Multi-Hybrid Algorithm for Searching Solutions to Optimization Problems: Case of Traveling Salesman Problem (TSP)

Nguimeya Tsofack Baudoin

October 30, 2020

UNIVERSITY OF DSCHANG

Enterprise, Research and Development Forum (EREDEF-2020)

Artificial Intelligence, Digital Economy and African Transformation

Algebra • Analysis • Computer Science

mailto://dept.math-info@univ-dschang.org

MULTI-HYBRID ALGORITHM FOR SEARCHING SOLUTIONS TO OPTIMIZATION PROBLEMS: CASE OF TRAVELING SALESMAN PROBLEM(TSP)

NGUIMEYA TSOFACK BAUDOIN

PhD Student in ***Computer Science*** *(Option:* ***Networks and distributed services****)*

LIFA

nguimeyabaudoin@yahoo.fr

ADDITIONAL INFORMATIONS:
Matricule number: CM04$_8$0SCI0404
Level: D3
Year of first registration: 2017 - 2018

RESEARCH FIELDS:
OPERATIONAL RESEARCH

SUPERVISORS:
Clémentin Tayou Djamegni
MC, University of Dschang

ABSTRACT: In this paper, we are interested in finding solutions to the travelling salesman problem(TSP) by proposing a multi-hybrid algorithm between a multi-colony ant colony algorithm (OCF) and the Optimization Towers heuristic (LKH). The proposed approach combines both insertional hybridization which is a new and very efficient approach and parallel hybridization. The goal of this hybridization is to reduce the computation time as well as the improvement of the optimal solution. The two algorithms move in parallel synchronously while communicating with each other. To assess the effectiveness of the proposed metaheuristic, we examined its performance on instances of the TSPLIB online instance library for the TSP. The experimental results indicate that the hybrid heuristic that we call (MOCFLKH - TSP) competes with the other existing methods in the literature by being illustrated by a great improvement in the minimization of computation time and computation costs.

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

1. Introduction

Among the optimization problems, the travelling salesman problem (TSP) particularly attracts the attention of many researchers in recent years,in fact, the search for exact or optimal solutions to TSP remains a real problem in IT. To date, we note that several works have already been the subject of intense research at 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.Despite these advances, the TSP remains difficult to solve when the size of the instances increases. With the concept of hybridization that we adopt from 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 OCF with multiple colonies by LKH. Our aims is to use the insertion and parallel hybridization that we explain below, to hybridize the OCF heuristic with the LKH heuristic in order to decrease the computation time of the sequential MOCF 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 **MOCFLKH-TSP** 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. State of the art

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 cite bib17. 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. 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. Stûtzle 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].

2.0.1. The Traveling Salesman Problem (TSP)

The Traveling Salesman Problem (TSP) 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 turn 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 does not know 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 Trade Traveler’s Problem can be formulated as follows: Let n cities and C_{ij}, the cost or distance corresponding to the $i \to j$ trip. Let the variable X_{ij}, which is 1 if the tour contains the trip $i \to j$, and 0 otherwise. The problem is spelled as follows: [6]:

$$\begin{cases} MinZ = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}X_{ij} & \\ \sum_{J=1}^{n} X_{ij} = 1 \quad \forall i & (1) \\ \sum_{i=1}^{n} X_{ij} = 1 \ \ \forall j & (2) \\ \sum_{i\in Q}^{n}\sum_{j\in Q}^{n} X_{ij} \geq 1 \ \ \forall Q & (3) \\ X_{ij} \in \{0,1\} \forall i, \forall j & \end{cases} \quad (1)$$

Where Q represents a subset of 1,..,n and its complement. The constraints (3) 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. 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:

3.1. OCF 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]:

3.2. Principle of the approach (OCF Multi colonies (MOCF)

This is a new version of OCF 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 To 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 Vk-tabu cities which will keep 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

Algorithme 3.1 : ALGORITHME OCF

```
Début
    Entry m: number of ants per colonies ;
    n: number of cities ;
    N_c ← 0 ;
    Initialize tabuList // with the starting city of each ant ;
    Initialize the matrix τ_ij
    While ( N_c < N_max) and ( convergence not reached) do
        For i ← 1 Do n Do
            For j ← 1 Do m Do
                Select the city j to be added or tour in progress according to the
                  formula p^k_ij (t) ;
                Perform the local update of the track according to the city pair (i, j) ;
            EnFor
        EnFor
    EndWhile
    For each ant k Do
        Evaluate the solution k for each of the steps ;
        Insert the solution k in the listTABOU ;
        Perform the Global update of the trace according to the best solution of the
          cycle ;
    EnFor
    NC ← NC + 1;
End
```

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 $^{k,l}_{ij}$ (1).

$$p_{ij}^{k,l}(t) = \begin{cases} \frac{[\eta_{ij}]^{\beta}[\tau_{ij}(t)]^{\alpha}}{\Sigma u \in v_{k_l} [\eta_{iu}]^{\beta}[\tau_{iu}(t)]^{\alpha}} & \text{If I} \in v_{k_l} \\ 0 & (1) \end{cases}$$

(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 city i to city 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:

$$\tau_{ij}(t+1) = \rho\, \tau_{ij}(t) + \triangle\, \tau_{ij} \text{ (2) [19]}$$

(3)

The evaporation equation of the pheromone matrix is ??described by the formula (3)

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

(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 (t_{ij}) present between i and j at cycle t. These parameters are the same as those used in OCF [2].

3.3. Procedure of Lin and Kernighan (LK)

3.3.1. 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 an 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].

Algorithme 3.2 : MACO ALGORITHM

```
Début
    Entry m : number of ants per colony ;
    L : number of colonies ;
    n : number of cities ;
    N_c ← 0 ;
    D_ij ← 0 // Global matrix ;
    d_ij ←t_0 // local matrix of pheromons ;
    Initialize listeTaboue // with the origin city of each ant ;
    // The initialisation of the TabooList and the matrix Dij ;
    Place each colony at random in a starting point (city) ;
    // Parallel construction of the towers by the different colonies ;
    For K ← 1 à m Do
        Construction of a round by each ant at random;
        Gradual deposition of pheromones in the Dij matrix of each colony;
        Evaluation and selection of the best colony;
        Initializing the Dij matrix with the best colony;
    EnFor
    // Construction of the optimal solution ;
    The ants are placed in each origin city ;
    While ( N_c < N_max) do
        For i ← 1 à n Do
            For j ← 1 à L Do
                For K ← 1 à m Do
                    Select the city V_i to be added or tour in court according to
                      formula (1);
                    Evaluate the solution of ant K on route (i,j) ;
                    Perform global track update according to city pair (i,j). if it's
                      better than the previous ants according to formula 5.2 and 5.3
                      (evaporation) ;
                    Insert as you go (i,j) into the taboo list so as to construct the
                      solution to K.'s problem gradually;
                EnFor
            EnFor
        EnFor
    EndWhile
    S ← the best solution: Each colony provides a partial solution ;
End
```

3.4. Lin - Kergnighan - Helsgaun algorithm

It is the modified and extended version of LK's 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 city. 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.

4. 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].In this section we propose a classification of types of hybridization.

4.0.1. 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 [2]. On the other hand, in high level hybridization, each metaheuristic keeps its property during [15] [2] hybridization. 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 [2] 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].

4.0.2. Low-level relay hybridization

It encompasses single solution-based metaheuristics in which another method is incorporated to form a new [2] algorithm.

4.0.3. Low-level coevolutionary 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 [2] search.

4.0.4. 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].

4.0.5. 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].

4.0.6. 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 of the [15] search space. Partial hybridization, on the other hand, breaks down a problem into sub-problems where each has its own search space. At the end of this section we note in a summary way, 03 major types of hybridization: In parallel, in series and in insertion.

5. PROPOSED METHOD: MULTI COLONY HYBRID ALGORITHM OF ANT - LKH (MOCFLKH -TSP)

In this part, we will discuss the structure of our algorithm. It is known that the OCF or LKH algorithms are very efficient for difficult problems and TSP in particular

The idea of ??insertion hybridization came to minimize the computation time. However, we have chosen to hybridize the two heuristic algorithms MOCF and LKH by inserting the LKH in MOCF and by evolving the two heuristics in parallel. n other words, after a fixed number T of iteration by each of the n ants of the m colonies, the LKH tower improvement heuristic is inserted to replace the MOCF evaluation function in order to improve the pseudo optimal solutions if possible. OCF or MOCF algorithms are known to be very effective for difficult problems such as TSP, but sometimes their downside is time consuming. As they are heavy and sometimes greedy, We propose the idea of ??parallel hybridization of the MOCF algorithms by The LKH. From this experiment, we notice improvements, obtained at the level of costs and also the minimization of computation time. The structure of our algorithm baptized "MOCFLKH-TSP" is as follows. The multi-colony MOCF algorithm as described above is made up of four main parts (initialization, construction of towers, deposition and evaporation of pheromones, evaluation of solutions). Its disadvantage is that it is slow and greedy (memory, processor) when the size of the population becomes large, it may have the advantage of allowing greater exploration of the search space. To do this, we hybridized two heuristic algorithms MOCF multi colony and LKH by inserting the LKH in OCF multi colony.

Clearly, after exploring the solutions using MOCF, to accelerate the process of diversification of solutions on the one hand and intensification on the other hand, the LKH procedure replaces the probabilistic function used by the MOCF procedure and then follows. pheromone deposits and global communication matrix update. To allow ants to have the global state of the pheromone matrix in real time, we develop the two metaheuristics in pseudo parallel ways: After a fixed number T ($3 < T <= n$) of transitions, the MOCF process breaks down by the ant k and the LKH process takes over and triggers the turn intensification procedure on the various intermediate solutions. Once all the nodes (towns) have been visited, the ants return to their starting point and we move on to the next cycle. the algorithm is as follows:

INSTANCES TSP	TAILLE	AG	ACS_LKH	AG_LKH	OCM-multi colonie	MOCFLKH_TSP
LIN105(100)	**105**	14379	14379	14379	14379	14379
Pr124	**124**	5977.5	58537	58537	58537	58537
LIN318	**318**	44235	42029	42029	42029	42026
Attr532	**532**	ID	276787.700	276791.00	276787.07	276780.00
Atti537	**535**	ID	202339	202339	202339	202339
Rat783	**783**	ID	88060	88060	88060	88060
Std1655	**1655**	ID	62128.6	63128.00	62128.6	62128.004
Vm1748	**1748**	ID	336557	336557.02	336557	336557.0
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

Figure 1: Comparison of MOCFLKH -TSP calculation costs with other Algorithms

5.1. RESULTS OF TESTS AND COMPARISON WITH OTHER METHODS

In this section, we present the numerical results obtained by the proposed algorithm MOCFLK-TSP. The interest is to show that MOCFLK-TSP gives better results. The MOCFLKH-TSP 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, 01 TB DD* after 100 executions. By making a comparative study between the heuristic MCOFLKH-TSP with the heuristics ACS- LKH, AG, Multi-colony OCF, 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 fig1 and fig2 below:

The results comparison tables show that the multi-colony MOCF and MOCFLKH-TSP approaches are better. the MOCFLKH-TSP 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.

INSTANCES TSP	TAILLE	AG	ACS_LKH	AG_LKH	OCM-multi colonie	MOCFLKH_TSP
LIN105	**105**	0.03115	0.03115	0.03115	0.0300	0.0300
Pr124	**124**	0.92	0.92	0.46	0.03	0.029727
LIN318	**318**	1.8	0.8	1.600	42029	0.34
Attr532	**532**	121.1	15.62916	13.74344	11.7400	11.73456
Atti537	**535**	ID	30.948184	38.26224	1.940893	1.267002
Rat783	**783**	ID	2.8886	3.2095	2.8886	1.129335
Std1655	**1655**	ID	128.7337	151.5335	117.07694	117.07694
Vm1748	**1748**	ID	71.86591	111.07	41.8657	41.8113
Pr2392	**2392**	ID	1.19524	1114.100	0.972540	0.94214
Usa13509	**13509**	ID	2113.3	ID	2110.397	163.3311
Pla33810	**33810**	ID	ID	ID	0	0

Figure 2: Comparison of MOCFLKH - TSP calculation times with other Algorithms

6. Conclusion and perspectives

In this work, we have proposed a new hybrid multi-colony ant-LKH heuristic (MOCFLKH -TSP). The proposed MOCFLKH-TSP algorithm shows improvements both in costs and in execution times. The tests or comparisons carried out prove MOCFLKH-TSP is beautiful and very 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 resolution time and improve the quality of the solutions provided.

References

[1] DEHAN MICHAËL Distribution en Belgique des médicaments issus du plasma sanguin: une étude de cas , *universite catholique de louvain .*, 3(4), p 50-73 , 2018.

[2] AHMIA IBTISSAMAL, Une nouvelle metaheuristique pour les problemes d'optimisation combinatoire : la Monarchie Metaheuristique, *Universite des sciences et de la technologie houari boumediene, Phd Thesis ,* no 1(1), pp 1-5, 2019

[3] K. HELSGAUN, An effective implementation of the Lin-Kernighan traveling salesman heuristic, *AIEEE Transactions on Evolutionary Computation*, *Ochoa S.F., Roman GC. (eds)* no 12, pp 106-130, 2000.

[4] B. NGUIMEYA TSOFACKM. SOHL.P. FOTSO, Algorithmes hybrides pour la résolution du problème du voyageur de commerce, *Proceedings of CARI 2016*, *Ochoa S.F., Roman GC. (eds)* , pp 63-74, 2016.

[5] TSPLIB95, Traveling Salesman Problem Library , *http://www.iwr.uniheidelberg.de/iwr/comopt/software/TSPLIB95*, , Last visit:December 23,2019.

[6] T. STUETZLE, The Traveling Salesman Problem: State of the Art, *TUD SAP AG Worshop on Vehicle Routing*, *Ochoa S.F., Roman GC. (eds)* 12, july 10, 2003.

[7] BP. DELISLE, Parallélisation dun algorithme dOptimisation par Colonies de Fourmis pour la résolution dun problème dordonnancement industriel, *IEEE*, 12, pp 5366, 2002

[8] D.S.JOHNSONL.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 TETIL.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. MGAMBARDELLA, Ant Colony System : A cooperative learning approach to the traveling salesman problem , *AIEEE 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. (eds) vol. no 1(1)* 2018 .

[14] C. AUDET, J.E.JR. DENNIS, Mesh adaptive direct search algorithms for constrained optimization , , *SIAM Journal on Optimization* 2006 .

[15] HANAÂ 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] SOH, NGUIMEYA, TAYOU, MODELES ET APPROCHES MATAHEURISTIQUES POUR LA RECHERCHE DES SOLUTIONS OPTIMALES AUX PROBLEMES NP-DIFFICILE : APPLICATION AU PROBLEME DU VOYAGEUR DE COMMERCE (TSP) , *CARI*, *UDs* 2020.

[20] M. DORIGO, V. MANIEZZO, ., and A Colorni. Ant system : optimization by a colony of cooperating agents. , *IEEE Trans. on Man. Cyber. Part B,*, 2020.

Algorithme 5.1 : MOCFLKH ALGORITHM-TSP

Début

Entry m: number of ants per colonies ;
L: number of colonies ;
n: number of cities ;
T: the displacement step ;
$T_c \leftarrow 3$;
$N_c \leftarrow 0$;
$D_{ij} \leftarrow 0$;
$d_{ij} \leftarrow t_0$ / local pheromone matrix ;
Initialize tabuList // with the starting city of each ant ;
// `Initialization:`
// initialization Pheromone = t0; // Place each colony at random in a starting point;
j ← 1 to L
K ← 1 to m Randomly build a tour by each ant ;
Application of the LKH method after each T transition ;
Pheromone deposit by the best ant ;
// Tower construction
While *($N_c < N_{max}$)* **do**
    For *$i \leftarrow 1$ to n* **Do**
        For *$j \leftarrow 1$ to L* **Do**
            K ← 1 ;
            While *($K < m$)* **do**
                For each ant K des L colony Carry out T transition by Selecting for each transition The city j to be added or turn in progress according to the probabilistic formula (1) ;
                Application of the LKH Process on each partial solutions obtained ;
                Insert progressively (i, j) in taboo list so as to build gradually the solution of K ;
                // Pheromone deposit
                Perform the global update of the trace according to the best ant of the L colony according to equations 2 and 3 ;
                K = k + 1 ;
            EndWhile
        EnFor
    EnFor
    $N_c \leftarrow N_c + 1$;
EndWhile
S ← the best solution ;

End