University of Mohamed El Bachir El Ibrahimi of Bordj Bou Arreridj
Faculty of Mathematics and Computer Science

Master's Thesis

Specialty: **OPERATIONAL RESEARCH**

THEME

(ACO+GPE) for solving the traveling salesman Problem (TSP)

Presented by:

SOUADIA KHAOULA ICHRAK

DJABOU BOUTHEINA

In front of the jury composed of:

Président :	*Mr. MAACHE Salah*
Examiner :	*Mr. RAMDANI Zoubir*
Superviser :	*Mr. SAHA ADEL*

Promotion: 2020/2021

Abstract

Today, optimization methods inspired from nature are widely used to solve optimization problems, due to their ability to provide innovative solutions for complex problems and it becomes more effective especially if we use hybrid methods. Therefore, we proposed a new model (ACO+GPE) to solve the traveling salesman problem. We applied the model to real problems, then we compared the results with the results of another model (ACO+PSO+3-OPT).

KEY WORDS: Ant colony optimization, general pairwise exchange, hybrid methods, the traveling salesman problem

Résumé

Aujourd'hui, les méthodes d’inspiration biologique sont largement utilisées pour résoudre des problèmes d’optimisation, en raison de leur capacité à apporter des solutions innovantes à des problèmes complexes et elle devient plus efficace surtout si l'on utilise des méthodes hybrides. Par conséquent, nous avons proposé un nouveau modèle (ACO+GPE) pour résoudre le problème du voyageur de commerce. Nous avons appliqué le modèle sur des problèmes réels, puis on a comparé les résultats avec les résultats d'un autre modèle (ACO+PSO+3-OPT).

Mots-clés : optimisation par colonies de fourmis, GPE, des méthodes hybrides, le problème du voyageur de commerce

الملخص

اليوم ، تُستخدم طرق التحسين المستوحاة من الطبيعة على نطاق واسع لحل مشكلات التحسين ، نظرًا لقدرتها على تقديم حلول مبتكرة للمشكلات المعقدة وتصبح أكثر فاعلية خاصة إذا استخدمنا طرقًا هجينة. لذلك ، اقترحنا نموذجًا جديدا لحل مشكلة البائع المتجول. قمنا بتطبيق النموذج على مشاكل حقيقية ، ثم قمنا بمقارنة النتائج بنتائج نموذج آخر.

للكلمات المفتاحية طرق حساب مشتقة من الطبيعة طرق هجينة مشكلة البائع المتجول

Promotion: 2020/2021

Thanking

First of all, we thank "God" who gave us the wisdom and health to do this modest work.

We thank our supervisor MR SAHA ADEL for his guidance and supervision in the preparation of this work.

Our thanks also go to all members of the jury for having accepted to honor our work with their judgment, and for their jury participation.

Promotion: 2020/2021

Dedication

I dedicate this modest work:

To my parents, my dear father NACER and my dear mother SALIHA for their encouragement and prayers throughout my studies, for all that they had done to have this result.

To my sister RADHIA for her encouragement and her help.

To my brothers ABD SALEM and ABD NOUR.

To my colleague SOFIANE, and I really appreciate his encouragement and his help.

To my friend INSAF for her support.

To all my big family.

To my friend and partner BOUTHAINA.

ICHRAK

Promotion: 2020/2021

Dedication

I gladly wish to dedicate this modest work:

To the dearest person in my life, my mother.

To the one who made me my woman, my father.

To my dear sisters and brother, a source of joy and happiness.

To all my cousins, a source of hope and motivation.

To the most adorable child Rouen.

To all my friends.

To ICHRAK, my dear friend before she became a partner, for all the good times shared together, thank you for being there.

BOUTHEINA

Promotion: 2020/2021

CONTENTS TABLE

Chapter I combinatorial optimization

Chapter 3: IMPLIMENTATION & RESULTS

FIGURE LIST

TABLES LIST

ABBREVIATION LIST

(ERP)	**E**nterprise **R**esource **P**lanning
(SCM)	**S**upply **C**hain **M**anagement
(ACO)	**A**nt **C**olony **O**ptimization
(GPE)	**G**eneral **P**airwise **E**xchange
(PSO)	**P**article **S**warm **O**ptimization
(TSP)	The **T**raveling **S**alesman **P**roblem
(PLNE)	**L**inear **O**ptimization In **I**ntegers
(POC)	**P**roof **O**f **C**oncept
(NIOA)	**N**ature-**I**nspired **A**lgorithms
(GA)	**G**enetic **A**lgorithm
(ABC)	**A**rtificial **B**ee **C**olony

GENERAL INTRODUCTION:

Operational research is a scientific method of decision support. Its history is recent: it dates back to the Second World War. It was in England that this discipline received its name and proved its effectiveness by bringing together scientists and military officers responsible for preparing major decisions related to operations. At the end of the war, the success of operational research techniques has continued to spread among the range of decision areas.

Operational research is a vast branch of mathematics which encompasses many diverse areas of minimization and optimization. It is the discipline of developing quantitative tools to assist decision makers with these often complex decisions. Also, it is the most often used to analyze complex real life problems typically with the goal of improving or optimizing performance, and it will provide a clear formulation of the criteria guiding the choices, Thus, rationalizing the decision-making process.

Doing operational research consists in the practice of mathematically modeling a given problem and then solving the modeled problem, the first step requires know-how and experience. Second, we have rigorous algorithms. The discipline has developed with computer science: mathematically modeling complex problems would be useless if we did not have computers to carry out the calculations.

The field operational research provides a more powerful approach to decision making than ordinary software and data analytics tools. Employing operational research professionals can help companies achieve more complete data tests, consider all available options, predict all possible outcomes and estimate risk. Additionally, operational research can be tailored to specific business processes because business managers face an endless list of complex issues every day. They must make decisions about financing, where to build a plant, how much product to manufacture, how many people to hire, and so on. Often the factors that make up business are complicated and they may be difficult to comprehend. Operational research is a way to deal with all these thorny problems.

Operational research can be applied to a variety of use cases, including: scheduling and time management, urban and agricultural planning, enterprise resource planning (ERP) and supply chain management (SCM), inventory management, network optimization and, and risk management.

We can summaries the operational research in one world which is optimization Combinatorial optimization is an essential tool combining various techniques of discrete mathematics and computer science in order to solve real life problems. The problems can be combinatorial (discrete) or with continuous variables, with one or more objectives (mono-objective or multi-objective optimization), static or dynamic. It should be noted that the optimization of a multi-objective problem is often more complicated and more difficult compared to the optimization of the mono-objective problems.

Many problems of practical and theoretical importance can be modeled as optimization problems. indeed, we appeal to the concept of optimization in our daily life without necessarily being aware of it, looking for the shortest route in time or distance to reach a given destination, seeking to find the best way to store the business, by seeking to increase our productivity at work, by trying to manage our salary well so that it lasts until the end of the month. The list of situations that require the concept of optimization directly or indirectly is very long: better route, better organization, better management, ... because all resources (time, space, energy, money, ...) are limited, the concept of optimization is naturally required. And we do not forget the most common classic problem in mathematics operations research and optimization, the traveling salesman problem since our work is based on it, which is easy to define but it is difficult to solve.

In a simple way, solving a combinatorial optimization problem for a set of data amounts to finding the optimum of an objective function in order to find an optimal solution in a reasonable execution time among a finite number of choices, often very large under certain constraints.

However, this goal is far from achievable for several problems due to their increasing complexity, on the one hand the great difficulty of optimization problems and on the other hand through many practical applications that can be formulated in the form of a combinatorial optimization.

The complexity theory presented by Gray and his research team allows optimization problems to be classified according to their complexity and provides relevant information used in choosing solution methods.

Due to the importance of these problems, many methods of resolution have been developed in operations research. These methods can be roughly classified into two broad categories: exact methods which ensure completeness of resolution and approached methods which lose completeness in order to gain efficiency.

Several techniques have been used to ensure that problem solving results provide the pertinence the user expects. Among the most famous are those based on metaheuristics, and mainly algorithms inspired by nature. The latter gives us plenty of examples of solving complex problems through various phenomena grouped by the biological field that inspired each, for example: ant colony, bee colony... etc.

Some of these methods alone do not give a satisfactory result, that is why we apply a hybrid method to improve the inspiration biology algorithm, for example (ACO + PSO).

PROBLEMATIC: we want to answer the following question:

Is the proposed model composed of ant colony optimization and general pairwise exchange better than the model composed of ant colony optimization, particle swarm optimization and 3-OPT algorithm?

GOAL: In the context of our work, we will implement the new hybrid model (ACO+GPE) to solve the traveling salesman problem; apply it on real problems, and compare the results with those founded by the model (ACO+PSO+3-OPT).

CONTRIBUTION: This is the proposal of the new (ACO+ GPE) model.

Organization of the thesis:

We start our thesis with a general introduction introducing the field of operational research and optimization, talking about their importance in our life and their complexity for solving them.

The first chapter was based on generality about combinatorial optimization problems and some definitions, then some examples of combinatorial optimization problems and their complexity, at the end we mentioned different existing resolution methods.

The second chapter was about the bio-inspired methods for the resolution of optimization problems. We talk about real ants and social insects in general, intelligence and collective behavior of ants, similarities and differences between artificial ants and real ants, then the two algorithms ACO and GPE. At the end we mentioned the (ACO + PSO + 3-OPT) and (ACO + GPE) models with a discussion, explanation of them and how they work.

In the last chapter we applied the (ACO+GPE) model to real problems, then we compared the results with the results of the existence (ACO+PSO+3-OPT) model.

Finally, we conclude our thesis with a general conclusion ensuring that the existence model (ACO+PSO+3-OPT) is better than the proposing model (ACO+GPE).

CHAPTER I

combinatorial optimizatio

I. 1. Introduction:

Combinatorial optimization is an important path of study, it occupies a very considerable place in operations research and computer science, it is also known as mathematical programing, collection of mathematical principles and methods used for solving quantitative problems in many disciplines, including physics; biology; engineering; economics; and business.

The subject arose from a realization that quantitative problems in distinctly different disciplines have important mathematical elements in common. Because of this commonality; many problems can be formulated and solved by using the unified set of ideas and methods that make up the field of optimization.

I. 2. Combinatorial problem:

A combinatorial problem is a problem where it is a question of finding the best possible combination of solutions. That problem can be either a decision problem, a search problem, or an optimization problem, depending on which question one is supposed to answer. [1]

I.2.1. Decision problem:

A decision problem is a problem where the resolution is limited to answering <yes> or <no> to the question of whether there is a solution to the problem. Therefore, it is not necessary to find the actual solution. [1]

I.2.2. Research problem:

In this precise case, the resolution of the problem no longer stops at the point of knowing if the problem admits or not a solution. The algorithm must be able to provide the solution if it exists. Therefore, any decision problem can be extended to a research problem.[1]

I.2.3. Optimization problem:

An optimization problem is obtained from a research problem by associating a value with each solution. It consists in finding among a set of possible solutions the best one that meets certain criteria described in the form of an objective function to be maximized or minimized, i.e., we will seek a solution of optimal value, minimum if we minimize the objective function, and maximum if we maximize it. [1]

I.2.3.1. Objective function:

This is the name given to the function f (it is also called the cost function or optimization criterion). It is the function that the optimization algorithm will have to "optimize" (Find an optimum).

Mathematically, an optimization problem is written in the following form:

$min\ \ f(\bar{x})$ (Function to optimize) (I.1)

With $\underline{g}(\bar{x}) \leq 0$ (Inequality constraints) (I.2)

And $\underline{h}(\bar{x}) = 0$ (Equality constraints) (I.3)

We have $\underline{x}\ \in R^n, \underline{g}(\underline{x}) \in\ R^m, \underline{h}(\underline{x})\ \in R^p$ [2] (I.4)

I.2.3.2. Decision variables:

They are grouped into the vector x. by changing this vector we are looking for the optimum of the function f. [2]

I.2.3.3. Neighbors:

let $x$be a solution, we say that $x^\star$is a neighboring solution to x, if we can get $x^\star$in slightly modifying x. the neighborhood $v(x^\star)$ of $x$is the set of neighboring solutions of x . [3]

I.2.3.4. Definitions:

$\boldsymbol{x^\star}$ is a local minimum if $\boldsymbol{f(x^\star) \leq f(x)\ \forall\ x\ \in v(x^\star)}$. **(I.5)**

$\boldsymbol{x^\star}$ is a global minimum if $\boldsymbol{f(x^\star) \preccurlyeq f(x)\ \forall\ x\ \in D_f}$ **(I.6)**

$\boldsymbol{x^\star}$ is a local maximum if $\boldsymbol{f(x^\star) \geq f(x)\ \forall\ x\ \in v(x^\star)}$ **(I.7)**

$\boldsymbol{x^\star}$ is a global minimum if $\boldsymbol{f(x^\star) \succ f(x)\ \forall\ x\ \in D_f}$ **(I.8)**

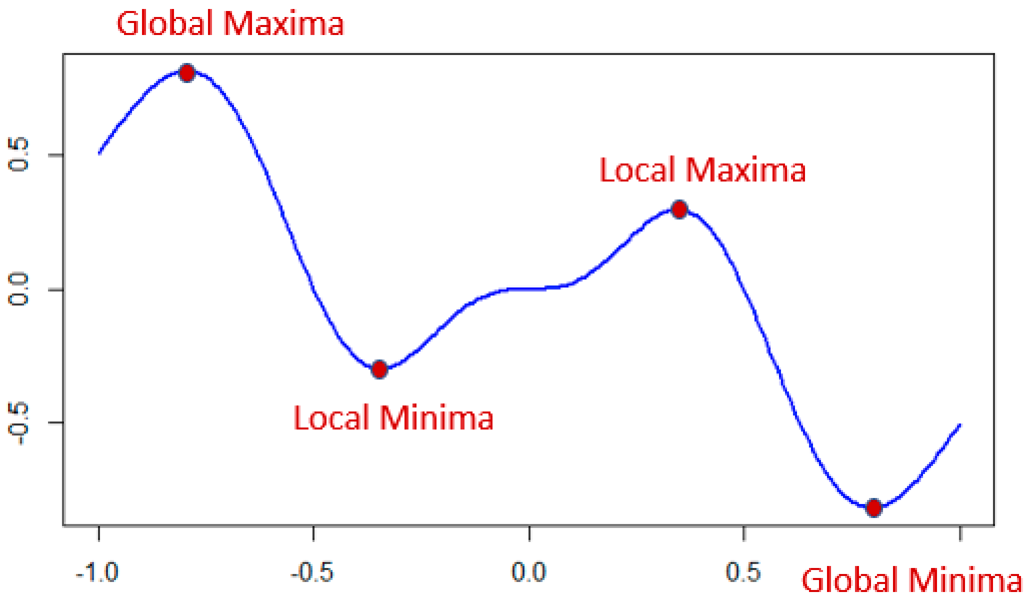


Figure I.1: different minima and maxima [4]

I. 3. Examples of combinatorial optimization problems:

I.3.1. The traveling salesman problem (TSP):

A traveling salesman with n cities to visit wants to set up a tour that allows to pass once and only once in each city to finally return to its starting point, this by minimizing the length of the path traveled. Given a graph G = (X, U) in which the set X of the vertices represent the cities to visit, as well as the city of departure of the tour and U the set of the arcs of G, represent the possible routes between cities. With all arcs $(i, j) \in U$, we associate the travel distance di, j of the city i to city j. the length of a path in G is the sum of distances associated with arcs of this path. The TSP then comes down to the search for a Hamiltonian circuit of minimum length in G.

TSP can be modeled as follows: By associating with each pair (i, j) of cities to be visited (i = 1, ..., n; j = 1…, n and $i \neq j$) a distance $\delta i,j$ equal to di,j if there is a way to go directly from i to j (i.e., (i, j) $\in U$, set to ∞ Otherwise and a succession variable, xi,j, binary, which takes the value 1 if the city j is visited immediately after city i in the route and which takes the value 0 otherwise.

The TSP is then modeled by:

$$Min\sum_{i=1}^{n} \quad \sum_{j=1}^{n} \quad \delta_{i,j}x_{i,j} \qquad (I.9)$$

$$\sum_{j=1}^{n} \quad x_{i,j} = 1 \qquad \forall i = 1, \dots, n \qquad (I.10)$$

$$\sum_{i=1}^{n} \quad x_{i,j} = 1 \qquad \forall j = 1, \dots, n \qquad (I.11)$$

$$\sum_{i \in S, j \notin S} \quad x_{i,j} \geq 2 \qquad \forall\, S \subset X, S \neq \emptyset \qquad (I.12)$$

$$x_{i,j} \in \{0,1\} \qquad \forall i = 1 \dots\ n, \forall j = 1 \dots n \qquad (I.13)$$

The constraint (I.10) - (I.11) translates the fact that each city must be visited exactly once, the constraint (I.12) prohibits solutions composed of disjoint sub-turns, and it is generally called the sub-tower elimination constraint. [3]

I.3.2. Knapsack problem:

Consider n objects, denoted i = 1... n each providing a utility u but having a weight pi. We want to store these objects in a “bag” of capacity c. The knapsack problem consists in choosing the objects to take among the n objects so as to have a maximum utility and respect the capacity constraint, c, not to be exceeded.

The PLNE formulation of the knapsack problem is very simple. We use for each object $i \in 1 \dots n$, a binary variable xi corresponds to 1 if the object i is taken 0 otherwise. The knapsack problem is modeled as follows:

$$Max\sum_{i=1}^{n} \quad u_i x_i \qquad (I.14)$$

$$pixi \leq c \qquad (I.15)$$

$x_i \in \{0,1\}\ \forall\ i = \{1, \dots\ , n\}$. [3] (I.16)

I.3.3. The assignment problem:

By assignment problem we mean the problem of associating each element of a set of N objects with a single element of another set of M objects (with $N \geq M$) with minimal cost. The assignment problem frequently arises in operational research. It consists in performing a bijection of the elements i of a set I on elements of a set J, of the same cardinality, in such a way that a certain cost function, depending on the choice of pairs (i, j) or minimal.

When this cost function is linear, it is a classic problem and its solution is given by a polynomial algorithm).

However, there are problems belonging to many fields, as varied as electronics, economics, IT, etc., for which the cost function is quadratic. The assignment problem is then called "quadratic assignment". It's a NP- Complete problem and therefore much more difficult to solve than linear assignment. [3]

I.3.3.1. Time use problem:

In schools, each year the administration is faced with a problem, it tries to plan a certain schedule while respecting certain constraints such as:

- Each teacher must be assigned to a single room for a certain period of time (Example: one hour) and the same teacher cannot be assigned to two rooms at the same time.
- Each room is occupied by a single teacher (two teachers cannot occupy the same room at the same time).

The problem to be solved consists in reconciling all these constraints in order to offer a job for the time over a certain period (one semester). [3]

I.3.3.2. Assignment problem (Hungarian algorithm):

Given a matrix M (n, n) representing the costs of assigning n individuals to n jobs, it is necessary to assign a different job to each individual for a minimum overall cost.

	T1	T2	T3
P1	6	5	8
P2	15	20	14
P3	3	5	8

For the matrix, the assignment (P1, T2) (P2, T3) (P3, T1) is optimal and for the cost of 22.

I.3.4. The shortest path problem:

Let G be a valued graph and X0 a root of G; determine the minimum values of the paths going from A to F.

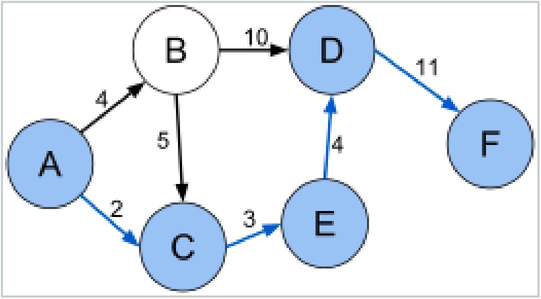


Figure i.2: shortest path from A to F

I.3.5. Graph coloring problem:

This is a problem among the NP-Complete problems due to the difficulty found during its resolution, it consists of assigning a defined number of colors k to the vertices of a non-oriented in a way that the colors of adjacent nodes are different. Coloring minimum uses the smallest possible number of colors (chromatic color). The version decisive point of graph coloring (k-coloring) requests which vertices in the graph can be colored using a number ≤ k colors for a known k. [5]

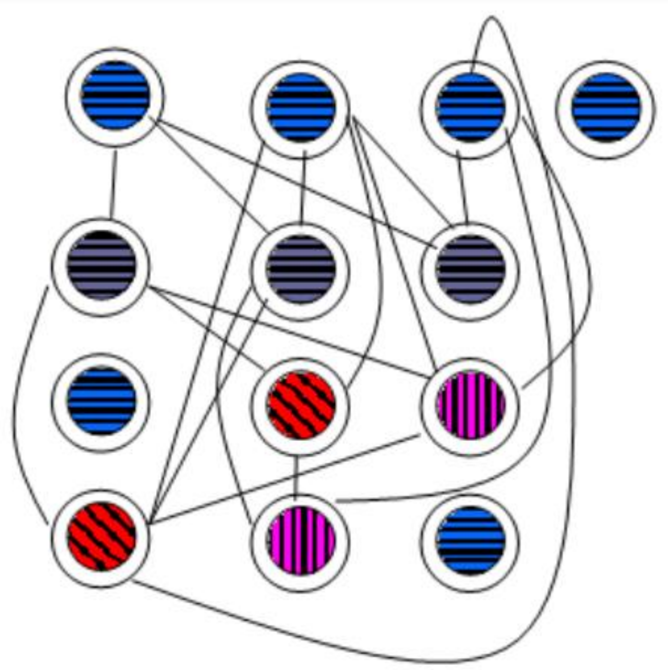


Figure I.3: Example of a colored graph. [5]

I. 4. Complexity of combinatorial problems:

I.4.1. P: Class of decision problems that can be solved in polynomial time.

I.4.2. NP: The NP problems are those whose solutions are hard to find but easy to verify and are solved by a Non-Deterministic Machine in polynomial time.

Note that problems in P have short proofs for both YES and NO answers. This means that P ⊆ NP

I.4.2.1. NP-Hard Problem:

A Problem X is NP-Hard if there is an NP-Complete problem Y, such that Y is reducible to X in polynomial time. NP-Hard problems are as hard as NP-Complete problems. NP-Hard Problem does not have to be in NP class.

I.4.2.2. NP-Complete Problem:

A problem X is NP-Complete if there is an NP problem Y, such that Y is reducible to X in polynomial time. NP-Complete problems are as hard as NP problems. A problem is NP-Complete if it is a part of both NP and NP-Hard Problem. A non-deterministic Turing machine can solve NP-Complete problems in polynomial time. [6]

I. 5. Resolution methods:

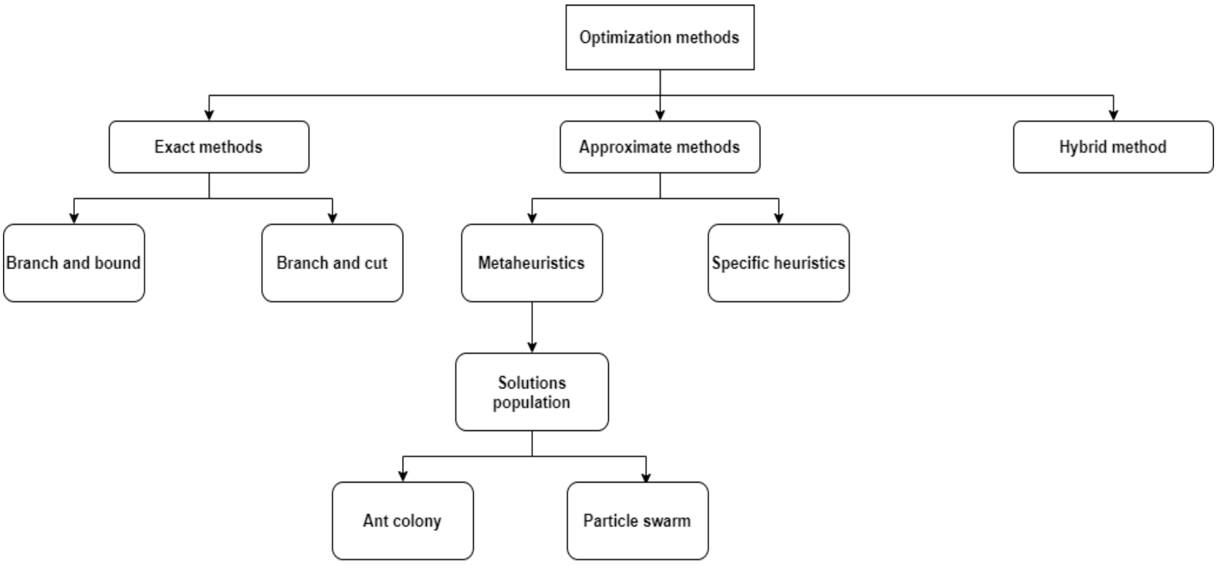


Figure I.4: taxonomy of resolution methods.

I.5.1. The exact methods:

Exact methods seek to find the optimal solution with certainty by explicitly or implicitly examining the entire search space. They have the advantage of guaranteeing the optimal solution, however, the computational time necessary for reaching this solution becomes very excessive depending on the size of the problem (combinatorial explosion) and the number of objectives to optimize. What limits the use of this type of method for small size problems. These generic methods are: Branch & bound, Branch & cut and Branch & price, other methods are less general, such as: linear programming in integers, the algorithm of A *. Other methods are specific to a given problem like Johnson's algorithm for scheduling. [7]

I.5.1.1. The branch & bound algorithm:

(Procedure by evaluation and progressive separation) initially proposed in linear programming in integers, is a large classic to exactly solve optimization problems. It's about a general method, adaptable to many combinatorial problems. It is found for the first time applied to the MP problem. It consists of enumerating the solutions in a clever way in that, using certain properties of the problem in question, this technique manages to eliminate partial solutions that do not lead to the solution that we are looking for. As a result, we often manage to obtain the desired solution in a short time. reasonable. Of course, in the worst case, we always fall back on the explicit elimination of all the solutions to the problem. [7]

I.5.2. Approximate methods:

These methods are used for problems where no algorithms are known resolution in polynomial time and for which one seeks to obtain a "good" solution, without any guarantee that it will be the best. So, they are very useful to be able to approach larger size issues. They bring together heuristics specific to a Particular POC and metaheuristics. The former is not very reusable (the methods constructive, greedy, . . .). On the other hand, metaheuristics are more general and are independent of the processed POCs. In this work, we are exclusively interested in metaheuristic. [7]

I.5.2.1. Heuristics:

For some problems, the algorithms are too complex to obtain a result in a reasonable time, even if one could use a power of phenomenal calculation. We are therefore led to seek a solution as close as possible to an optimal solution by proceeding by successive tests. Since not all combinations can be tried, certain strategic choices must be made. These choices, generally very dependent on the problem treated, constitute what is called a heuristic. The goal of a heuristic is not to try all the possible combinations before finding the one which answers the problem, in order to find a suitable approximate solution (which can be exact in some cases) within a reasonable time. In order to resolve problems and decision-making, heuristics nevertheless find their place in the algorithms that require the exploration of a large number of cases, because these allow us to reduce their average complexity by first examining the cases that are most likely to give the answer. [7]

I.5.2.2. Metaheuristics:

Metaheuristics have grown considerably since their appearance in the 1970s. They are presented by Osman and Laporte (1996) as being approximation methods designed to many complex optimization problems that could not be solved effectively by heuristics and methods of classical optimization. These same authors formally define the notion of metaheuristic as an iterative process that guides a subordinate heuristic in intelligently combining different concepts to explore and exploit the research space, and who uses learning strategies to structure information in order to find efficient solutions as close as possible to the optimal solution. The development of metaheuristics is part of a sustained effort invested in the field of combinatorial optimization.

The most classic metaheuristics are those based on the notion of course. In this perspective, the algorithm changes a single solution on the search space to each iteration.

So, we can classify metaheuristics into two large families: those with a population of solutions and others based on a single solution. [7]

I.5.2.2.1. Ant colonies:

Social insects, such as ants, bees or termites are usually imagined in a simple, non-intelligent way. Nevertheless, they collectively exhibit impressive problem-solving skills.

Ant colony optimization (ACO) is a metaheuristic algorithm which has been proven a successful technique and applied to a number of combinatorial optimization problems and is taken as one of the high-performance computing methods for Traveling salesman problem (TSP). [8] In the next chapter, we will discuss the ACO in detail.

I.5.2.2.2. Particle swarm optimization:

PSO is a stochastic global optimization method which is based on simulation of social behavior [9]. It is a biologically inspired computational search and optimization method developed in 1995 by Eberhart and Kennedy based on the social behaviors of birds flocking or fish schooling. A number of basic variations have been developed due to improved speed of convergence and quality of solution found by the PSO. On the other hand, basic PSO is more appropriate to process static, simple optimization problems. [10]

I.5.3. Hybrid methods:

Another way to improve the performance of an algorithm or to overcome some of its shortcomings is to combine it with another algorithm. This general principle, called hybridization, can be applied for a large number of methods. Evolutionary algorithms are no exception to the rule, and a multitude of hybrid algorithms have appeared in recent years.

I. 6. Study diagram of a combinatorial optimization problem:

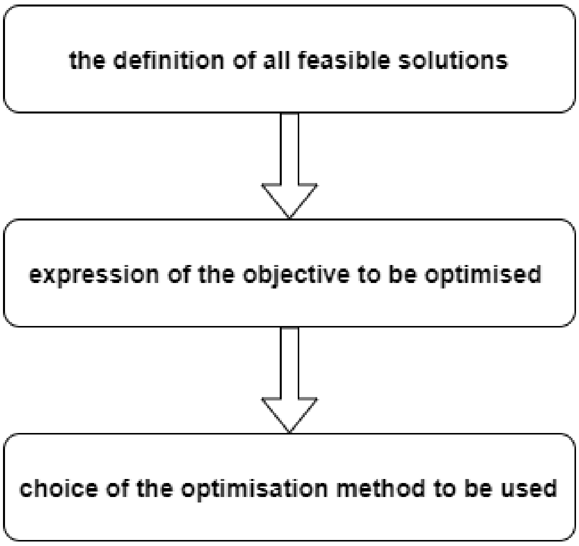

the definition of all feasible solutions
expression of the objective to be optimised
choice of the optimisation method to be used

Figure I.5: Study diagram of a combinatorial optimization problem.

The first two points relate to the modeling of the problem, the third of its resolution. In order to define the set of feasible solutions, it is necessary to express the set constraints of the problem. This can only be done with a good knowledge of the problem under study and its field of application.

I. 7. Conclusion:

In this chapter, we talked about combinatorial optimization problems in general with some definitions, then some examples of combinatorial optimization problems. We talked about the existing methods of resolution, we approached the notions of exact methods and approximate methods with focusing on metaheuristics since our work is based on them.

CHAPTER II

(ACO+GPE) FOR SOLVING THE TRAVELING SALESMAN PROBLEM (TSP)

II. 1. Introduction:

Significant progress has been made allowing the emergence of a new generation of powerful and general approximate methods, often called meta-heuristics, most of which are biologically inspired.

The use of meta-heuristics has captured the attention of the research community for 30 years. The first two decades were marked by the application of standard meta-heuristics. However, in recent years it has become evident that focusing on a single meta-heuristic to solve a complex problem is rather restrictive, a combination with other optimization techniques can provide efficient behavior of the method, thus great flexibility, especially when it comes to large-scale real-world problems.

The meta-heuristic research community agreed that a good meta-heuristic must respond positively and effectively to two criteria, diversity and intensity. Diversity is related to the exploration of the research space, while intensity is related to the ability to experience that space.

A search method is rarely as efficient at exploiting as at exploring the search space. The solution is to associate a method with a very high exploration capacity with a method characterized by a good exploitation of the research space, hence the current emergence of hybrid methods, which strive to take advantage of specific advantages of different approaches by combining them at different levels. Hybrid methods quickly gained traction, successfully producing the best results for many problems.

The hybridization of bio-inspired methods has been very popular and has made it possible to benefit from the strengths of each of these methods and overcome their limitations. Hybridization has made it possible to have a compromise between exploration and exploitation of the solution search space. To have a good exploitation, an algorithm is used to locate the best regions of the search space, another is used to converge towards the global optimum. Hybridization is also used to optimize general parameters. We can conclude that the hybridization of meta-heuristics is the most promising way for improving the quality of solutions in many real applications. Thus, the choice of a hybrid approach is now becoming decisive for obtaining better performance when solving complex problems.

II. 2. Biological inspiration:

Nature is naturally a great and immense source of inspiration for solving difficult and complex problems in optimization [11]. Nature has inspired many researchers in many ways and thus is a rich source of inspiration. Nowadays, most new algorithms are nature-inspired, because they have been developed by drawing inspiration from nature. [12]

Several questions concern biologists: in a colony of social insects, such as ants, bees, termites, etc. Why is the group often considered while each individual seems autonomous? How will the activities of all individuals be coordinated without supervision? The ethological programs that study the behavior of social insects have resulted in the emergence of a new smart comparable paradigm inspired by nature "computing inspired by nature" to deal with complex and dynamic problems of the real world.

The nature-inspired algorithms are meta-heuristics that mimic nature to solve optimization problems. These nature-inspired metaheuristic algorithms can be based on swarm intelligence, biological systems, physical and chemical systems. Therefore, these algorithms can be called swarm-intelligence-based, bio-inspired, physics- and chemistry-based, depending on the sources of inspiration. Though not all of them are efficient, a few algorithms have proved to be very efficient and thus have become popular tools for solving real-world problems. Some of the algorithms have been insufficiently studied. [12]

Thus far, a large number of common NIOAs and their variants have been proposed, such as genetic algorithm (GA), particle swarm optimization (PSO) algorithm, artificial bee colony (ABC) algorithm, ant colony optimization (ACO) algorithm. [13]

For the past decades, much research effort has been focused in this particular area. Still young and the results being very astonishing, expands the scope and viability of bio-inspired algorithms exploring new application areas and more opportunities in computing. Approach inspired bio-inspired is the discipline of studying natural systems in order to find solutions for optimization problems that cannot be solved by conventional methods. Indeed, researchers have noted many similarities between the problems encountered by natural systems and optimization problems. [11]

II. 3. Generality about ants:

The ants can measure from a few millimeters to several centimeters in length. Its body is made up of 3 main parts: the head, the thorax and the abdomen.

- **The Head:** The head is equipped with a pair of antennae, which have the function of taste, touch and smell. Indeed, the antennas can pick up a certain number of odorless, volatile and chemical substances called pheromones which will be used for communication between ants.
- **The thorax:** are covered with hairs representing the sensory organ of the ant. The thorax supports 3 pairs of very long legs ending in two claws. The extremities the thorax, the stomach and the intestine.
- **The abdomen:** The abdomen contains several digestive cavities such as the social crop, the stomach and the intestine. It also comprises the respiratory ducts and the Du fur or "perfume Falcon" gland which secretes the "pheromone". In queens and males the reproductive system is located in the abdomen. [14]

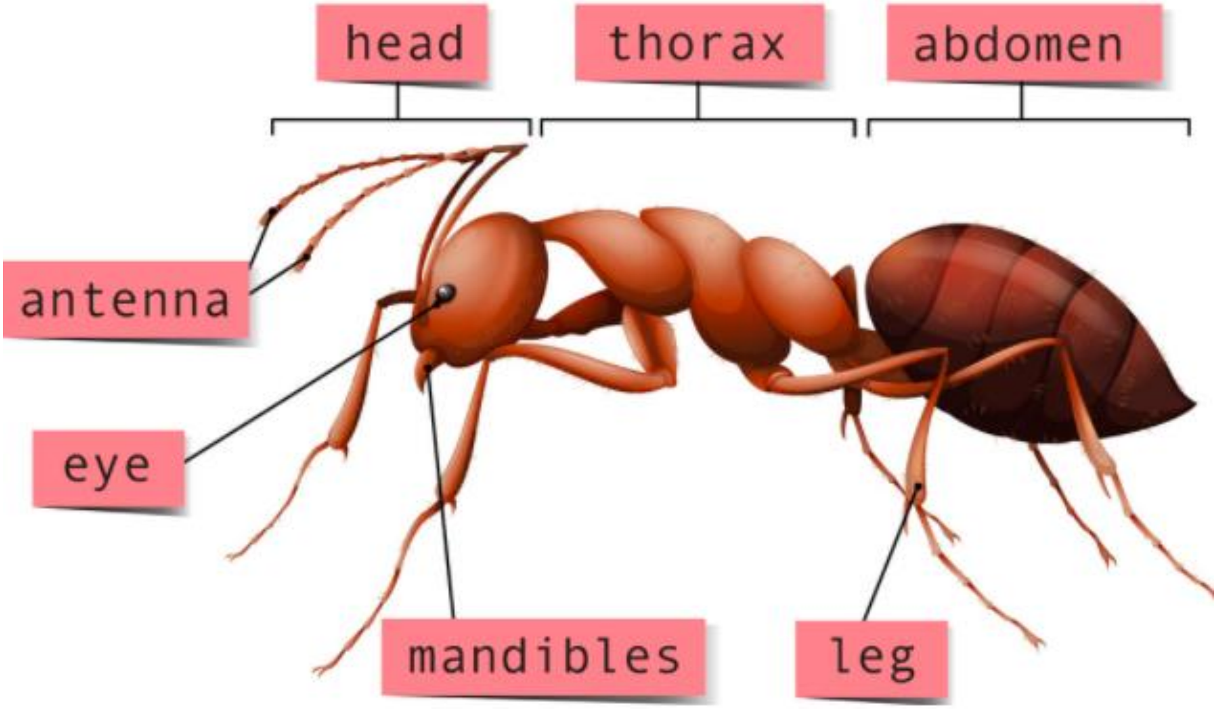


Figure II.1 parts of an ant [15]

II. 4. The real ants:

- The domination of ants is proof of their adaptation to very varied environments.
- The study of ants is done quite easily since they adapt without too much difficulty to environments different from their original habitat.
- Ants have a wide variety of behaviors, collective or individual. [16]

II. 5. The social insects:

Ants are not like other social insects its either from its social behavior or its organization, its development is very advanced and results from the sharing of their production activities "foraging, nest defense, maintenance, nest construction and maintenance of larvae and their food supply «All these activities are important for the development and survival of the colony, these activities have a direct relation with the weather and the environment. This is why the place of ants in the study of animal societies is central.

The number of social species (about 13,500 known) is quite small compared to the number of listed insect species, around 750,000, while social insects represent half of the biomass of insects. The great diversity of ants (approximately 10,000 known species) offers a wide variety of morphologies and behaviors. The study of ants, myrmecology, is therefore a vast and fascinating field of investigation. [2]

I.5.1. Collective intelligence of ants:

Ants, first of all, have something to teach us about how nature works. Any system whose behavior arises from the interactions of its components has something in common with ant colonies. Using ants and other social insects as models, computer scientists have developed software agents that cooperate to solve complex problems, such as the rerouting of traffic in a busy telecom network or internet. Another example, the famous traveling salesman problem, in which a salesman tries to find the shortest and fastest route between many cities, is almost impossible to solve definitively. But with the methods inspired by ants the problem can be solved at least approximately, because ants are very good at finding the shortest path between the food and the nest collectively. Collective robotics borrowed from collective intelligence in ant colonies is being used to manage systems composed of lots of robots in synchronization. [17]

I.5.1.1. The communication:

- **Ants solve the barriers of communication in several ways:**
    - Scent (pheromones)
    - Touch
    - Body language
    - Sound
- **How ants communicate through Scent (pheromones)?**

  The ant antennas are the keys to the mystery of their communication. With the help of an advanced system of pheromones they can "smell" a wide range of topics, ranging from colony activity to territorial conquest. Through millions of years the ants have developed specific pheromone-cocktails to communicate different things to fellow ants. To receive the messages they use their antennas, much the same way we would use our nose if blind and deaf. [18]
- **How do ants communicate through motion and touch?**

  When a worker ant meets a member of the colony, she can tell it things by moving her body in a specific manner, or simply by the touching of antennas. The other ant is then provided with a relatively clear image of what it should look for at the end of the trail. If the first ant has found something edible, she will most likely give the other ant a taste of it from a sample out of her mouth.[18]
- **How do Ants communicate through body language?**

  Just as humans, the ants use body language to communicate things. They can tell the other ants things by lightly touching or stroking the receiver in different ways. This way, they can combine signals of pheromones with that of touch and body language, providing an advanced form of communication. [18]
- **How do ants communicate through sound?**Another peculiar way of how ants communicate is by sound. A majority of ant species use it to communicate, although it is commonly unknown to most people because of its low resonance. The ants can procure different sounds by scraping their legs on a washboard-like part of their body, thus accomplishing different sounds. [18]

I.5.1.2. Division of labor:

An ant colony is like a factory. Nestmates work together to convert resources (food) into products (more ants). This process is made more efficient through division of labor, where different individuals specialize in different jobs. The queen has the very specific role of laying eggs, which she spends most of her life doing. Worker ants perform other duties, often depending on their age. Younger ants work inside the nest, taking care of the queen and her brood. Older workers go outside to gather food and defend the nest against enemies. Despite her size and royal title, the queen doesn't boss the workers around. Instead, workers decide which tasks to perform based on personal preferences, interactions with nestmates, and cues from the environment. [19]

I.5.1.3. Collective Behavior of ants:

Pheromone tracks: By walking from the nest to the food source and vice versa (which at first is essentially done randomly), the ants deposit on the ground a fragrant substance called pheromones. This substance thus makes it possible to create a chemical track, on which the ants find themselves, in fact, other ants can detect the pheromones thanks to sensors on their antennas. Pheromones have a role of path marker: when ants choose their path, they tend to choose the track that carries the highest concentration of pheromones.

This allows them to find their way back to their nest when they return. On the other hand, odors can be used by other ants to find sources of food found by their congeners. This behavior helps to find the shortest path to food when the pheromone tracks are used by the entire colony. In other words, when several marked paths are available to an ant, the latter can know the shortest path to its destination. [20]

I.5.2. Artificial ants:

Artificial ants have a dual nature. On the one hand, they model the abstract behaviors of real ants, and on the other hand, they can be enriched by abilities that real ants do not have, in order to make them more effective than the latter. We are now going to synthesize these similarities and differences. [20]

II. 6. Similarities and Differences between Artificial Ants and Real Ants:

Common point:

a- Colony of cooperating individuals: As with real ants, a virtual colony is a set of unsynchronized entities, which come together to find a "good" solution to the problem at hand. Each group of individuals must be able to find a solution even if it is bad.

b- Pheromone tracks: These entities communicate through the mechanism of pheromone tracks. This form of communication plays a big role in the behavior of ants: its main role is to change the way the environment is perceived by the ants, depending on the history left by these pheromones.

c- Pheromone evaporation: The ACO meta-heuristic also includes the possibility of pheromone evaporation. This mechanism allows them to slowly forget what happened before. This is how they can direct their research in new directions, without being too constrained by their old decisions.

d- Finding the Shortest Path: Real and artificial ants share a common goal: finding the shortest path from a starting point (the nest) to destination sites (food). Local movement Real ants do not jump out of boxes, just like artificial ants, they just move between adjacent sites on the ground.

e- Random choice during transitions. When at one site, real and artificial ants must decide which adjacent site to move to. This decision-making is done at random and depends on the local information posted on the current site. It must take into account the pheromone tracks, but also the starting context (which amounts to taking into account the data of the combinatorial optimization problem for a virtual ant). [20]

Differences:

Criteria	Real Ants	Artificial Ants
Pheromone Depositing Behavior	Pheromone is deposited both ways while ants are moving (i.e., on their forward and return ways).	Pheromone is often deposited only on the return way after a candidate solution is constructed and evaluated.
Pheromone Updating Amount	The pheromone trail on a path is updated, in some ant species, with a pheromone amount that depends on the quantity and quality of the food	Once an ant has constructed a path, the pheromone trail of that path is updated on its return path with an amount that is inversely proportional to the path length stored in its memory.
Memory Capabilitie s	Real ants have no memory capabilities.	Artificial ants store the paths they walked into in their memory to be used in retracing the return path. They also use its length in determining the quantity of pheromone to deposit on their return.
Return Path Mechanism	Real ants use the pheromone deposited on their forward path to retrace their return way when they head back to their nest	Since no pheromone is deposited on the forward path, artificial ants use the stored paths from their memory to retrace their return way.
Pheromone Evaporatio n Behavior	Pheromone evaporates too slowly making it less significant for the convergence	Pheromone evaporates exponentially making it more significant for the convergence
Ecological Constraints	Exist, such as predation or competition with other colonies and the colony's level of protection.	Ecological constraints do not exist in the artificial/virtual world.

Table II.1: Differences between Real Ants and Artificial Ants. [21]

II. 7. The inspiration source of ant colony:

Ant algorithms were inspired by the observation of real ant colonies. Ants are social insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention of many scientists because of the high structuration level their colonies can achieve, especially when compared to the relative simplicity of the colony’s individuals. An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find shortest paths between food sources and their nest. [22]

II. 8. Basic flow of ACO

- Represent the solution space by a construction graph.
- Set ACO parameters and initialize pheromone trails
- Generate ant solutions from each ant‘s walk on the construction graph mediated by pheromone trails.
- Update pheromone intensities.
- Go to step 3, and repeat until convergence or termination conditions are met.

As shown in the basic flow of ACO above, the objective of ACO‘s third step is to construct ant solutions (i.e., find the quality paths on the problem‘s construction graph) by stochastically moving through neighbor nodes of the graph.

Ants are driven by a probability rule to sequentially choose the solution components that make use of pheromone trail intensities and heuristic information.

The solution of each ant is constructed when all solution components are selected by that ant (i.e., when the ant has completed a full tour/path on the construction graph).

Once an ant has constructed a solution, or while the solution is being constructed, the ant evaluates the full (or partial) solution to be used by the ACO ‘s next step (the pheromone updating step) in determining how much pheromone to deposit.

The probability rule (equation 1) is called Random Proportional Action Choice rule (or State Transition rule). It guides ant movement through a stochastic local decision policy that essentially depends on both pheromone information and heuristic information.

$$p_{ij}(t) = \{\frac{(\tau_{ij}(t))^{\alpha}*(n_{ij})^{\beta}}{\sum_{i\in j_j^k} \quad (\tau_{ij}(t))^{\alpha}*(n_{ij})^{\beta}}, \quad si \;\; j\epsilon\, j_i^k \qquad \text{(II.1)}$$

$$0 \qquad\qquad si \;\; j \notin j_i^k$$

Where:

$P_{ij}^{K}(t)$ Is the probability of the k^{th} ant to move from node i to node j at the t^{th} iteration/time step.

n_i^k Is the set of nodes in the neighborhood of the k^{th} ant in the i^{th} node.

$p_{ij}^{k}(t) = 0, \forall\, j \notin n_i^k$ Means the ants are not allowed to move to any node not in their neighborhood.

$[\tau\_ij\ (t)]^{\alpha}$ Is the pheromone amount on the arc connecting node i and node j, weighted by (an application-dependent constant). $\tau(t)$ Is the pheromone information, or trail intensity value, that encodes a long-term memory about the whole ant search process. It is updated by all ants after each iteration t (sometimes, however, in more recent ACO versions it is updated by only some ants – the best one(s) that constructed the iteration-best or best-so-far solution).

$[n_{ij}]^{\beta}$ Is the heuristic value of the arc connecting node i and node j, weighted by β (an application-dependent constant). n Is the heuristic information, or path visibility, that represents a priori information about the problem instance definition, or run-time information provided by a different source other than ants. The heuristic value n_{ij} is usually a non-increasing function in the moving cost from node i to node j, and it often does not change during algorithm execution unless the moving cost is not static.

α And β are weight parameters that control the relative importance of the pheromone versus heuristic information.

- A high value for α means that pheromone information is very important; thus, ants are strongly biased to choose nodes previously chosen bY other ants. This potentially leads to a stagnation situation in which all the ants would eventually follow the same path (usually suboptimal) and construct the same tour.

- A low value of α makes the algorithm very similar to a stochastic multi greedy algorithm with m starting points, as there is m number of ants that are initially randomly distributed over the construction graph.
- When α = 0, the ACO performs a typical stochastic greedy search strategy in which the next node (problem state) is selected only on the basis of its distance (cost) from the current node/state. As a result, the node with the minimum cost will be always favored regardless of how many other ants have visited it, and how much its pheromone intensity is.
- When β = 0, the pheromone information is only used to guide the search process, which would reflect the way that ants do in the real world (real ants do not use any heuristic information in their search process).

The objective of ACO‘s fourth step is to update pheromone trails. At the very beginning, the pheromone trails of all arcs on the construction graph are initialized to a small constant value (τ_0). Then after a tour (or, a solution path) is constructed, the pheromone trails are updated in two ways, as shown in equations 2 and 3.

Firstly, the pheromone trails of all arcs are decreased according to an evaporation rate (ρ) that allows ants to forget the suboptimal paths to which they previously converged. Pheromone evaporation rate is usually set to be sufficiently fast in order to favor the exploration of new areas of the search space, and avoid a premature convergence of the algorithm toward a local optimum. Secondly, the pheromone trail values of the visited arcs are increased with amounts inversely proportional to the cost of their tours (or, in other words, directly proportional to their tour quality).

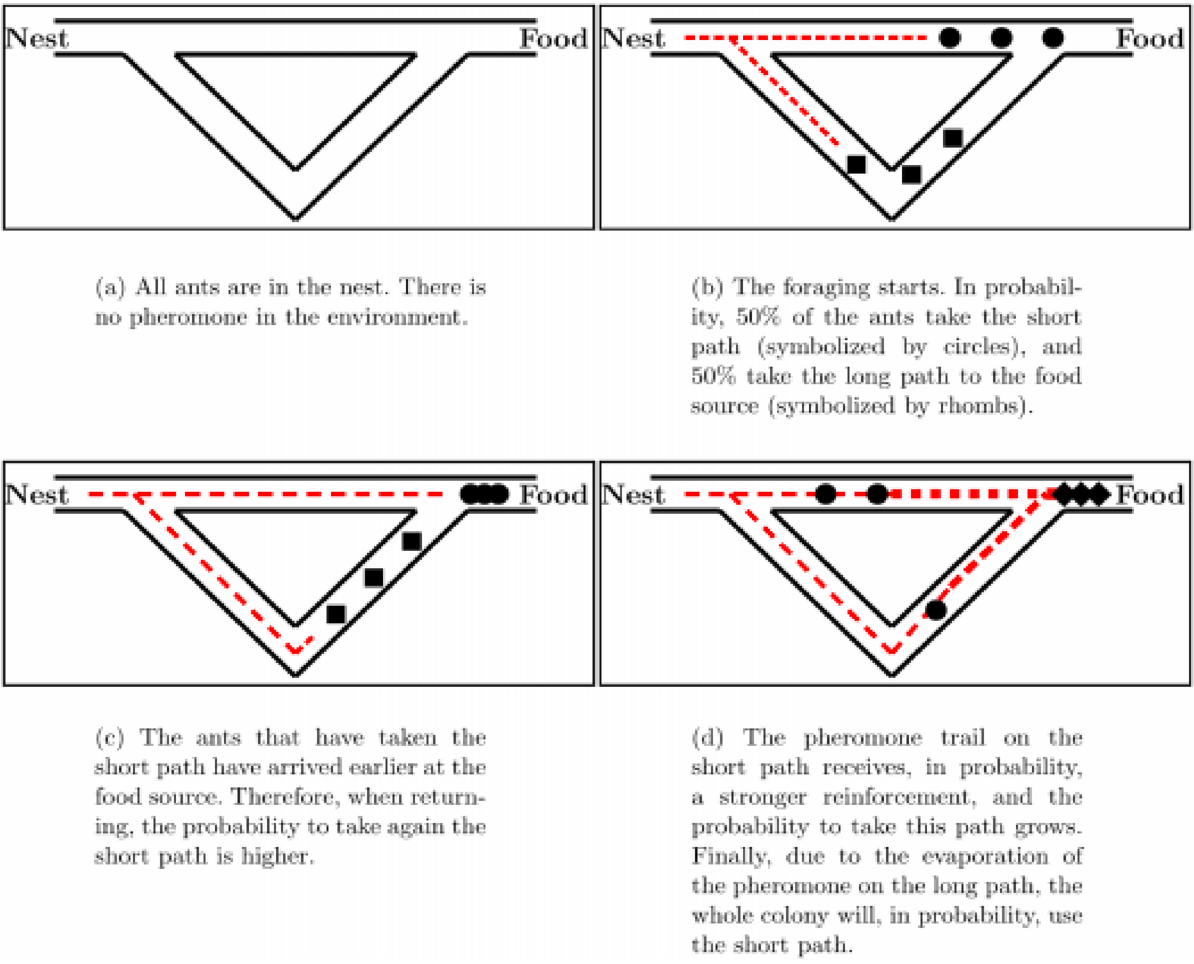
The pheromone depositing procedure implements a useful form of exploitation of quality paths by increasing their probability of being used again by future ants. The quality paths would include the solution components that were either used by many ants in the past, or that were used by at least one ant and which produced a high quality solution.

$$\tau_{ij}(t+1) \leftarrow (1-p) * \tau_{ij}(t) + \sum_{k=1}^{m} \quad \Delta\tau_{ij}^{k}(t), \qquad \forall\, i,j \ \in A,\ 0 \le p \le 1 \qquad (II.2)$$

$$\Delta\tau_{ij}^{k}(t) = \{\frac{Q}{C^{k}(t)} \ , if \ \ arc(i,j)\,\epsilon\, T^{k}(t) \qquad \text{(II.3)}$$

$$0 \ \ , \quad otherwise$$

Where: Q is an application-specific constant, m is the number of ants, A represents all arcs of the problem‘s construction graph, $C^k(t)$ is the overall cost function of tour $T^k(t)$ constructed by the K^{th} ant at the t^{th} iteration, and $T^k(t)$ is the set of all arcs visited by ant k at the iteration t. Other variations of ACO, however, restrict pheromone depositing to the arcs of the best tour T^{best} only. [21]



(a) All ants are in the nest. There is no pheromone in the environment.

(b) The foraging starts. In probability, 50% of the ants take the short path (symbolized by circles), and 50% take the long path to the food source (symbolized by rhombs).

(c) The ants that have taken the short path have arrived earlier at the food source. Therefore, when returning, the probability to take again the short path is higher.

(d) The pheromone trail on the short path receives, in probability, a stronger reinforcement, and the probability to take this path grows. Finally, due to the evaporation of the pheromone on the long path, the whole colony will, in probability, use the short path.

Figure 2.2 An experimental setting that demonstrates the shortest path finding capability of ant colonies. Between the ants’ nest and the only food source exist two paths of different lengths. In the four graphics, the pheromone trails are shown as dashed lines whose thickness indicates the trails’ strength. [23]

II. 9. ACO ALGORITHM FOR THE TSP:

In ACO algorithms ants are simple agents which, in the TSP case, construct tours by moving from city to city on the problem graph. The ants’ solution construction is guided by (artificial) pheromone trails and an a priori available heuristic information.

When applying ACO algorithm to the TSP, a pheromone strength τij (t) is associated to each arc (i, j), where τij (t) is a numerical information which is modified during the run of the algorithm and t is the iteration counter. If an ACO algorithm is applied to symmetric TSP instances, we always have τij (t) = τji(t); in applications to asymmetric TSPs (ATSPs), we will possibly have τij (t) ≠ τji(t).

Initially, each of the m ants is placed on a randomly chosen city and then iteratively applies at each city a state transition rule. An ant constructs a tour as follows.

At a city i, the ant chooses a still unvisited city j probabilistically, biased by the pheromone trail strength τij (t) on the arc between city i and city j and a locally available heuristic information, which is a function of the arc length. Ants probabilistically prefer cities which are close and are connected by arcs with a high pheromone trail strength.

To construct a feasible solution each ant has a limited form of memory, called tabu list, in which the current partial tour is stored. The memory is used to determine at each construction step the set of cities which still has to be visited and to guarantee that a feasible solution is built. Additionally, it allows the ant to retrace its tour, once it is completed.

After all ants have constructed a tour, the pheromones are updated. This is typically done by first lowering the pheromone trail strengths by a constant factor and then the ants are allowed to deposit pheromone on the arcs they have visited. The trail update is done in such a form that arcs contained in shorter tours and/or visited by many ants receive a higher amount of pheromone and are therefore chosen with a higher probability in the following iterations of the algorithm. In this sense the amount of pheromone τij (t) represents the learned desirability of choosing next city j when an ant is at city i. [24]

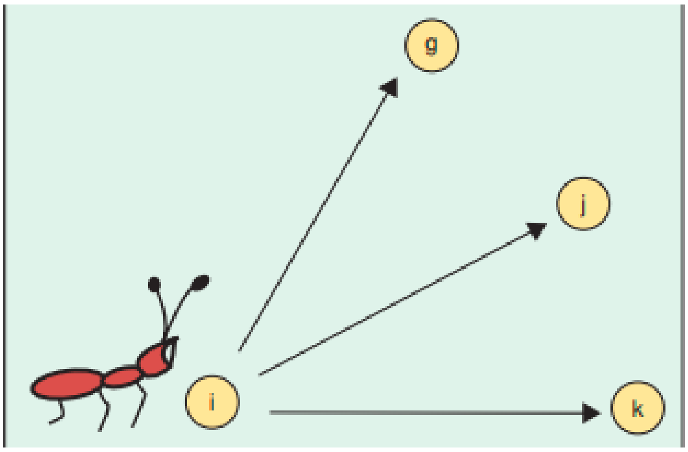


FIGURE II.3 An ant in the city i chooses the next city to visit [25]

II. 10.Improvement approaches:

Unlike constructive methods whose objective is to build a solution, improvement methods modify an initial solution, in order to improve its value. This initial solution can be either chosen at random or obtained using a pre-processing via a constructive method. At each step of the process, the current solution is transformed into another solution using elementary modifications. In the case of the traveling salesman, an elementary modification can be represented by the permutation of two cities in his round. The process stops when the predefined termination criteria is met. It can be a number of modifications (or steps), the exceeding of a threshold fixed by the cost function, or even the execution time.

II. 11.General pairwise exchange (GPE):

General pairwise exchange is an improvement algorithm, it is based on node exchanges by exchanging only two nodes in each tour (round). The idea consists of reducing the size of the problem in each step.

To illustrate this program, just imagine a traveling salesman problem that must visit a set of n nodes, it can improve its tour in the following way:

From a feasible solution, we exchange each node with all other nodes, provided that by changing two nodes on each tour choose the best solutions we keep approaching the solutions until there is no more approach.

- Forecast window=1

1	2	3	4	5

- Forecast window=2

1-2	2-3	3-4	4-5	5

- Forecast window=3

1-3	2-4	3-5	4-5	5

- Forecast window=4

1-4	2-5	3-5	4-5	5

- Forecast window=5

1-5	2-5	3-5	4-5	5

BEST SOLUTION 4←1←2←3←4 [28]

EXAMPLE:

0	**9**	**17**	**13**	**7**
9	**0**	**8**	**19**	**8**
17	**19**	**0**	**7**	**14**
13	**8**	**7**	**0**	**9**
7	**3**	**14**	**9**	**0**

2-5-1-4-3-2

3-5-1-4-2-3	L=46
5-2-1-4-3-5	L=61
1-5-2-4-3-1	L=42
4-5-1-2-3-4	L=51
2-1-5-4-3-2	L=51
2-4-1-5-3-2	L=61
2-3-1-4-5-2	L=61
2-5-4-1-3-2	L=61
2-5-3-4-1-2	L=46
2-5-1-3-4-2	L=42

Best solutions are: 1-5-2-4-3-1 AND 2-5-1-3-4-2

II. 12.The (ACO+PSO+3_OPT) model:

The Traveling Salesman Problem (TSP) is one of the standard test problems used in performance analysis of discrete optimization algorithms. The Ant Colony Optimization (ACO) algorithm appears among heuristic algorithms used for solving discrete optimization problems.

A new hybrid method is proposed to optimize parameters that affect performance of the ACO algorithm using Particle Swarm Optimization (PSO). In addition, a 3-Opt heuristic method is added to the proposed method in order to improve local solutions. The PSO algorithm is used for detecting optimum values of parameters α and β which are used for city selection operations in the ACO algorithm and determines significance of inter-city pheromone and distances. The 3-Opt algorithm is used for the purpose of improving city selection operations, which could not be improved due to falling in local minimums by the ACO algorithm.

The performance of this hybrid method is investigated on ten different benchmark problems taken from literature and it is compared to the performance of some well-known algorithms. Experimental results show that the performance of the proposed method by using fewer ants than the number of cities for the TSPs is better than the performance of compared methods in most cases in terms of solution quality and robustness. [26]

II. 13.The (ACO+GPE) model:

Our hybrid approach is based on the incorporation of the GPE improvement algorithm by combining it with ACO algorithm, which improves the initial phase of a simple metaheuristic construction (ACO) in which we choose the value of parameters α and β between [0 2] to get the best solution for the ACO algorithm, GPE is applied to only these good solutions. This will allow us to improve convergence by achieving a compromise between the exploration and the exploitation of the search space.

II. 14.CONCLUSION:

In this chapter, we touched on biological inspiration with a focus on ACO and GPE, then we mentioned the (ACO + PSO + 3-OPT) and (ACO + GPE) models with a discussion, explanation of them and how they work.

CHAPTER III

IMPLEMENTATION AND RESULTS

III. 1. Introduction:

Our application was created for a contribution which consists in proposing a hybrid approach based on a bio-inspired method and an improvement algorithm (ACO + GPE), to solve the traveling salesman problem (TSP) and compare the results with another approach (ACO + PSO + 3-OPT).

III. 2. MATLAB:

In our work we use the MATLAB language because it has several advantages over other methods or languages, MATLAB allows us to:

- Implement and test our algorithms easily
- Develop the computational codes easily
- Debug easily
- Perform extensive data analysis and visualization
- Develop application with graphics user interface

III. 3. GPE implementation:

```
function [tour, Cost, Best] = gpe (tour, D, Cost)
n = numel (tour);
L = 0;
Best = zeros ([], 1);
Best (1, 1) = Cost;
APE.p = tour;
while L &lt; = Cost
    for i = 1: n - 1
        for j = i + 1: n
            p = tour;
            b = tour (i);
            v = tour (j);
            c = b;
            b = v;
            v = c;
            p(i) = b;
            p(j) = v;
            l = TourLengt (p, D);
            L = l;
            if L &lt; Cost
                Cost = L;
                APE.p = p;
                Best (end + 1, 1) = Cost;
            end
            BC (i, j) = L;
```

```
                end
            end
            Cost;
            tour = APE.p;
            L = max (min (BC));
end
tour = [tour tour (1)];
Best;
Cost;
```

description of the parameters

D : distance matrix

Cost : tour value

n: number of cities

L: Cost element

b, c, v: permutation parameters

III. 4. The first modification of the Hybrid Method (ACO+PSO+3-OPT):

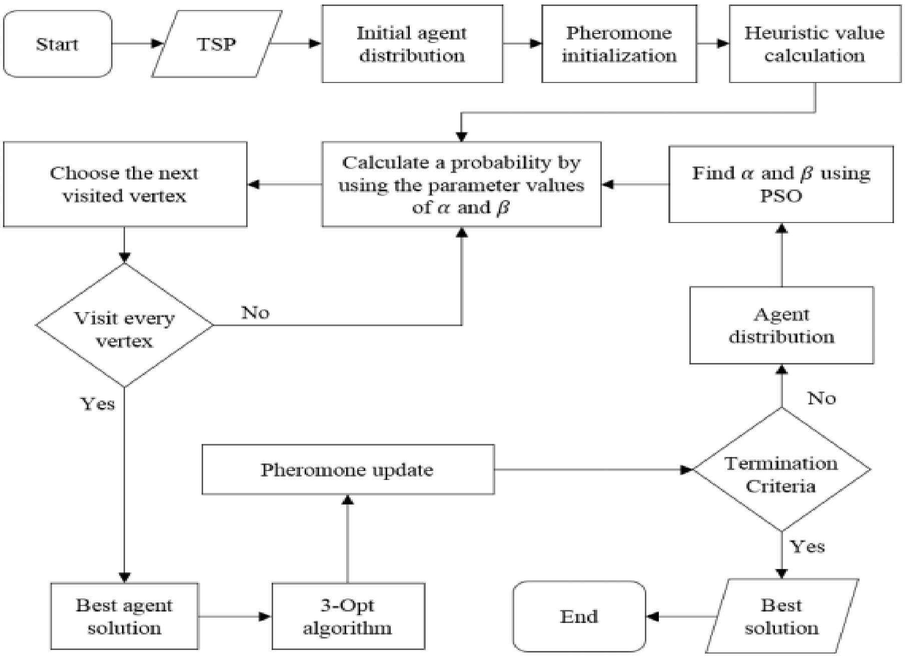

Start
TSP
Initial agent distribution
Pheromone initialization
Heuristic value calculation
Choose the next visited vertex
Calculate a probability by using the parameter values of α and β
Find α and β using PSO
Visit every vertex
No
Yes
Agent distribution
No
Pheromone update
Termination Criteria
Yes
Best agent solution
3-Opt algorithm
End
Best solution

FIGURE III.1: THE First Modification of Hybrid Method (ACO+PSO+3-OPT) [27]

DISCUSSION:

In their first modification method, 3- opt algorithm is applied to the feasible solution, as described in Figure III.1. The feasible solution will be chosen from all solutions generated by all agents. The 3-Opt algorithm will be applied to these feasible solutions. The next step is the third ACO process that updates the pheromone and follows the other steps until the process satisfies the terminal criteria. When the criteria are fulfilled, the best feasible solution will be the best solution of the first modification hybrid method.

III.4.1. The results of the first model (ACO+PSO+3-OPT):

The implementation uses 6 benchmarks problems from TSPLIB (Reinelt, 1991): Eil51, Berlin52, St70, Eil76, Rat99, and kroA200.

Every problem is tested in 10 runs with 50 iterations each.

Mean Solution (MS) is calculated by using the following equation:

$$MS = \frac{\sum_{i=1}^{n} \quad D_i}{n} \qquad (III.1)$$

Where Di is the solution of problem i, i = 1, 2… n. The percentage of the relative error (ER) is used to Determine how good the method solving the TSP. It calculated by the equation:

$$ER = \frac{MS-BKS}{BKS} * 100 \qquad (III.2)$$

Where BKS is the best-known solution. [27]

PROBLEMS	BKS SOLUTION	BEST SOLUTION	WORST SOLUTION	MEAN SOLUTION	ER (%)	TIME (SECOND)
EIL51	426	437	514	480.8	12.86	325.55
BERLIN52	7542	7930	8832	8351.6	10.73	328.27
ST70	675	683	777	732.8	8 .56	427.28
EIL76	538	605	663	634.4	17.92	488.64
RAT99	1211	1311	1499	1439.6	18.88	818.51
KROA200	29368	29957	36761	34039.4	15.91	1042.13

TABLE III.1. The results of the first model (ACO+PSO+3-OPT). [27]

III. 5. THE First Modification of the Hybrid Method (ACO+GPE):

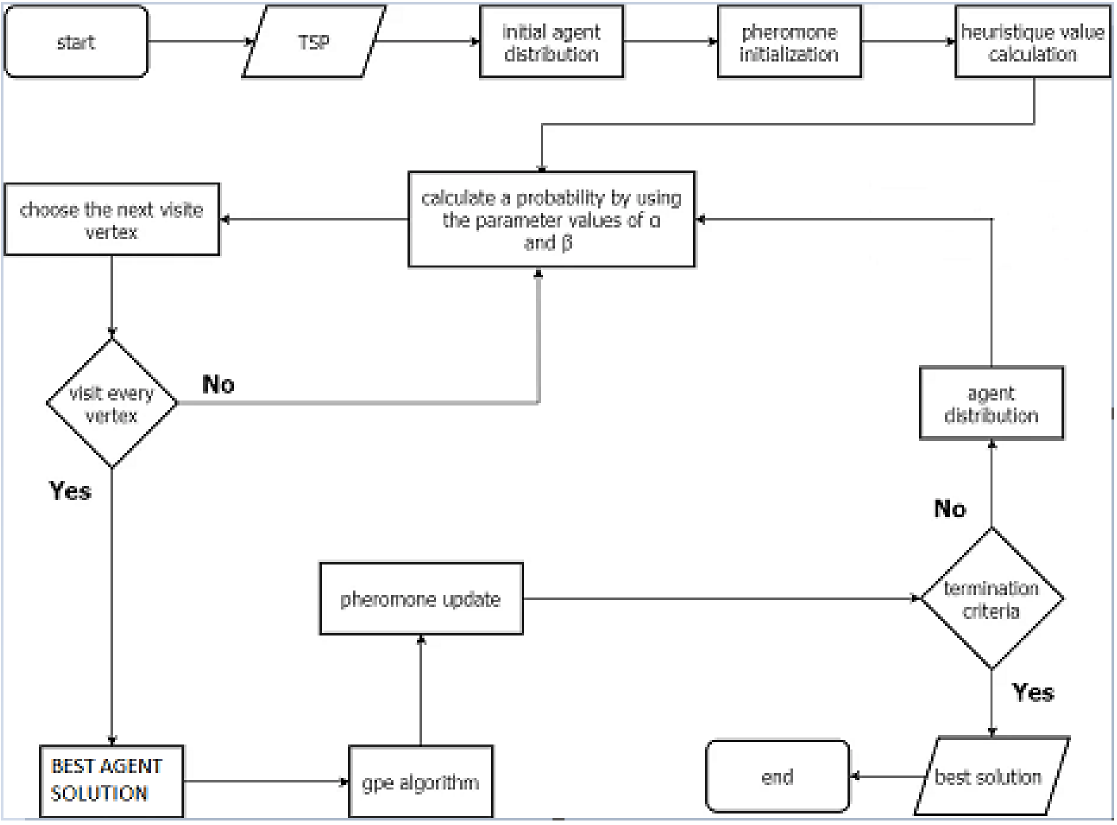


FIGURE III.2: The First Modification of Hybrid Method (ACO+GPE)

DISCUSSION:

In the first modification method, the GPE algorithm is applied to the feasible solution, as described in Figure III.2. The feasible solution will be chosen from all solutions generated by all agents. The GPE algorithm will be applied to these feasible solutions. The next step is the third ACO process that updates the pheromone and followed by the same steps as theirs until the process satisfies the terminal criteria. When the criteria are fulfilled, the best feasible solution will be the best solution of the first modification hybrid method.

III.5.1. The results of the second model (ACO+GPE):

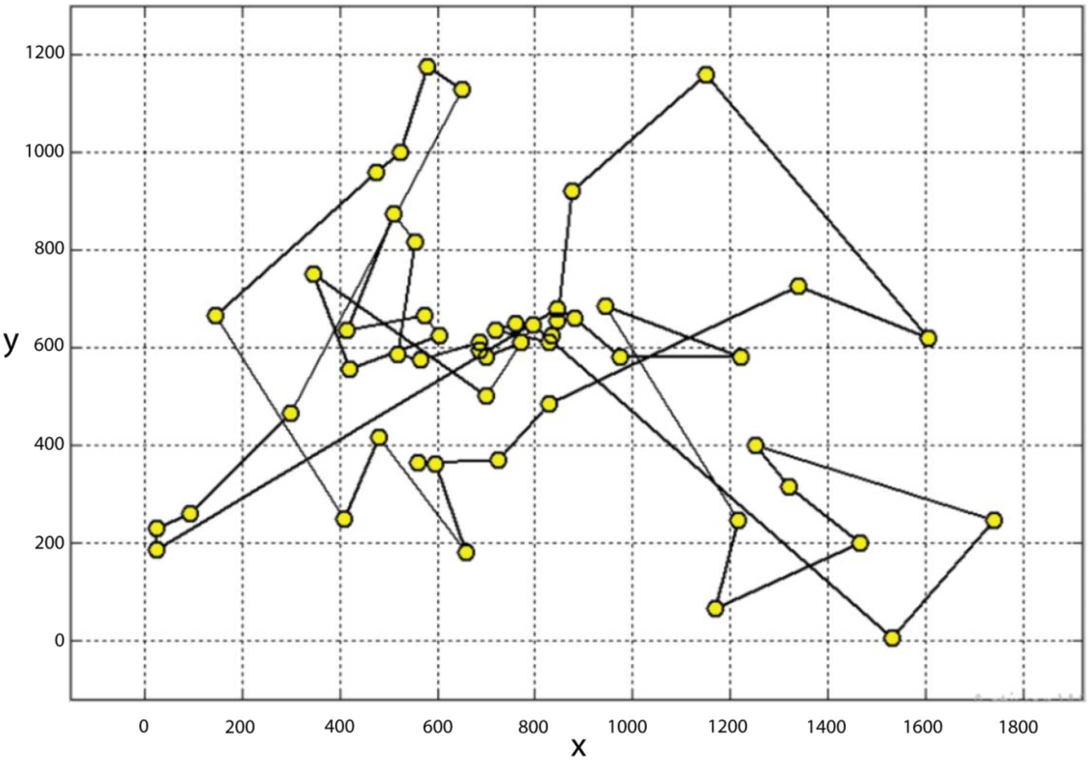


FIGURE III.3: Best solution of Berlin52 after 10 runs

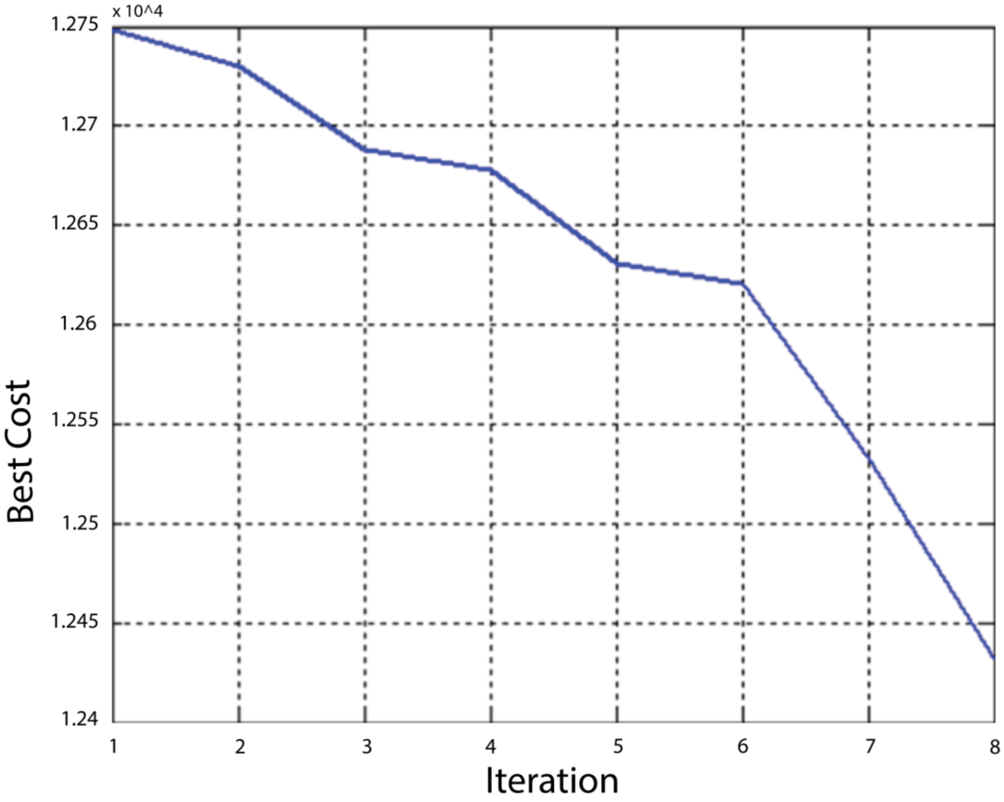


FIGURE III.4: Best cost of Berlin52 after 10 runs

PROBLEMS	BKS SOLUTION	BEST SOLUTION	WORST SOLUTION	MEAN SOLUTION	ER (%)	TIME (SECOND)
EIL51	426	691.75	763.55	725.87	70.39	28.18
BERLIN52	7542	11859.40	12464.83	12044.20	59.69	28.43
ST70	675	1188.26	1339.82	1257.65	86.31	26.76
EIL76	538	964.70	1111.56	1053.7	95.85	32.19
RAT99	1211	2534.78	2806.25	2701.95	123.11	33.977
KROA200	29368	76517.86	86742.22	83071.96	182.86	38.004

TABLE III.2. The results of the second model (ACO+GPE).

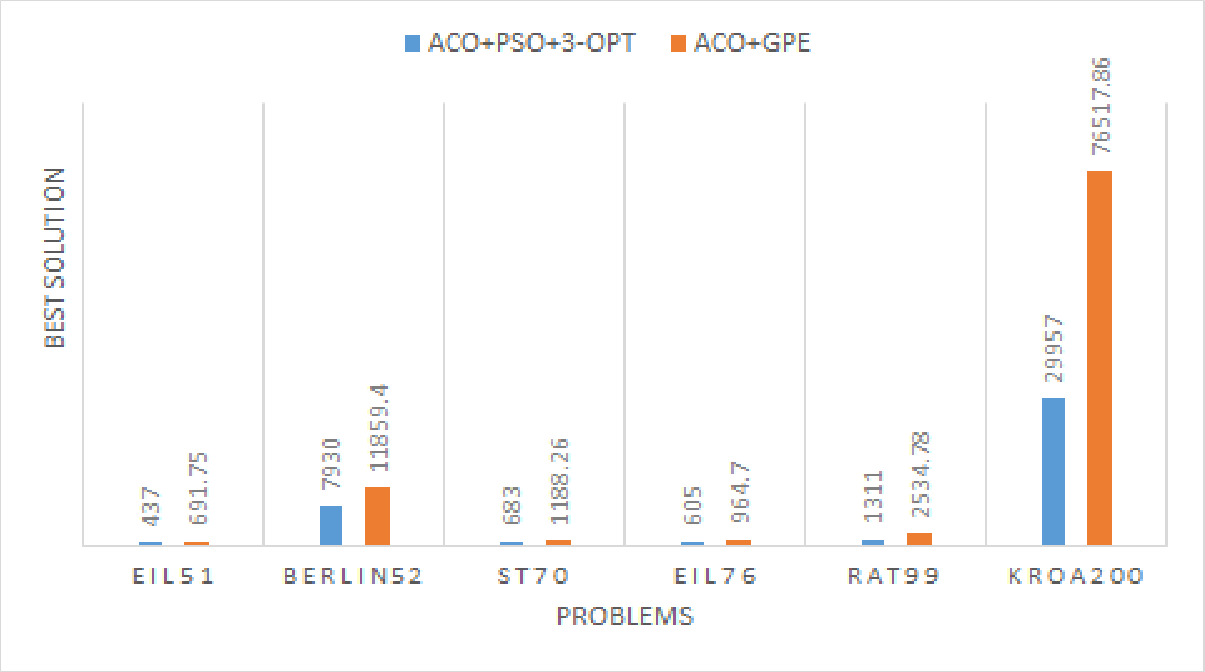


FIGURE III.5 The differences between the best solutions of the two methods

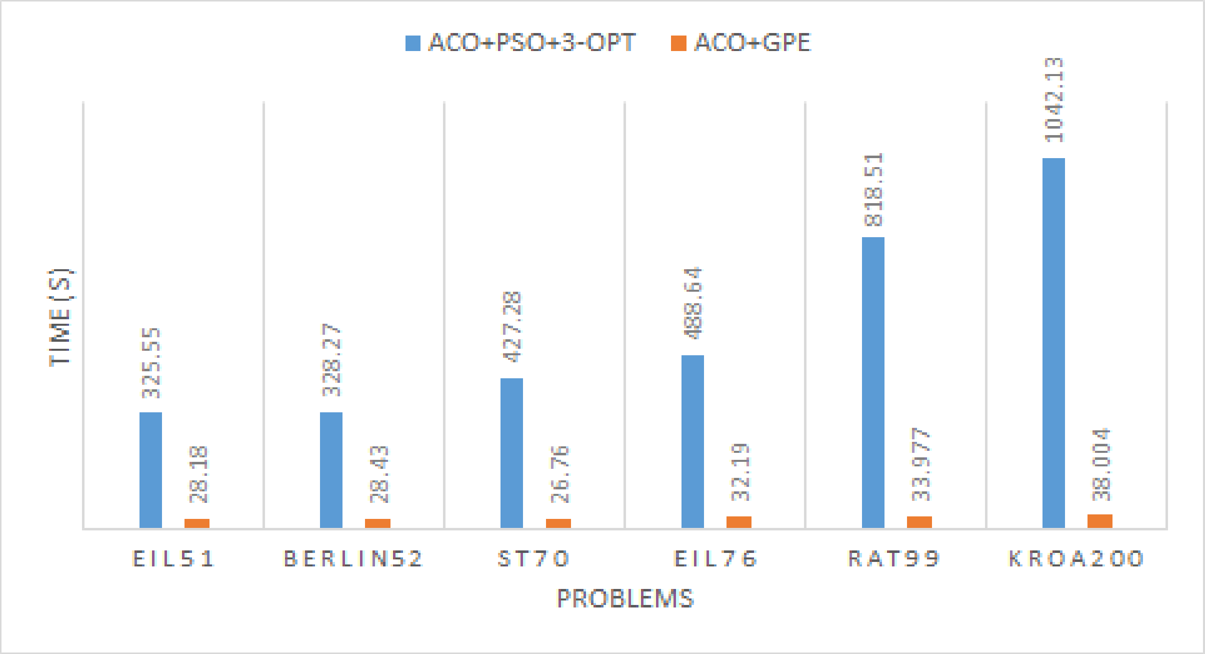


FIGURE III.6 The differences between the times of the two methods

III. 6. Discussion about the first hybrid modification method:

Table III.2 shows that the first hybrid modification method does not perform better for all problems. It shows the corresponding best solution is larger than the best-known solution (BKS). Furthermore, the corresponding ER is also too high especially (RAT99 AND KROA200). But the results show less running time.

The first hybrid modification shows that the (ACO+PSO+3-opt) gives better results of best solutions than (ACO+GPE) nevertheless, if we consider the CPU time (ACO+GPE) gives better results.

III. 7. The second modification of hybrid method (ACO+PSO+3-OPT):

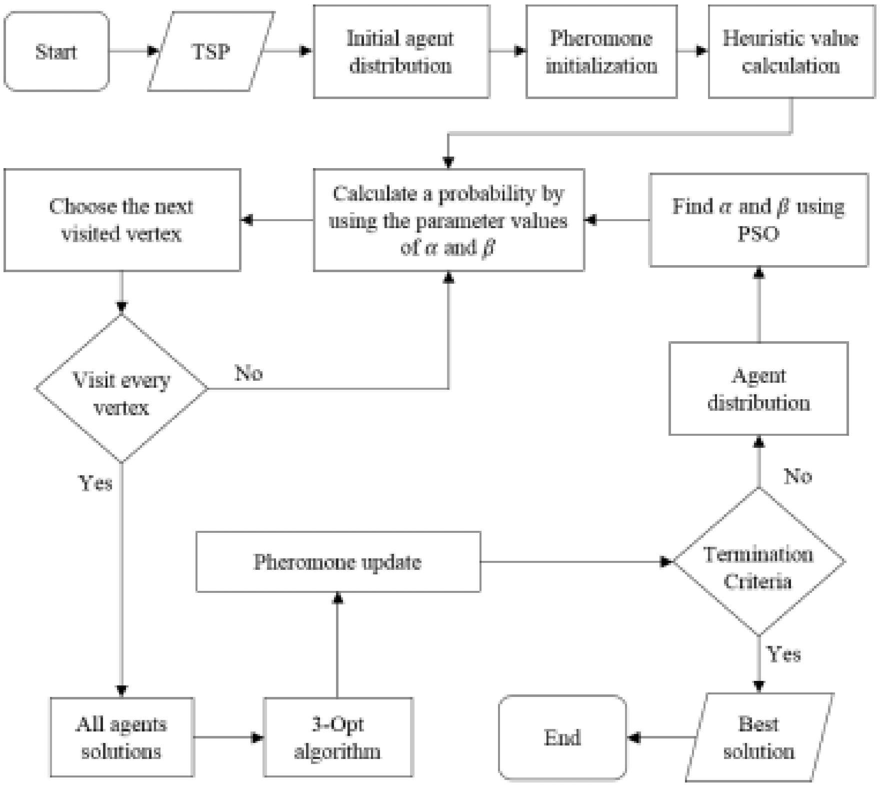

Start
TSP
Initial agent distribution
Pheromone initialization
Heuristic value calculation
Choose the next visited vertex
Calculate a probability by using the parameter values of α and β
Find α and β using PSO
Visit every vertex
No
Agent distribution
No
Yes
Pheromone update
Termination Criteria
Yes
All agents solutions
3-Opt algorithm
End
Best solution

FIGURE III.7: THE SECOND MODIFICATION OF HYBRID METHOD(ACO+PSO+3-OPT) [27]

DISCUSSION:

In the second hybrid modification method, we implement the 3-Opt algorithm to all agent’s solutions in each iteration as shown in Figure III.5. In the first modification method, the 3-Opt algorithm is only applied to the feasible solution in each ACO iteration, while in the second modification method, the 3-Opt algorithm is applied to all agent’s solutions in each iteration. After all solutions have been optimized by 3-Opt, the solution with the minimum total distance is selected as a feasible solution.

III.7.1. The results of the first model (ACO+PSO+3-OPT):

With the same rules, we calculate the results in the previous table

PROBLEMS	BKS SOLUTION	BEST SOLUTION	WORST SOLUTION	MEAN SOLUTION	ER (%)	TIME (SECOND)
EIL51	426	426	426	426	0.00	325.55
BERLIN52	7542	7542	7542	7542	0.00	328.27
ST70	675	675	675	675	0.00	427.28
EIL76	538	538	538	538	0.00	488.64
RAT99	1211	1211	1211	1211	0.00	818.51
KROA200	29368	29368	29368	29368	0.00	1042.13

TABLE III.3. The results of the first model (ACO+PSO+3-OPT). [27]

III. 8. The second Modification of Hybrid Method (ACO+GPE):

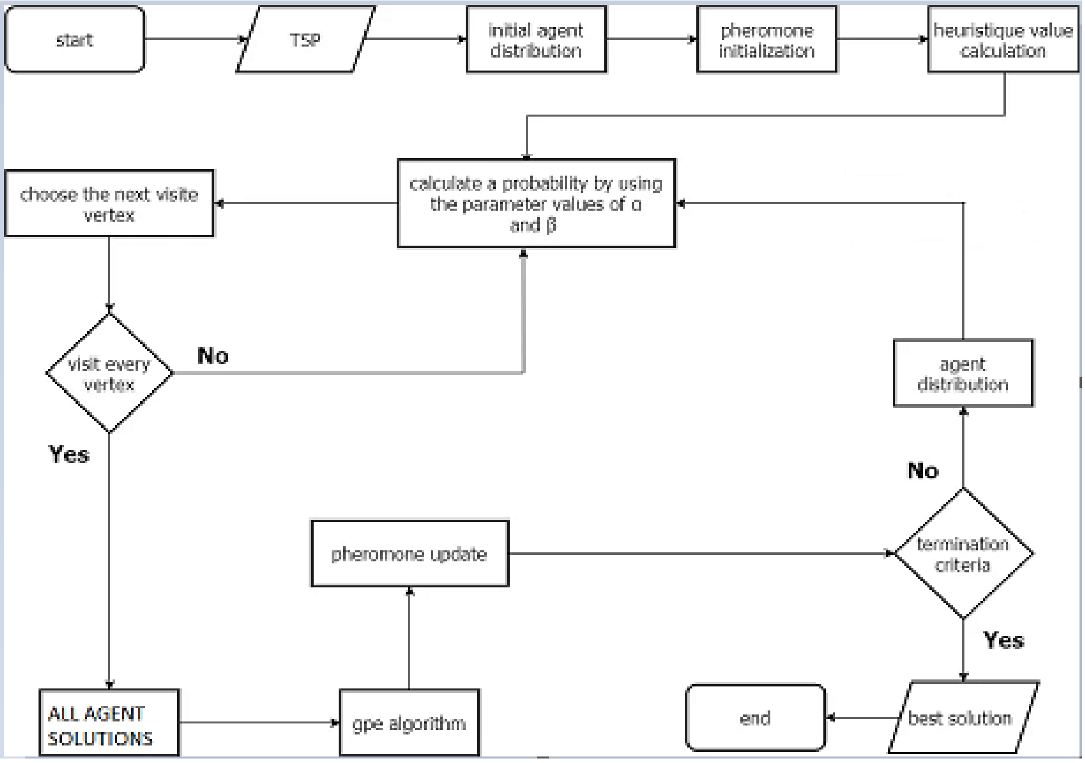


FIGURE III.8 The second Modification of Hybrid Method (ACO+GPE)

DISCUSSION:

In the second hybrid modification method, we implement the GPE algorithm to all agent's solutions in each iteration as shown in Figure III.6. After all solutions have been optimized by GPE, the solution with the minimum total distance is selected as a feasible solution.

III.8.1. The results of the second model (ACO+GPE):

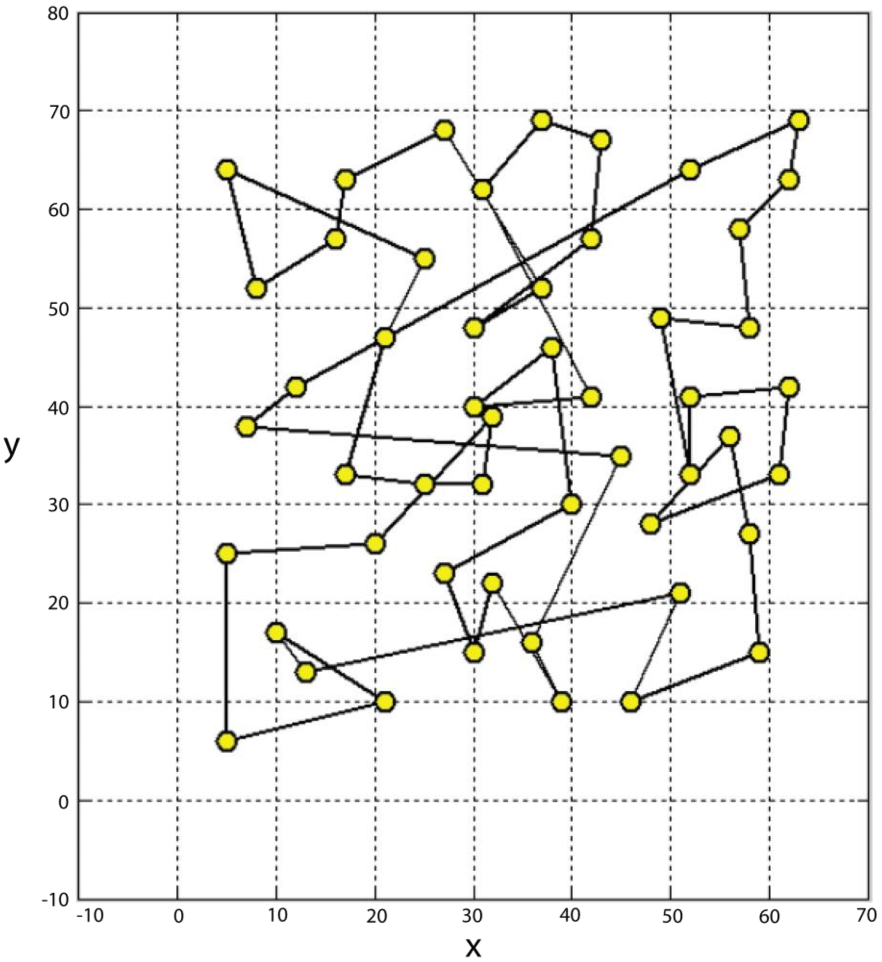


FIGURE III.9 Best solution of EIL51 after 10 runs

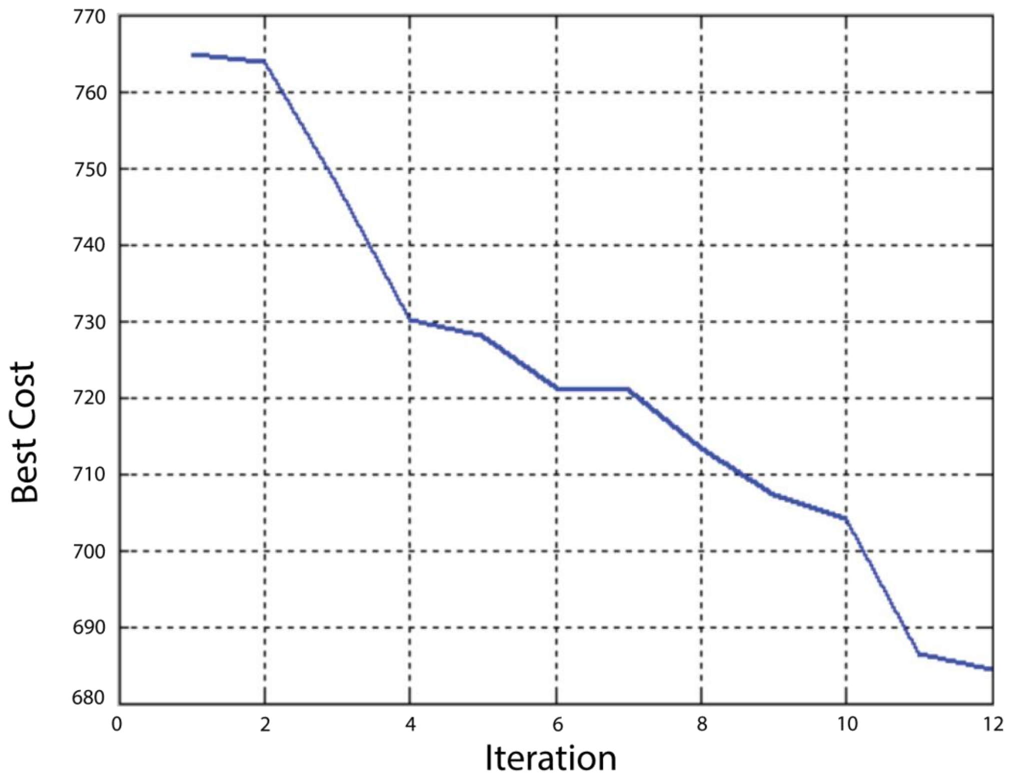


FIGURE III.10 Best cost of EIL51 after 10 runs

PROBLEMS	BKS SOLUTION	BEST SOLUTION	WORST SOLUTION	MEAN SOLUTION	ER (%)	TIME (SECOND)
EIL51	426	684.45	738.22	709.004	66.43	42.36
BERLIN52	7542	11259.15	11938.42	11597.63	53.77	44.15
ST70	675	1181.74	1306.82	1235.95	83.10	60.47
EIL76	538	964.70	1096.69	1044.36	94.11	67.44
RAT99	1211	2534.78	2802.79	2679.08	121.22	105.27
KROA200	29368	76517.86	85972.50	82662.87	181.47	441.96

TABLE III.4. The results of the second model (ACO+GPE)

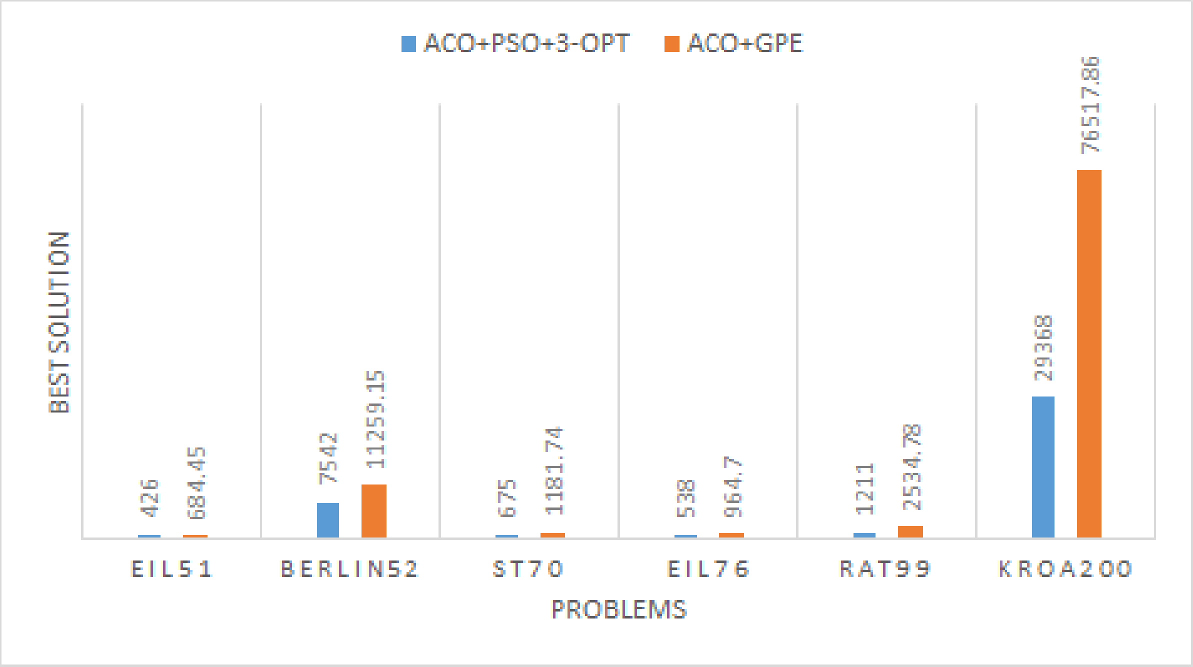


FIGURE III.11 The differences between the best solutions of the two methods

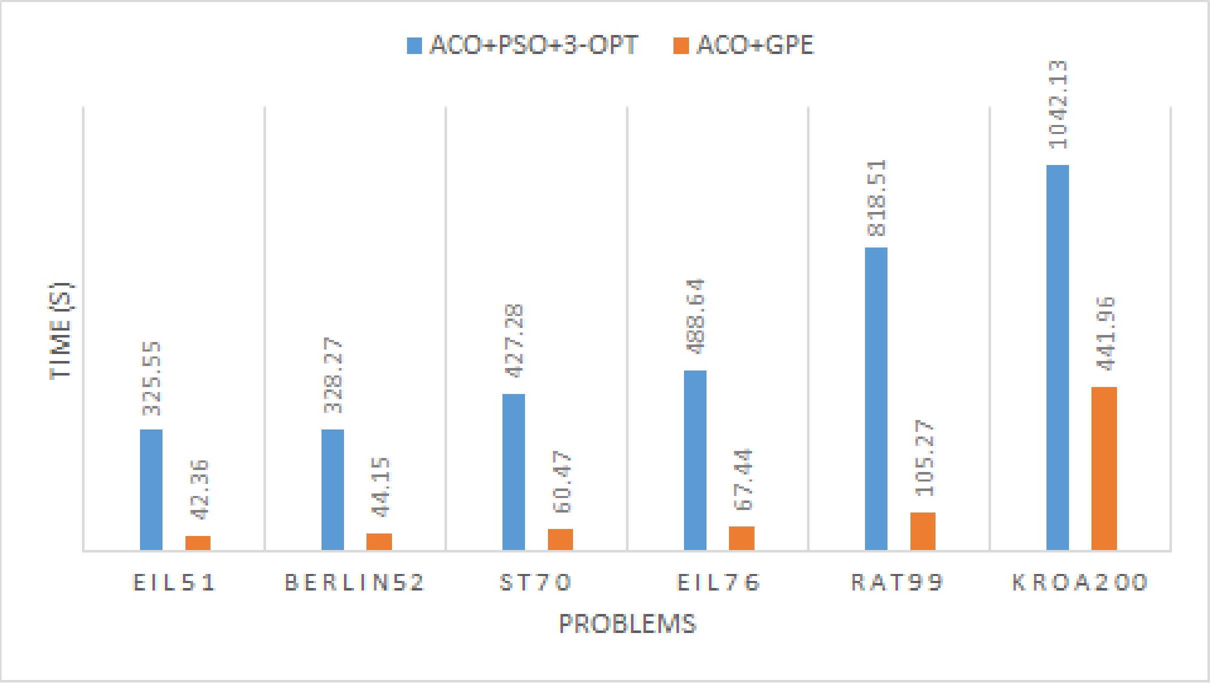


FIGURE III.12 The differences between the times of the two methods

III. 9. Discussion about the second hybrid modification method:

Table III.4 shows that the second hybrid modification method does not perform better for all problems. It shows the corresponding best solution is larger than the best-known solution (BKS). Furthermore, the corresponding ER is also too high especially (RAT99 AND KROA200). But the results show less running time.

The SECOND hybrid modification shows that the (ACO+PSO+3-opt) gives better results of best solutions than (ACO+GPE) but when we look at the time the (ACO+GPE) gives better results.

SO, TO SUMMARIZE:

The results of the two hybrid modifications in general show us that the existing model which is (ACO+PSO+3-OPT) is better than the (ACO+GPE) model.

III. 10. CONCLUSION:

In this chapter we applied the proposed model (ACO+GPE), then we compared their results with the results of the existing model (ACO+PSO+3-OPT).

CONCLUSIONS AND PERSPECTIVES:

In this work, we are interested in studying the traveling salesman problem by proposing a hybrid method for solving the TSP , this method combines two algorithms, ant colony optimization and general pairwise exchange (ACO+GPE), then comparing the results with another existing method which combining three algorithms ant colony optimization, particle swarm optimization and 3-OPT algorithm (ACO+PSO+3-OPT).

We have mentioned combinatorial optimization problems and among this problems the well-known traveling salesman which ranges among NP-hard problems

We discuss Ant Colony Optimization (ACO), which belongs to the group of evolutionary techniques and presents the approach used in the application of ACO to the TSP. We also discuss the general pairwise exchange (GPE)

We investigated the capabilities of the hybrid method for solving optimization problems better than improvement methods alone

Finally, we applied the algorithm on five distance matrix (EIL51, BERLIN52, ST70, EIL76, RAT99, and KROA200) and compare the results with the results of the other model, finding that the other model gives better results but the proposing model takes shorter time.

We will end this thesis with the following perspectives:

- We suggest adding the particle swarm optimization PSO to this proposing model and see if (ACO+GPE+PSO) model is good for solving the traveling salesman problem or not.
- You have to learn how to program earlier.

References

[1] Mme Troudi Fatiha ; Résolution du problème de l'emploi du temps : Proposition d'un algorithme évolutionnaire multi objectif (thèse de magistère) ; 2005-2006 ; Université Mentouri – Constantine

[2] SAHA ADEL ; Résolution des Problèmes Multi Objectifs à Base de Colonies de Fourmi (thèse de magistère) ; 2006-2007 ; l'université de BATNA

[3] Kherbouche Lynda et Oubahri Zohra ; Quelques méthodes de résolutions en optimisation combinatoire (MÉMOIRE DE MASTER) ; Octobre 2017 ; UNIVERSITE MOULOUD MAMMERI, TIZI OUZOU.

[4] Jobil Louis (January 2, 2018 at 3:30pm); Optimization techniques: Finding maxima and minima; Techtarget. https://www.datasciencecentral.com/profiles/blogs/optimization-techniques-finding-maxima-and-minima

[5] G. J. Chaitin et. Al; Register allocation via coloring; in Computer Languages; 6:47-57; 1981.

[6] Santosh Vempala; Combinatorial Optimization, NP-completeness; November 18 (2003). https://ocw.mit.edu/courses/mathematics/18-433-combinatorial-optimization-fall-2003/lecture-notes/l20.pdf .

[7] DEFFAS ZINEB ; Métaheuristiques parallèles à solution unique pour la résolution du problème du Q3AP sur grille de calcul (Magister) ; 29/04/2010 ; l'université d'OUM EL BOUAGHUI.

[8] C. S. H. Zar & A. K. May, Solving Traveling Salesman Problem by Using Improved Ant Colony Optimization Algorithm, International Journal of Information and Education Technology, vol. 1, no. 5, December 201 (ACO).

[9] Konstantinos E. Parsopoulos and Michael N. Vrahatis ; particle swarm optimization for constrained optimization problems; 2002 ;Department of Mathematics, University of Patras l

[10] International Journal of Computer Applications (0975 – 8887); January 2011; Particle Swarm Optimization: Technique, System and Challenges.

[11] Binitha S, S Siva Sathya; A Survey of Bio inspired Optimization Algorithms; International Journal of Soft Computing and Engineering (IJSCE); May 2012

[12] Iztok Fister Jr.1, Xin-She Yang2 , Iztok Fister1 , Janez Brest1 , Dusan Fister1;A Brief Review of Nature-Inspired Algorithms for Optimization; 2013; 1University of Maribor Faculty of electrical engineering and computer science, Smetanova 17, 2000 Maribor, Slovenia 2University of Middlesex School of Science and Technology, Middlesex University, London NW4 4BT, United Kingdom; REVIEW SCIENTIFIC PAPER.

[13] Wang, Z.; Qin, C.; Wan, B.; Song, W.W. A Comparative Study of Common Nature-Inspired Algorithms for Continuous Function Optimization. Entropy 2021, 23, 874. https://doi.org/10.3390/e23070874. 8 July 2021

[14] Harvard Forest (Established in 1907) Harvard University's 4000-acre laboratory; Body Structure. https://harvardforest.fas.harvard.edu/ants/body-structure

[15]https://www.vectorstock.com/royalty-free-vector/diagram-showing-parts-ant-vector-10112404 13/08/2021; 08:00AM

[16] Bendib sami,Benmessahel younes; implémentation de l’algorithme de colonie de fourmie sur le problème de voyageur de commerce(mémoire de fin d’étude; (2015/2016); université Mohamed el Bachir el Ibrahimi; Bordj Bou Arreridj.

[17] HASAN GUCLU (2004-10-01 00 :00 :00) ; COLLECTIVE INTELLIGENCE IN ANT COLONIES ; the fountain magazine ; https://fountainmagazine.com/2004/issue-48-october-december-2004/collective-intelligence-in-ant-colonies .

[18] How Ants Communicate - AntKeepers - Ant facts, Keeping ants and more; https://www.antkeepers.com/facts/ants/communication/ 02/06/2021 07:00 AM

[19] Tate Holbrook, Rebecca Clark, and Brian Haney; Secrets of a Super organism; Arizona State University; 20/08/2021 10:00 AM; https://askabiologist.asu.edu/explore/secrets-superorganism

[20] Wikimemoires (02/2014) ; Méta-heuristiques d’optimisation par colonie de fourmis.

[21] Hazem Ahmed and Janice Glasgow; Swarm Intelligence: Concepts, Models and Applications (Technical Report 2012-585); February 2012; School of Computing Queen's University Kingston, Ontario, Canada K7L3N6.

[22] Soner Haldenbilen, Ozgur Baskan and Cenk Ozan; An Ant Colony Optimization Algorithm for Area Traffic Control; February 20th 2013; https://www.intechopen.com/chapters/38724 .

[23] Christian Blum;ALBCOM, LSI, Universitat Politècnica de Catalunya, Jordi Girona 1-3, Campus Nord, 08034 Barcelona, Spain Accepted 11 October 2005; Ant colony optimization: Introduction and recent trends; Physics of Life Reviews 2 (2005) 353–373

[24] Thomas STUTZLE and Marco DORIGO IRIDIA, ACO Algorithms for the Traveling Salesman Problem; APRIL 1999; Universit´e Libre de Bruxelles, Belgium.

[25] Marco Dorigo, Mauro Birattari, and Thomas Stu¨tzle; Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique; (2006); Universite´ Libre de Bruxelles, BELGIUM.

[26] Mostafa Mahia, Ömer Kaan Baykan and bHalife Kodazb; A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem; 11 February 2015; Department of Computer Engineering and Information Technology, Payame Noor University, Tehran, Iran; Department of Computer Engineering, Faculty of Engineering, University of Selcuk, Konya, Turkey. https://www.sciencedirect.com/science/article/abs/pii/S1568494615000940

[27] Hertono. G.F., Ubadah, Handari, B.D.; The modification of hybrid method of ant colony optimization, particle swarm optimization and 3-OPT algorithm in traveling salesman problem; Department of Mathematics, Universitas Indonesia, Depok, 16424, Indonesia.

Journal of Physics: Conference Series, Volume 974, International Conference on Mathematics: Pure, Applied and Computation 1 November 2017, Surabaya, Indonesia.