insertion and parallel hybridization that we explain below, to hybridize the ACO heuristic with the LKH heuristic in order to decrease the computation time of the sequential MACO heuristic and to a certain extent improve the quality of the solutions obtained in the TSP resolution. We opted for insertion hybridization because it is new and efficient. The rest of the work is organized as follows:

Section I provides a non-exhaustive state of the art of TSP resolution methods. We also briefly give a formulation of the traveling salesman problem.

Section II presents in a detailed manner, the meta-heuristics used in the paper framework. Section III is devoted to the study of hybrid methods. It gives an overview of the different types of hybridization and emphasizes above all the type of hybridization we used: insertion hybridization and parallel hybridization.

And finally section IV presents the hybrid approach MACO-LKH the tests and the results obtained.

We concluded our work by summarizing the main results obtained and giving new perspectives on the basis of the work carried out.

2. Problem Statement

The Traveling Salesman Problem is defined as follows: given n points (cities) and the distances between each point, find a path of minimum total length that passes exactly once through each point and back to where we started. The distance can also be seen as the cost in general. This combinatorial optimization problem, therefore, consists in searching for the best solution among several possible choices. However, it is easy to state but difficult to solve. The problem is to determine a tour or Hamiltonian circuit, i.e. passing once and only once through the n cities, and that is of minimum cost. It is classified as an NP-difficult problem because it is not known a method of resolution that can provide accurate solutions in one reasonable time for large jurisdictions (large number of cities) to address the problem. For these large instances, one is very often satisfied with the

approximate solutions because after an explicit enumeration, the number of Hamiltonian paths is equal to $(n-1)!/2$ [10].

Mathematically, the TSP can be formulated as follows: Let n cities and C_{ij} , the cost or distance corresponding to the $i \rightarrow j$ trip. Let the variable X_{ij} , which is 1 if the tour contains the trip $i \rightarrow j$, and 0 otherwise. The problem is spelled as follows: [6]:

$$\left\{ \begin{array}{l} MinZ = \sum_{i=1}^n \sum_{j=1}^n C_{ij} X_{ij} \\ \sum_{j=1}^n X_{ij} = 1 \quad \forall i \\ \sum_{i=1}^n X_{ij} = 1 \quad \forall j \\ \sum_{i \in Q} \sum_{j \in Q} X_{ij} \geq 1 \quad \forall Q \\ X_{ij} \in \{0, 1\} \forall i, \forall j \end{array} \right. \quad (1)$$

Where Q represents a subset of $1, \dots, n$ and its complement. The constraints (c) express that the permutation of the n cities must be a turn, i.e. it cannot ...exist in the underbelly.

The Travelling Salesman Problem has direct applications in transportation, networks and logistics. For example, finding the shortest route for pickup buses school or in industry, for collection/distribution, to find the shortest distance that the mechanical arm of a machine will have to travel to drill holes in a printed circuit board[1, 7, 10].

3. State of the art

The TSP was one of the first NP-hard problems to be investigated, research began at Princeton University in the 1930 [21]. The problem has since attracted many researchers and thousands of different approaches and algorithms have been developed. In this section, we will focus on hybrid methods. Hybridization is a trend observed in much work done on metaheuristics in recent years to solve TSP. It makes it possible to take advantage of the advantages in one to fill what is seen as a limit in the other. In the literature there are two classes of hybridizations: on the one hand we have Hybridization between metaheuristics and exact methods. This is the case of Cotta[14] which proposes a hybridization between a genetic algorithm and the exact Branch and Bound method to replace the recombination operator. Let us also note the case of, Jahira who proposes hybridizations between a genetic algorithm and an exact method to solve the traveling salesman problem [17]. In this algorithm, the recombination operator is replaced by a Branch and Bound algorithm.

In the same vein, Chabrier et al [16] hybridized a local search with a Branch and Price algorithm to solve the problem of vehicle routes[14]. The execution of algorithms is carried out in parallel while keeping communication between the methods. Mavrovouniotis, Muller, and Yang [23] integrated the memetic ACO algorithm with local search operators to improve solutions in the population. They apply these local search operators to the best solution found in the population in order to possibly improve this solution; a similar idea is used in this article, but instead of local search operators, LKH is used.

on the other hand, we have the Hybridization between metaheuristics and metaheuristics. Among the works carried out in this direction, we have that of Martin and Otto [17] who inserted the descent method in a simulated annealing algorithm to solve the traveling salesman problem. This type of hybridization is referred to as low-level relay hybridization in which another algorithm is incorporated to form a new algorithm. Stützle and Hoos [18] incorporate a local search function in an ant colony algorithm to solve the traveling

salesman problem. This low-level co-evolutionary hybridization consists of incorporating one or more single solution-based metaheuristics into a solution population metaheuristic. The advantage of this type of hybridization is that it compensates for the operating power of a local search and that exploration of a global search. Fotso Laure et al in 2008 [3] propose two new hybrid heuristics for the TSP. The first between a Genetic (AG) and heuristic (LK) algorithm and the second between the ant colony algorithm (ACS) and heuristics (LK). The heuristics obtained were called respectively AG-LK and ACS-LK [1]. These authors use a single colony. The results of this experiment have sufficiently demonstrated the effectiveness of these hybrid approaches. on several instances of TSP [3]. Unfortunately the solution time resulting from this experience remains enormous. To improve the efficiency of the hybrid heuristics proposed by Fotso et al, Nguimeya, et al in 2016 implemented two new hybrid heuristics for the TSP. The first between a Genetic algorithm (AG) and heuristics (LKH) which is an improvement of LK by helsgaun [2] and the second between the ant colony algorithm (ACS) and heuristics (LKH). The heuristics obtained were called respectively "AG-LKH" and "ACS-LKH". The hybridization strategies differ from one author to another [1].

4. Presentation of some heuristics and meta-heuristics used to solve the TSP

Metaheuristics are a family of stochastic methods which consist in solving optimization problems. One of the advantages of these is their ability to optimize a problem from a minimum amount of information, however they do not offer any guarantee as to the optimality of the best solution found. Metaheuristics are gaining more and more popularity and are constantly evolving. As a result, a large number of metaheuristic classes currently exist. We can cite:

4.1. Ant colony algorithm

This metaheuristic is inspired by collective depositing and tracking behaviors observed in ant colonies [20]. In fact, ants communicate with each other indirectly by depositing chemical substances, called pheromones, on the ground. This type of indirect communication is called stigmergy. Indeed, if an obstacle is introduced on the path of the ants, the latter will, after a search phase, all tend to take the shortest path between the nest and the obstacle. The higher the pheromone level in a given location, the more likely an ant will be attracted to that area. The ants that reached the nest the fastest through the food source were those that took the shortest branch of the route. The Algorithm is the following [19]:

4.2. Principle of the approach Multi Ant Colony Optimization (MACO)

This is a new version of ACO that we developed in a previous work [19] and that we use in this work as a starting algorithm for the design of our hybrid algorithm.

Step 1: Construction of the path by each ant (Solution)

Initially (at time $t = 0$), the algorithm positions m ants on n cities and the intensity of the trace for all pairs of cities (ij) is set to a small positive value T_0 in the pheromone matrix. A taboo list is maintained to ensure that a city cannot be visited twice during the same round. Each ant k will therefore have its own list of V_k -tabu cities which will keep

Algorithm 4.1 : ACO Algorithm

```
1 Début
2   Entry m: number of ants per colonies ;
3   n: number of cities ;
4    $N_c \leftarrow 0$  ;
5   Initialize tabuList // with the starting citie of each ant ;
6   Initialize the matrix  $\tau_{ij}$ 
7   While (  $N_c < N_{max}$ ) and ( convergence not reached) do
8     For  $i \leftarrow 1$  Do  $n$  Do
9       For  $j \leftarrow 1$  Do  $m$  Do
10        Select the citie j to be added or tour in progress according to the
11        formula  $p_{ij}^k(t)$  ;
12        Perform the local update of the track according to the citie pair (i,
13        j) ;
14      EnFor
15    EnFor
16    For each ant  $k$  Do
17      Evaluate the solution k for each of the steps ;
18      Insert the solution k in the listTABOU ;
19      Perform the Global update of the trace according to the best solution of
20      the cycle ;
21    EnFor
22     $NC \leftarrow NC + 1$  ;
23 End
```

in memory the cities already visited. During one iteration of the algorithm, several ants take turns visiting a sequence of cities. A cycle (NC) is completed when the last of the m ants has completed its construction.

After the initialization phase, starting from this sequence of cities already visited, within the different colonies, the ants move this time on the different nodes of the graph according to a probability and therefore the equation is that of formula (1). It allows colonies to favor the shortest paths during the different iterations. During a round, each ant k of a colony L records in its tabu list (memory) the list of cities already visited. The probabilistic formula for the selection of a node by the ant k of colony L is defined by the expression of $p_{ij}^{k,l}(t)$ (2).

$$p_{ij}^{k,l}(t) = \begin{cases} \frac{[\eta_{ij}]^\beta [\tau_{ij}(t)]^\alpha}{\sum_{u \in v_{k_l}} [\eta_{iu}]^\beta [\tau_{iu}(t)]^\alpha} & \text{if } j \in v_{k_l} \\ 0 & \text{else} \end{cases} \quad (2)$$

Where v_{k_l} represents the set of nodes or cities not visited by ants from colony L.

Step 2: pheromone deposit

When an ant moves from citie i to citie j, it leaves a certain amount of pheromone (value) on the arc (ij). A matrix which is the pheromone matrix records information about the use of the arc (ij). At each step of the turn this matrix is updated so that the more this use has been important in the past, the greater the likelihood that these bows will be used again in the future. The evolution of the update equation is as follows:

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \quad (3)$$

The evaporation equation of the pheromone matrix is described by the formula

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) \quad (4)$$

The inverse of the distance between cities $\eta_{ij} = \frac{1}{C_{ij}}$ called visibility is static information used to guide the choice of ants to nearby cities and avoid too many cities. distant. The α and β parameters are used to determine materiality of the intensity of the trace and of the visibility in the construction of a solution. This is based on a compromise between visibility (η_{ij}) and the quantity of pheromone (τ_{ij}) present between i and j at cycle t. These parameters are the same as those used in OCF [2].

4.3. Procedure of Lin and Kernighan (LK)

4.3.1. The basic algorithm

The K- opt algorithm is based on the K- Optimality concept:

Definition: A visit is said to be k optimal (or simply k - Opt) if it is impossible to get a shorter visit by replacing k links with any other set of k links [17]. From this definition, it is obvious that every optimal k-visit is also K ' optimal for $1 < k' < k$. It's also easy to see that a tour that contains n cities is optimal if and only if it is n - optimal. Unfortunately, the number of operations to test all k-exchanges is increasing rapidly. In a naive implementation, testing a k-link exchange has a time complexity of $O(n^k)$. Accordingly, the values $k = 2$ and $k = 3$ are commonly used. It is a disadvantage that k must be specified in advance because it is difficult to know which k to use to achieve the best compromise between current time and quality of solution. Lin and Kernighan corrected this drawback by introducing a powerful k variable algorithm which changes the values of k during its execution, by deciding on each iteration what the value of k should be. At each step of the iteration the algorithm examines, for ascending values of k, whether and exchange of k links can make it possible to obtain a shorter visit. At each step the algorithm considers an increasing set of potential exchanges (starting with $k = 2$). If the crawl is successful in finding a new, shorter visit, then the actual visit is replaced with this new visit. With a feasible visit, the algorithm performs exchanges that repeatedly reduce the length of the current visit, until a visit is reached and no other exchange can improve it [14].

4.4. Lin - Kergnighan - Helsgaun algorithm

It is the modified and extended version of LK algorithm. indeed A central rule in the original algorithm is the heuristic rule which restricts the inclusion of links in the visit to the five nearest neighbors to a given citie. This rule directs the search to a shorter visit and reduces the search effort substantially. However, there is a certain risk of not being able to find the optimal solution. Helsgaun amends this rule.

5. Study Of Hybrid Methods

Hybridization is a technique which consists in combining the characteristics of two different methods to derive the advantages of the two methods [4] [10]. In the literature, the hybridization of metaheuristics can be divided into two main parts: hybridization of metaheuristics with metaheuristics and hybridization of metaheuristics with exact methods [15] [10].

Algorithme 4.2 : MACO Algorithm

```
1 Début
2   Entry m : number of ants per colony ;
3   L : number of colonies ;
4   n : number of cities ;
5    $N_c \leftarrow 0$  ;
6    $D_{ij} \leftarrow 0$  // Global matrix ;
7    $d_{ij} \leftarrow t_0$  // local matrix of pheromons ;
8   Initialize listeTaboue // with the origin cite of each ant ;
9   // The initialisation of the Tabouliste and the matrix Dij
10  Place each colony at random in a starting point (cite) ;
11  // Parallel construction of the towers by the different colonies ;
12  For  $K \leftarrow 1$  à  $m$  Do
13    Construction of a round by each ant at random;
14    Gradual deposition of pheromones in the Dij matrix of each colony;
15    Evaluation and selection of the best colony;
16    Initializing the Dij matrix with the best colony;
17  EnFor
18  // Construction of the optimal solution ;
19  The ants are placed in each origin cite ;
20  While ( $N_c < N_{max}$ ) do
21    For  $i \leftarrow 1$  à  $n$  Do
22      For  $j \leftarrow 1$  à  $L$  Do
23        For  $K \leftarrow 1$  à  $m$  Do
24          Select the cite  $V_i$  to be added or tour in court according to
          formula (2);
25          Evaluate the solution of ant K on route (i,j) ;
26          Perform global track update according to cite pair (i,j). if it's
          better than the previous ants according to formula 3 and 4
          (evaporation) ;
27          Insert as you go (i,j) into the taboo list so as to construct the
          solution to K.'s problem gradually;
28        EnFor
29      EnFor
30    EnFor
31  EndWhile
32   $S \leftarrow$  the best solution: Each colony provides a partial solution ;
33 End
```

5.0.1. Hierarchical classification of metaheuristics

This classification is characterized by the level and mode of hybridization. The level of hybridization can be low (Low-Level) or high (High-Level) [2]. In the low level, a metaheuristic replaces an operator of another method which encompasses it. On the other hand, in high level hybridization, each metaheuristic keeps its property during hybridization [15] [2]. Each level of hybridization generates two modes of cooperation namely, relay mode and co-evolutionary mode. In relay mode, the methods are executed sequentially, that is to say the result of the first method is the start of the following method [15]. When the different methods work in parallel to explore the search space, we speak of co-

evolutionary mode. The combination of modes and levels gives four classes of hybridization which are: low-level relay hybridization, low-level co-evolutionary hybridization, high-level relay hybridization and high-level co-evolutionary hybridization [2].

5.0.2. Low-level relay hybridization

It encompasses single solution-based metaheuristics in which another method is incorporated to form a new algorithm[2].example: the descent method can be inserted in a simulated annealing algorithm.

5.0.3. Low-level co-evolutionary hybridization

It consists in incorporating one or more metaheuristics based on a single solution in a metaheuristic with a population of solutions [4] [15]. The advantage of this type of hybridization is to compensate for the exploitation power of a local search and that of the exploration of a global search[2] .

5.0.4. High-level relay hybridization

It takes place when metaheuristics are used sequentially i.e. the final solution (s) of the first metaheuristic is the initial solution (s) of the following metaheuristic [15]. In this procedure, all the methods keep their integrity. Example introduced Taboo research at the end of a genetic algorithm to improve the solutions obtained [15].

5.0.5. high level coevolutionary hybridization

In this case, the metaheuristics used work in parallel by exchanging information between them in order to find the optimal solution of the problem posed [2] [15].

5.0.6. Flat classification of hybrid metaheuristics

This is another classification of hybrid methods characterized by the type of hybridized methods, their field of application and the nature of their functions [15]. According to the type of hybridization, one finds homogeneous hybridized methods where the algorithms used are based on the same metaheuristic and heterogeneous hybridized methods where the metaheuristics used are different [15]. The field of application of hybridized metaheuristics makes it possible to distinguish two main classes of hybridization, global hybridizations and partial hybridizations. Global hybridization takes place when all the hybridized methods are applied to the whole search space[15].

Partial hybridization, on the other hand, breaks down a problem into sub-problems where each has its own search space.

6. PROPOSED METHOD: MULTI ANTS COLONY HYBRID ALGORITHM OF ANT - LKH (MACO-LKH)

6.1. Hybridization Aspects to exploit on replace

In this part, we discuss the structure of our algorithm. It is known that the MACO or LKH algorithms are very efficient for difficult problems and TSP in particular [4], but the disadvantage of MACO is the consumption of time. On the other hand , LKH must always start its execution with a good starting solution.

In the MACO algorithm described previously, for each ant, the path of a cities i to a city j depends on:

- The list of cities already visited, which gives the possible choices at each transition, when the ant k is over the cities ;
- The inverse of the distance between cities called visibility. η_{ij}
- The amount of pheromone deposited on the ridge connecting two cities, called intensity of the track. This quantity defines the attractiveness of a track, and it is modified after the passage of an ant.

The multi-colony MACO algorithm as described above is made up of three main parts:

- initialization;
- Construction of paths (Generations of paths) according to the probabilistic formula (2);
- pheromone deposition and evaporation: Evaluate the solution of each ants in each colonies.

To hybridize the two heuristics **MACO** and **LKH**, the visibility function in the MACO heuristic is replaced by the mechanisms of the LKH heuristic to be more efficient in restricting the nearest neighbors which results in the gain in execution time. Then the pheromone track update process is replaced by the λ -opt mechanisms used by LKH heuristics in order to accelerate the convergence towards the optimal solution and in another register optimize the optimal solution.

6.2. Hybridization of LKH and MACO: (MACO-LKH)

We opted for an insertion hybridization. That is, we hybridize the MACO and LKH heuristics by inserting LKH into MACO. in clear the main characteristics are:

- The visibility equation between cities i and j disappears. This information is replaced by the mechanisms of the LKH process in order to direct the ants more quickly to nearby towns and thus avoid too long journeys.
- After each round of any ant, there is no longer any direct deposition of pheromone at the end, but rather triggering of the LKH process on this round until the best possible round of ant k and then pheromone deposition according to MACO equation(3).

In fact, the algorithm is defined in two steps:

Step 1: MACO algorithm

- Generation of solutions by ants from different colonies;
- pheromone deposit.

Step 2: Heuristic LKH

- Optimization of tourneys ;
- Injection of the best ant into the communication matrix.

the algorithm is the following:

Algorithme 6.1 : MACO-LKH ALGORITHM

```
1 Début
2   Entry m: number of ants per colonies ;
3   L: number of colonies ;
4   n: number of cities ;
5   T: the displacement step ;
6    $T_c \leftarrow n$  ;
7    $N_c \leftarrow 0$  ;
8    $D_{ij} \leftarrow 0$  ;
9    $d_{ij} \leftarrow t_0$  / local pheromone matrix ;
10  Initialize tabuList // with the starting citie of each ant ;
11  // Initialization:
12  // initialization Pheromone = t0; // Place each colony at random in a starting
    point;
13  j  $\leftarrow$  1 to L
14  K  $\leftarrow$  1 to m Randomly build a tour by each ant ;
15  Application of the LKH method after each T transition ;
16  Pheromone deposit by the best ant ;
17  // Solutions construction
18  While ( $N_c < N_{max}$ ) do
19    for i  $\leftarrow$  1 to n do
20      for j  $\leftarrow$  1 to L do
21        K  $\leftarrow$  1 ;
22        While ( $K < m$ ) do
23          For each ant K of colony L carry out the transition T by
            selecting for each transition the city j to be added to the
            current turn according to the probabilistic formula (2) of
            MACO and using the dynamic visibility function of LKH.;
24          K = k + 1 ;
25        EndWhile
26        Application of the LKH Process on each partial solutions
            obtained ;
27        // Pheromone deposit
28        Perform the global update of the trace according to the best ant of
            colony L according to equations 3 and 4 after performing the
            dynamic  $\lambda$ -opt operations of LKH on the partial solution
29        ;
30      end for
31    end for
32     $N_c \leftarrow N_c + 1$  ;
33  EndWhile
34  S  $\leftarrow$  the best solution ;
35 End
```

7. RESULTS OF TESTS AND COMPARISON WITH OTHER METHODS

In this section, we present the numerical results obtained by the proposed algorithm MACO-LKH. The interest is to show that MACO-LKH gives better results.

7.1. Implementation environment

The MACO-LKH approach has been implemented in C language on instances of the TSPLIB online instance library for the TSP on a server with the following characteristics: *4 GHZ processor, 08 GHZ RAM, 500 GO DD* after 100 executions.

7.2. Results - Discussions and Analysis

By making a comparative study between the heuristic MACO-LKH with the heuristics ACS- LKH, AG, Multi-colony ACO, AG -LKH which make it from the best heuristics for the TSP of the literature, and under identical test conditions, we obtain the results of TAB1 and TAB2 below:

Instances TSP	SIZE	AG	ACS-LKH	AG-LKH	MACO	MACO-LKH
Lin 105	105	14379	14379	14379	14379	14379
Pr124	124	5977.5	58537	58537	58537	58537
LIN318	318	44235	42029	42029	42029	42020
att532	532	ID	276787.7	276790	276787.07	276787
ALi535	535	ID	202339	202339	202339	202339
rat783	783	ID	88060	88060	88060	88060
std1655	1655	ID	62128.6	63120.00	62128.0	62128
Vm1748	1748	ID	336557	3365557.02	336557	336557
pr2392	2392	ID	378034.25	37803.8	378032.2	378032.00
Usa13509	13509	ID	19884705.00	ID	19849706	19849705.001
pla33810	33810	ID	ID	ID	ID	ID

Tableau 1. Comparison of MACO-LKH calculation costs with other Algorithms

The results comparison tables show that the multi-colony MACO and MACO-LKH approaches are better. the MACO-LKH approach is better than all the other methods studied in runtime since the optimal solution is quickly reached and the algorithm stops without necessarily reaching the maximum number of cycles. At least 99 percent of the optimal solution is always achieved.

For small and medium instances, the difference between the execution times of the hybrid algorithm with the other algorithm is insignificant. For large instances (usa13509 and Pr2392), there is a significant difference in the execution time of the hybrid heuristic compared to other heuristics (AG-LKH, ACO-LKH, MOCF). This is thanks to some simple improvements to be made in MOCF during hybridization (for example, limiting the search to a number of the nearest neighbors thanks to the mechanisms of LKH.)

Instances TSP	SIZE	AG	ACS-LKH	AG-LKH	MACO	MACO-LKH
Lin 105	105	0.039	0.0315	0.0300	0.0300	0.0300
Pr124	124	0.920	0.920	0.46	0.03	0.029727
LIN318	318	8.1	0.8	1.600	0.0341	0.034
att532	532	121.1	15.629	13.741	11.7400	11.73456
ALi535	535	ID	30.94	38.26224	1.9489	1.267002
rat783	783	ID	28.886	12.095	28.886	11.1293
std1655	1655	ID	128.7337	151.53	117.076	117.0705
Vm1748	1748	ID	71.865	111.07	0.97254	0.94214
pr2392	2392	ID	119.524	114.100	97.2540	94.214
Usa13509	13509	ID	2113.3	ID	2110.39	163.3311
pla33810	33810	ID	ID	ID	ID	ID

Tableau 2. Comparison of MACO-LKH calculation times with other Algorithms

8. Conclusion

In this work, we have proposed a new hybrid multi-colony ant-LKH heuristic (MACO-LKH). It has been applied to solve the TSP. Compared to other heuristics, MACO-LKH is able to provide better solutions than ACS-LKH, MACO, AG-LKH heuristics. The proposed MACO-LKH algorithm shows improvements both in costs and in execution times. The tests and comparisons carried out prove MACO-LKH is competitive with the best heuristics of the hour for the TSP. A limit linked to our approach will undoubtedly be the use of much more resources (memory and processor). We also find that although hybrid heuristics and metaheuristics provide solutions of good quality in reasonable time to the TSP, they remain however always costly in computing time for certain instances of problems (pla33810). As a perspective, we believe that the parallelization of the heuristic search processes is imperative to reduce the execution time and improve the quality of the solutions provided. Extension of the MACO-LKH algorithm for other NP problems such as Vehicle Routing Problems, Multi-Depot Vehicle Routing Problem and their variants.

9. Bibliographie

- [1] DEHAN MICHAËL « Distribution en Belgique des médicaments issus du plasma sanguin: une étude de cas », *université catholique de Louvain*, G. REINELT, « A traveling salesman problem library », *ORSA J. Comput.*, 1991.
- [2] AHMIA IBTISSAM, AL, « Une nouvelle metaheuristique pour les problèmes d'optimisation combinatoire : la Monarchie Metaheuristique », *Université des sciences et de la technologie Houari Boumediène, Phd Thesis*, pp 1-5, 2019
- [3] K. HELSGAUN, « An effective implementation of the Lin-Kernighan traveling salesman heuristic », *IEEE Transactions on Evolutionary Computation*, Ochoa S.F., Roman GC. (eds) pp 106-130, 2000.
- [4] SOH, BAUDOIN NGUIMEYA, L.P. FOTSO, « Algorithmes hybrides pour la résolution du problème du voyageur de commerce », *Proceedings of CARI 2016*,
- [5] TSPLIB95, « Traveling Salesman Problem Library », <http://www.iwr.uniheidelberg.de/iwr/comopt/software/TSPLIB95>, December 23, 2020.
- [6] T. STUETZLE, « The Traveling Salesman Problem: State of the Art », *TUD SAP AG Workshop on Vehicle Routing*, Ochoa S.F., Roman GC. (eds) July 10, 2003.
- [7] BP. DELISLE, « Parallélisation d'un algorithme d'optimisation par colonies de fourmis pour la résolution d'un problème d'ordonnement industriel », *IEEE*, pp 5366, 2002
- [8] D.S. JOHNSON, L.A. MC GEOCH, « The traveling salesman problem: A case study in local optimization », *Local Search in Combinatorial Optimization*, H.L. Aarts and J.K. Lenstra (eds) 1997.
- [9] I. ALAYA, « Optimisation multi-objectif par colonies de fourmis Cas des problèmes de sac à dos », *PhD Thesis, Université de la Manouba*, 2009.
- [10] B. TADUNFOCK TETI, L.P. FOTSO, « Heuristiques du problème du voyageur de commerce », *Proceedings of CARI 2006*, Fowler, D. and Dawson, L. (eds.) pp 1-8, 2006.
- [11] M. DORIGO, L. M. GAMBARDILLA, « Ant Colony System : A cooperative learning approach to the traveling salesman problem », *IEEE Transactions on Evolutionary Computation*, Ochoa S.F., Roman GC. (eds) vol. no 1(1) no pp 53-66, 1997.
- [12] S. LE DIGABEL, « Problème du voyageur de commerce (TSP) », *Roman GC*. 2018.

- [14] C. AUDET, J.E.JR. DENNIS, , « Mesh adaptive direct search algorithms for constrained optimization », , *SIAM Journal on Optimization* 2006 .
- [15] HANAË HACHIMI, , « Hybridations d'algorithmes métaheuristiques en optimisation globale et leurs applications », , *THÈSE DE DOCTORAT, Université Mohammed* 2013 .
- [16] EMILIE DANNA AND DAVID L, , « How to select a small set of diverse solutions to mixed integer programming problems) », , *Operations Research Letters* 2009.
- [17] R. MARTIN., , « Single-interval learning by simile within a simulated hebbian neural network. », , *Computers Mathematics with Applications* 1990.
- [18] THOMAS STUTZLE AND HOLGER H. HOOS., , « Max-min ant system. », , *Future Generation Computer Systems* 2000.
- [19] M.SOH, B.NGUIMEYA, C.TAYOU DJAMEGNI, , « A Multi Ant Colony Heuristic Approach For Solving The Traveling Salesman Problem », *Revue Africaine de la Recherche en Informatique et Mathematique Appliquees INRIA, Volume 34*, 2021, DOI:10.46298/arima.6752
- [20] M. DORIGO, V. MANIEZZO, ., , « and A Coloni. Ant system : optimization by a colony of cooperating agents. », *IEEE Trans. on Man. Cyber. Part B*, , 2020.
- [21] XIAOXIA ZHANG., LIXIN TANG, « A New Hybrid Ant Colony Optimization Algorithm for the Traveling Salesman Problem », *IThe Logistics Institute, Northeastern University, Shenyang, China*, 2008.
- [22] OLIEF ILMANDIRA RATU FARISI1, BUDI SETIYONO2, R. IMBANG DANANDJOJO3, , « A Hybrid Firefly Algorithm ? Ant Colony Optimization for Traveling Salesman Problem », *IThe Logistics Institute, Northeastern University, Shenyang, China*, 2015.
- [23] PETR STODOLA , KAREL MICHENKA , JAN NOHEL, MARIAN RYBANSKÝ, « Hybrid Algorithm Based on Ant Colony Optimization and Simulated Annealing Applied to the Dynamic Traveling Salesman Problem », *Department of Intelligence Support, University of Defence, Kounicova* , 2020.