Multimodal Ant Colony Optimization for Mixed-Variable Optimization Problems

Muhammad Awais Ali
16i-1047

Masters of Science (Computer Science)

Department of Computer Science
National University of Computer & Emerging Sciences

Islamabad, Pakistan.

July 2018

Multimodal Ant Colony Optimization for Mixed-Variable Optimization Problems

Submitted by

Muhammad Awais Ali
16i-1047

Supervised by

Dr. Waseem Shahzad

Masters of Science (Computer Science)

A thesis submitted in partial fulfillment of the requirements for the degree of
Masters of Science (Computer Science)
at National University of Computer & Emerging Sciences

Department of Computer Science
National University of Computer & Emerging Sciences

Islamabad, Pakistan.

July 2018

Plagiarism Undertaking

I take full responsibility of the research work conducted during the Masters Thesis titled *Multimodal Ant Colony Optimization for Mixed-Variable Optimization Problems* . I solemnly declare that the research work presented in the thesis is done solely by me with no significant help from any other person; however, small help wherever taken is duly acknowledged. I have also written the complete thesis by myself. Moreover, I have not presented this thesis (or substantially similar research work) or any part of the thesis previously to any other degree awarding institution within Pakistan or abroad.

I understand that the management of National University of Computer and Emerging Sciences has a zero tolerance policy towards plagiarism. Therefore, I as an author of the above-mentioned thesis, solemnly declare that no portion of my thesis has been plagiarized and any material used in the thesis from other sources is properly referenced. Moreover, the thesis does not contain any literal citing of more than 70 words (total) even by giving a reference unless I have the written permission of the publisher to do so. Furthermore, the work presented in the thesis is my own original work and I have positively cited the related work of the other researchers by clearly differentiating my work from their relevant work.

I further understand that if I am found guilty of any form of plagiarism in my thesis work even after my graduation, the University reserves the right to revoke my Masters degree. Moreover, the University will also have the right to publish my name on its website that keeps a record of the students who plagiarized in their thesis work.

____

Muhammad Awais Ali

Date: ____________________

Author's Declaration

I, Muhammad Awais Ali, hereby state that my Masters thesis titled *Multimodal Ant Colony Optimization for Mixed-Variable Optimization Problems* is my own work and it has not been previously submitted by me for taking partial or full credit for the award of any degree at this University or anywhere else in the world. If my statement is found to be incorrect, at any time even after my graduation, the University has the right to revoke my Masters degree.

Muhammad Awais Ali

Date:

Certificate of Approval

It is certified that the research work presented in this thesis, entitled "Multimodal Ant Colony Optimization for Mixed-Variable Optimization Problems " was conducted by Muhammad Awais Ali under the supervision of Dr. Waseem Shahzad.

No part of this thesis has been submitted anywhere else for any other degree.

This thesis is submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of Masters of Sciencein Computer Science

at the

National University of Computer and Emerging Sciences, Islamabad, Pakistan

July' 2018

Candidate Name: Muhammad Awais Ali Signature: ____________

Examination Committee:

1. Name: Dr. Hasan Mujtaba Kayani Signature: ____________
   Assistant Professor, FAST-NU Islamabad.
2. Name: Dr. Labiba Fahad Signature: ____________
   Assistant Professor, FAST-NU Islamabad

Dr. Zohaib Iqbal ____________
Graduate Program Coordinator, National University of Computer and Emerging Sciences, Islamabad, Pakistan.

Dr. Waseem Shahzad ____________
Head of the Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan.

Abstract

Existing algorithms mostly focus on solving continuous or discrete optimization problems, rather than solving mixed variable optimization problems. This paper focuses on solving multi-modal mixed-variable optimization problems using ant colony optimization algorithm. The solution construction step of the ants in ACO and the structure of the solution archive is modified to construct solutions for mixed decision variables. Furthermore, to achieve multiple optimal solutions clustering method is used for distributing search space into different clusters, which ultimately increases the chances of finding global optimal solutions respectively. The proposed approach needs to find all the optima simultaneously whereas the computational resources are very limited. Therefore, the diversity enhancement operation is introduced in $MV - MACO$ to avoid unnecessary search efforts. Moreover, a detailed analysis is performed on the results obtained with the results of the state of the art techniques and $MV - MACO$ outperforms the mentioned state of the art algorithms under most conditions.

Acknowledgements

I would like to thank all the people who made this possible specially my supervisor and my family.

Dedication

This is dedicated to my parents.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Swarm intelligence is a system of natural or artificial individuals, that are self-organized and learn from their collective behaviour with co-individuals and environment. An individual can be an agent, that interact with each other and with their environment, in order to achieve a certain target or a goal. There are different types of systems present in swarm intelligence, but this thesis will focus on Ant Colony Optimization a branch of swarm intelligence proposed by Dorigo in 1990s Socha and Dorigo (2008).

For many real-world problems, decision makers prefer to have multiple solutions to a problem before the final decision can be made Li et al. (2017). This is important because if one solution is not applicable, then the other solution can be adopted immediately. Seeking multiple solutions for a particular problem usually helps to explore the hidden properties of the real world problem. There exist some relations which cannot be modelled using mathematical expressions, so having multiple solutions gives a decision maker an option to consider multiple solutions having similar quality. Furthermore, having multiple solutions helps to study the sensitivity of a problem which is a quality step towards providing robust solutions. For finding multiple optima the computational effort is not focused towards the single region of the search space but is distributed in different regions. Thus, finding multiple optima using metaheuristic algorithms[1] increases the chances of finding globally optimal solutions where classical algorithms fail to achieve promising results.

It is common that evolutionary algorithms usually find the global optimal solution after some time, but will ultimately lose it because the population quickly start losing its diversity and prematurely converge to local optima. This is due to the effect of ge-

[1] A meta-heuristic is an algorithmic framework that can be modified a little to apply on different optimization problems.

netic drift[2] Therefore, running evolutionary algorithm to find multiple global optimal solutions in multiple regions of the search space helps to maintain the diversity of the population Zhang et al. (2017).

In literature, many different types of evolutionary algorithms have been proposed to solve real-world problems like particle swarm optimization, differential evolution, ant colony optimization etc. These algorithms mostly focus on solving single optimization problems, rather than solving multi-modal optimization problems because these algorithms are based on global learning and updating schemes which usually tries to guide the whole population towards the single global optimal solution. However, these algorithms cannot be applied directly to solve multi-modal optimization problems.

To tackle multi-modal optimization problems different methods have been proposed in the literature. Among those, the most adopted method is known as niching and speciation. Niching is basically the division of an environment and the subpopulation maintained in a specific niche is called species. In terms of optimization, niche refers to the area of a fitness landscape where within the bounds of it an optimal solution resides and species is the subpopulation around this optimal solution.

Furthermore, to solve multi-modal optimization problems, researchers have used existing niching strategies along with evolutionary algorithms to form new stable niches in every generation and new updating schemes were also introduced for the long-established evolutionary algorithms to solve multi-modal optimization problems. These techniques will be discussed in the upcoming sections.

1.1 Motivation

The proposed techniques to solve multi-modal optimization problems usually suffer from poor performance on irregular multi-modal fitness landscapes. Even these techniques are mostly influenced by sensitive parameter settings and perform well on a particular fitness landscape. Furthermore, reduction in the performance of these evolutionary algorithms is observed in the literature as the dimensions and the complexity of the problem increases Yang et al. (2017). Moreover, most of the real world problems are modelled using both discrete[3] and continuous variables[4], whereas these proposed

[2]The population losing quickly it's diversity and prematurely converges to local optima.
[3]Can only take certain values
[4]Can take any value, within a range

multi-modal techniques fail to handle both of these variables together.

1.2 Problem Statement

The purpose of this research is to modify the ant colony optimization algorithm to tackle mixed-variable multi-modal optimization problems. Many attempts have been made to model the initial ant colony algorithm for solving real-world problems but these modified algorithms have inferior performance, only tackle continuous optimization problems, or lose efficiency when the dimensions of the problem increase.

1.3 Goals and Objectives

We will focus on extending the ant colony optimization algorithm to solve multi-modal mixed-variable optimization problems. Therefore, the above-mentioned drawbacks motivate us to present the proposal of Mixed-Variable Multi-Modal ACO (MV-MACO) for mixed-variable multi-modal optimization problems.

1.4 Scope

This paper will be a focus on modifying ACO to handle mixed-variables. Modifications will be made in the solution construction step of the ants. Furthermore, the structure of the solution archive will be modified to handle both continuous and discrete decision variables. Moreover, to solve multi-modal optimization problem niching methods will be introduced for distributing search space into different regions which ultimately increases the chances of finding globally optimal solutions using ant colony optimization algorithm.

The rest of this paper is organized as follows. Chapter 2 presents the background, chapter 3 presents the literature review, chapter 4 presents the proposed methodology and chapter 5 presents the results and discussion.

Chapter 2

Background

This thesis presents an approach to solve multi-modal mixed variable optimization problem using ant colony optimization algorithm. Before going in to the approach, it is important to describe what we mean by discrete, continuous, mixed variable and multi-modal optimization problems. Moreover, how ant colony algorithm was used to solve such kind of problems in the literature.

2.1 Optimization Problem

In computer science, an *optimization problem* is locating an optimal solution from the set of all feasible solutions. In mathematical terms it can be explained as follows:

Definition 2.1.1 *Given a function f having search space S as domain and the real number $\Re$ as its range. A solution $X^* \in S : \forall_{X \in S}$ is called an optimal solution if it minimizes or maximizes the objective function f.*

Optimization problems are further dependent upon the type of decision variables being used. Therefore, there are three types of optimization problems determined by the decision variable properties of the fitness landscape. Each one of these optimization problem is explained below:

2.1.1 Discrete Optimization Problems

Problems in which decision variables can take on certain values from the available set of values are called discrete optimization problems. The set of available values are predefined before solving the problem and the size of these available values have a significant effect on the performance of the algorithm.

Definition 2.1.2 *Decision variable X_i is discrete if it belongs to a finite set D_i where $i = 1, ..., n$;*

Combinatorial Optimization Problems

As the name indicates, it deals with combinations and permutations of optimal solutions of the given problem. It is the subset of the discrete optimization problem. For the combinatorial optimization problems, there exists no algorithm which can find an optimal solution to a problem in polynomial time. Hence, these problems are solved via using heuristic methods, the methods, which can solve combinatorial optimization problems more quickly by finding an approximate solution instead of using conventional methods. These heuristic algorithms include evolutionary algorithms, particle swarm optimization, ant colony optimization etc.

2.1.2 Continuous Optimization Problems

Problems in which decision variables can take on values from within the certain range of real numbers are called continuous optimization problems.

Definition 2.1.3 *Decision variable X_i is continuous if it belongs to an infinite set of $\Re$ where $i = 1, ..., n$;*

2.1.3 Mixed-Variable Optimization Problems

These optimization problems are an amalgam of both continuous and discrete optimization problems and decision variables are a combination of both categorical and continuous variables.

Definition 2.1.4 *Problems having n decision variables where $n = a + b$. Variable X_i is discrete for domain D_i where $i = 1, ..., a$, and continuous for domain D_i where $i = a + 1, ..., a + b$;*

These types of problems are usually difficult to handle because they possess a difficulty of both continuous and discrete optimization problems.

2.2 Multi-Modal Optimization Problems

Multi-Modal optimization (*MMO*) problems represent an important class of optimization problems. In many real-world problems, decision makers prefer to have multiple solutions to a problem before the final decision can be made Li et al. (2017). This is important because if one solution is not applicable then the other solution can be adopted immediately.

The multi-modal optimization problem can be explained as follows:

Definition 2.2.1 *Given a function f having search space S as domain and real number $\Re$ as its range. A solution $X^* \in S : \forall_{X \in S}$ is called an optimal solution if it minimizes or maximizes the objective function f. Here X is an n dimensional vector having discrete, continuous or is a combination of both discrete and continuous decision variables. For multi-modal optimization problem locate all possible optimal solutions $x \in X^*$ which maximize or minimize the objective function f value.*

The all possible optimal solution peaks of a problem X^* are also surrounded by some inferior solutions known as *localoptima*. This is clearly visible in figure 2.1, which shows the multi-modal fitness landscape having global optimum peaks for Vincent 2D function.

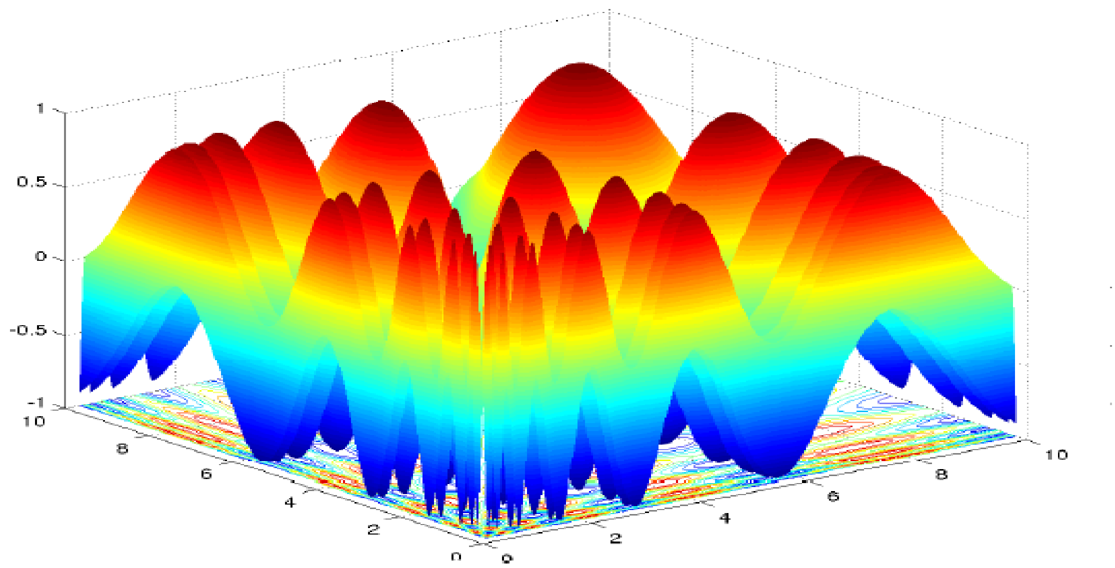

Figure 2.1: Multi-modal fitness landscape having global peaks for Vincent 2D function Li et al. (2017)

2.2.1 Niching and Speciation

To tackle multi-modal optimization problems, different methods have been proposed in the literature. Among those, the most adopted method is known as *niching* and *speciation*. *Niching* is basically the division of an environment and the subpopulation maintained in a specific niche is called *species*. In terms of optimization, niche refers to the area of a fitness landscape where within the bounds of it an optimal solution resides and species is the subpopulation around this optimal solution. Figure 2.2 shows the pairs of clustered global peaks obtained using niching for Shubert 2D function.

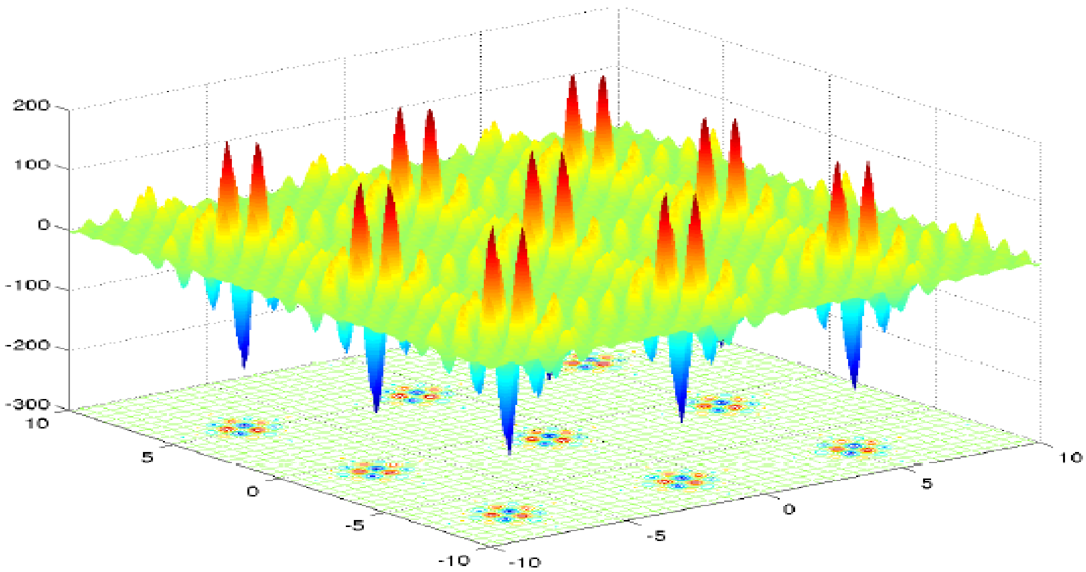

Figure 2.2: The arrangement of the solutions for a multi-modal Shubert 2D function

2.3 Ant Colony Optimization

This section gives a brief introduction of the Ant colony optimization algorithm Socha and Dorigo (2008) proposed by Dorigo in 1992. ACO[1] belongs to the research field known as swarm intelligence, which studies the behavior of swarms. ACO is inspired by the way of behaving of ants when searching for the food source. Ants at first explore the area around their nest in a random manner. As the food source is located, an ant carries some food back to the nest and deposits the pheromone along its route

[1] Ant Colony Optimization

as a guiding mechanism for other ants to follow. This is the example of indirect communication among ants using pheromone which enables ants to find the shortest path between their nest and the food source. By increasing the quality of the path via depositing more pheromone as more ants follow the same path. This mechanism is shown in figure 2.3.

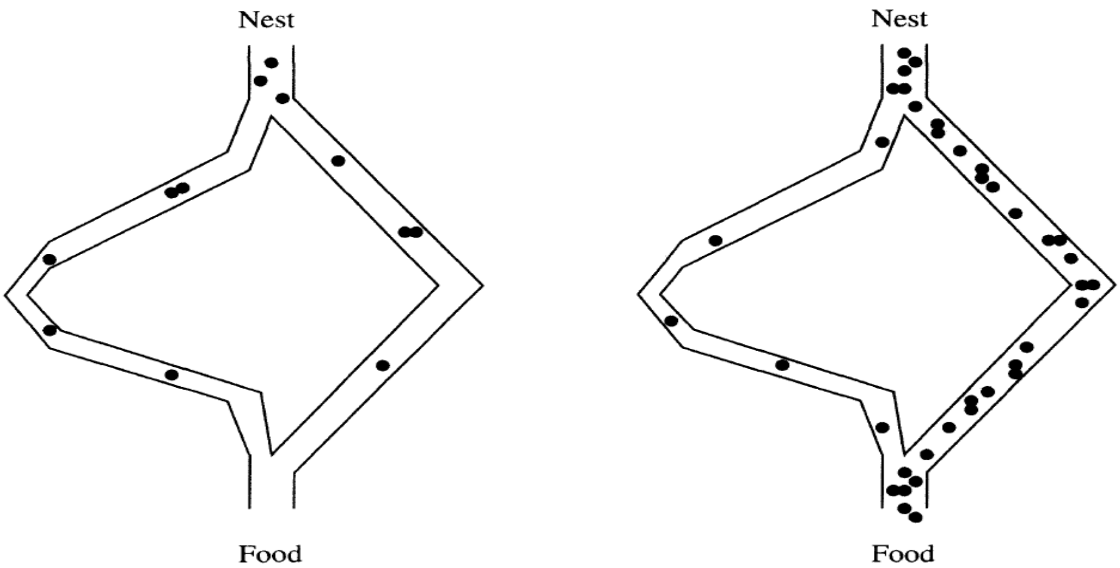

Figure 2.3: Pheromone trail following of ants

2.3.1 ACO Meta-Heuristic Algorithm

The ACO meta-heuristic algorithmic framework is shown in Algorithm 1. In the fol-

Algorithm 1 The Ant Colony Meta-heuristic Algorithm (adopted from Socha and Dorigo (2008))

```
Initialize parameters, initialize pheromone trials
while  termination conditions are not met do
    AntBasedSolutionConstruction
    UpdatePheromone
    ApplyDaemonActions
end while
```

lowing subsections ACO algorithm is explained in more detail.

Ant Based Solution Construction

Ants construct solutions from n available solution components. The ant select solution component probabilistically as shown in equation 2.1.

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \in J_i^k} [\tau(t)]^{\alpha} \cdot [\eta]^{\beta}} \tag{2.1}$$

where τ_{ij} is the pheromone amount related with the solution component and η_{ij} is a heuristic value for each solution component. Here, α and β are the parameters, whose values control the influence of pheromone value and the heuristic information in the selection of solution component Socha and Dorigo (2008).

Pheromone Update

The purpose of pheromone update is to the increase the pheromone influence on the paths with promising solutions and decrease on those having worst solutions Socha and Dorigo (2008). This is achieved by using the equation 2.2.

$$\tau_{ij} = \begin{cases} (1-\rho) \cdot \tau_{ij} + \rho \Delta \tau & \text{if } \tau_{ij} \in \text{chosen good solution} \\ (1-\rho) \cdot \tau_{ij} & \text{otherwise,} \end{cases} \tag{2.2}$$

where ρ is the evaporation rate, to avoid premature convergence on local optima, thus favoring the exploration. All other variants of the ant colony optimization algorithm differ in a way how they update the pheromone. They either update the pheromone associated with iteration best solution or associated with best so far solution found from the start of the algorithm.

Daemon Actions

Daemon action is available to implement but is not compulsory. It can be used to apply actions like local search to refine the obtained solutions or whether it is favorable to deposit more pheromone to bias the search process via using the global information Yang et al. (2017).

2.3.2 ACO to Solve Discrete Optimization Problem

One of the applications of the ACO is to solve the traveling salesman problem STUTZLEz et al. (1999). To summarize how ACO can be applied to solve TSP[2] problem a pseudo-code algorithm 2 is given below: This section discusses the ACO with reference to the TSP problem. ACO was applied to solve many optimization problems like scheduling problem, assignment problem etc. In the next section, ACO is discussed with reference to solving continuous optimization problems.

Algorithm 2 Ant colony optimization (ACO) for the TSP (adopted from STUTZLEz et al. (1999))

```
Initialize the pheromone on each edge between cities a and b to small positive random values;
Place all ants m ∈ 1, ..., m on the starting city;
Let S+ be the shortest trip, and L+ the length of that trip;
for t = I to tmax do
    For each ant, build the trip S_m(t) by choosing the next city n-1 times (n is the number of cities), with probability calculated using equation 2.1;
    Compute the length of the route, L_m(t), of each ant;
    If an improved route is found, update S+ and L+;
    Update the pheromone deposits on each edge using equation 2.2;
end for
Output the shortest route S+.
```

---

[2]Traveling Salesman Problem

2.3.3 ACO to solve continuous optimization problems

ACO is successful in solving many real-life problems which were considered as NP-hard problems, but since now, no one has focused on solving the continuous optimization problems. The authors in Socha and Dorigo (2008) had modified the ant colony algorithm to make it useful for solving continuous optimization problems, by proposing the modified version named as ACO_R. They have modified the discrete probability distribution used by the ants to select the new solution component to a continuous by using Gaussian probability density function as shown in equation 2.3.

$$g\left(x^d, \mu^d, \sigma^d\right) = \frac{1}{\sigma^d\sqrt{2\pi}} e^{-\frac{\left(x^d-\mu^d\right)^2}{2\left(\sigma^d\right)^2}} \tag{2.3}$$

where μ is defined as the mean of the dimension of the problem and standard deviation σ is calculated as follows in equation 2.4:

$$\sigma^d = \xi \sum_{i=1}^{\mathrm{NP}} \frac{|x_i^d - x_j^d|}{\mathrm{NP}-1} \tag{2.4}$$

This is the average distance between i^{th} continuous variable and j^{th} continuous variable, where ξ parameter has the same effect as the pheromone persistence in discrete ACO, whose value further controls the exploration and exploitation behavior of the algorithm. In this algorithm it's value is initialized via uniform random generation between (0,1].

Furthermore, the pheromone component of a solution is replaced by the weight of the solution, which affects the selection probability of the solution as shown in equation 2.5.

$$w_j = \frac{1}{q\mathrm{NP}\sqrt{2\pi}} e^{-\frac{(\mathrm{rank}(j)-1)^2}{2q^2\mathrm{NP}^2}} \tag{2.5}$$

where $rank(j)$ return the rank of the j^{th} solution and q is a parameter of the algorithm. It has a special effect on the weight of the solution. With small value of q

top ranked solutions will be preferred, where as the large value suggests a uniform probability distribution of the solutions.

In ACO_R solutions are kept in solution archive as shown in figure 2.4.

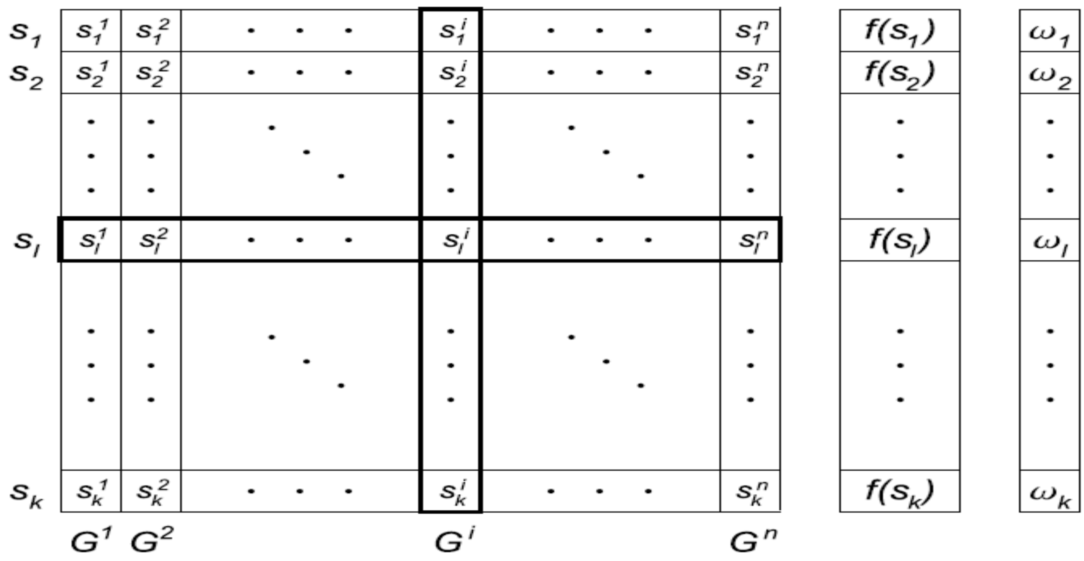

Figure 2.4: The arrangement of the solution archive k used by ACO_R (adopted from Socha and Dorigo (2008))

The solution archive k contains k complete solutions of the problem. An ant select probabilistically a solution form the solution archive k using following probability in equation 2.6:

$$p_j = \frac{w_j}{\sum_{i=1}^{\mathrm{NP}} w_i} \tag{2.6}$$

where w_j is the weight of the solution calculated using equation 2.5 and each ant iteratively constructs solution for every variable of a problem in each iteration using the normal probability density function having mean μ and standard deviation σ as defined in equation 2.3.

Chapter 3

Literature Review

Several techniques have been discussed in the literature to solve mixed variable multi-modal optimization problems. Algorithms proposed in recent years, either try to solve the problem with inferior performance, or they are adaptive to a particular landscape and parameter settings which restricts them to perform optimally on selected problems, and perform poorly on irregular multi-modal fitness landscape Yang et al. (2017). This chapter gives an overview of the techniques proposed in the literature.

3.1 Methods Proposed to Solve Multi-modal Optimization Problems

The existing algorithms use global learning strategies which tries to drive the whole population towards one global optima, which is obviously not the case with the multi-modal optimization problems. Hence, it's not an easy task to apply these evolutionary algorithms directly on multi-modal optimization problems. Therefore, researchers have modified the already existing algorithms to tackle multi-modal optimization problems. The recent work on these modified techniques is discussed in this section.

In recent years most of the research done on multi-modal optimization is focused on niching techniques. An overview of the niching is already given in chapter 2. Evolutionary algorithms integrated with niching techniques are now a days widely used to obtain multiple optimal solutions of a problem. Li et al. (2017) provided an updated survey on niching methods. Furthermore, another survey paper Dasa et al. (2011), presents a comprehensive review of the existing niching techniques with a detailed explanation of their complexity and how these niching techniques were integrated with different evolutionary algorithms to solve multi-modal optimization problems.

As it is already known from the literature that evolutionary algorithms have some limitations in exploitation and exploration, therefore in recent years researchers have proposed new techniques, which incorporate the niching methods with these evolutionary algorithms as a new update strategy to deal with multi-modal optimization problems and to enhance the abilities of these algorithms in terms of exploitation and exploration. In Gao et al. (2014), the authors have incorporated the multi population strategy in differential evolution algorithm to deal with multi modal optimization problems. The authors have partitioned the whole population in the form of clusters, where clusters are representing different promising search regions in the whole search space. Hence, different subpopulations can locate different optimal solutions. Even a self adaptive parameter control strategy was applied to improve the search ability of differential evolution algorithm for subpopulations. Furthermore, in Qu et al. (2012), authors have proposed neighborhood mutation operator and integrated it with the existing variants of niche based differential evolutionary algorithms namely CDE[1], SDE[2] and sharing DE. This neighborhood mutation operator works by performing mutation within each euclidean neighbor hence, preserving stable niche behavior and is able to locate multiple global optima. Furthermore, existing speciation methods were also improved to solve multi-modal optimization problems. In Li and Tang (2015), authors have proposed a parameter free speciation method called history based topological speciation to solve multi-modal optimization problems. Traditional topological based niching methods require sampling and evaluation of new individuals to get the overview of the fitness landscape, which usually results in additional fitness evaluations.

In literature swarm based algorithms were also modified to solve multi-modal optimization problems. In Fieldsend (2014), authors have used particle swarm optimization algorithm for solving multi-modal problem. The proposed algorithm exploits the regions that lie outside the hyper-cuboid structure of PSO[3] and used multiple swarms, each exploiting multiple regions in the search landscape, searching for multiple optimal solutions. These swarms can merge if they lie in the regions having same peak of optimal solutions. Furthermore, niche based particle swarm optimizers are extensively used by the research community to solve multi-modal optimization problems, but they require prior knowledge about the multi-modal fitness landscape to specify niching parameters and often they have poor performance, therefore, in Qu et al. (2013), authors have proposed improved PSO which definitely does not need to specify niching

[1]DE with a crowding scheme called crowding DE (CDE)
[2]DE with speciation scheme
[3]Particle Swarm Optimization Algorithm

parameters and have improved performance on multi-modal fitness landscapes. This proposed approach uses many local best individuals to guide the search process of individuals instead of using global best individual. Moreover, stable niches are formed via estimating the euclidean distance with its neighbors. In Yang et al. (2017), authors have used ant colony algorithm to make it useful for solving continuous multi-modal optimization problems. They have modified the algorithm used by Socha and Dorigo (2008) to solve multi-modal optimization problems. Furthermore, crowding technique have been incorporated to obtain multiple optimal solutions of a problem. Differential evolution mutation operator was used to increase the performance of the algorithm. Furthermore, local search have been introduced to increase the value of the solution.

Due to the growing interest of the research community to apply evolutionary algorithm to solve multi-modal optimization problems, few of them come up with the solutions that use multi-objective optimization techniques to solve multi-modal optimization problems. Wang et al. (2015) have proposed a transformation technique called MOMMOP which basically transforms the multi-modal optimization problem in to two conflicting objectives which can be used to find the optimal solutions of the problem. Furthermore in Basak et al. (2013), introduced a bi-objective technique which uses the differential evolution with non-dominated sorting technique to get multiple global and local optimal solutions for a particular problem to solve.

3.2 Methods Proposed to Solve Mixed-Variable Optimization Problems

In literature not much effort has been put forward by researchers to solve mixed-variable optimization problems using Ant Colony Optimization. In Liao et al. (2012), authors have presented an hybrid approach based on ACO_R proposed by Socha and Dorigo (2008) for solving mixed-variable optimization problems. They have proposed three approaches, one of them implements ACO_R with Hooke and Jeeves local search scheme, the second one is a cooperation between ACO_R and differential evolution approach, and the third one is obtained by integrating the second approach with the local search scheme. All of the proposed approaches are more efficient in finding optima as compared to ACO_R. Furthermore, in Liao et al. (2014) ,authors have further improved their proposed approach in Socha and Dorigo (2008) by incorporating ACO_{MV} for solving mixed-variable optimization problem. ACO_{MV} constructs solution using ACO_R for continuous variables ,$ACO_{MV} - o$ for ordinal variables and $ACO_{MV} - c$ for tackling categorical attributes of an optimization problem. The authors have also proposed a set of benchmark functions for testing ACO_{MV} on various mixed variable optimization problems.

3.3 Ant Colony Optimization Variants for Solving Continuous Optimization Problems

In literature a lot of work have been proposed by the researchers to modify discrete version of ACO to solve continuous optimization problems. Impressive work was done by Socha and Dorigo (2008), they had modified the ant colony algorithm to make it useful to solve continuous optimization problems by proposing the modified version named as ACO_R. They have modified the discrete probability distribution used by the ants to select the new solution component to a continuous by using Gaussian probability density function. Most of the work in the literature is mostly the extension of the approach proposed by Socha and Dorigo (2008), either researchers have extended the ACO_R or incorporated local search schemes to improve its performance. In Liao et al. (2011), authors have proposed a variant of ACO_R with local search mechanism to further improve the obtained solutions, which significantly improves the performance of ACO_R on continuous fitness landscape. Furthermore, in Liao et al. (2014), the authors have proposed UACOR algorithm for continuous optimization problems. This new approach is a combination of ACO_R proposed by Socha and Dorigo (2008), $DACO_R$ and $IACO_R - LS$ proposed by Liao et al. (2011), all three previously proposed ACO algorithms for solving continuous optimization problems. Furthermore, Xiao-Min et al. (2010) proposed a new framework $SamACO$ which uses an effective discretization method to sample the continuous fitness landscape, so traditional ACO can operate to solve continuous optimization problems. Moreover, authors of Chen et al. (2017) proposed a robust ant colony optimization algorithm with adaptive domain adjustment, pheromone increment, search space division and adaptive ant size without applying any major changes in the traditional Ant Colony optimization algorithm for solving continuous optimization problems.

3.4 Hybrid ACO Models

Few of the literature have used existing evolutionary algorithms along with Ant colony algorithm to enhance the search capabilities of ACO. These hybrid approaches have shown massive improvements in the runtime and the accuracy of the ACO. In Gong et al. (2015), the authors have modified the ACO_R algorithm proposed by Socha and

Dorigo (2008) through introducing an operation similar to crossover operation in genetic algorithm, which is used to generate new probability density function set in the fitness landscape. In $COACO_R$, authors have used three crossover methods $BLX - \alpha$, $UNDX$ and PNX to improve the performance of ACO_R and after crossover the resultant pheromone information complement the algorithm in finding the global optimal efficiently. From the results of new algorithm when compared with other evolutionary algorithms, shows that the fitness evaluations of the proposed algorithm are much fewer than those of other algorithms. Moreover, in Ciornei and Kyriakides (2012) hybridization of RCGA a version of genetic algorithm for continuous optimization problems and API a class of ant colony algorithm for continuous domain is implemented to prevent individuals from getting struck in local optima due to non smoothness of fitness landscape. Furthermore, in Lin et al. (2016), authors have proposed a scheme that uses principles of both particle swarm optimization and ant colony optimization. This simple population based meta-heuristic is a simple composition of two operations, namely SELECT+COPY and RANDOM, for creating a new solution. Moreover, in Xiao and Li (2011), another hybrid approach was proposed which uses differential evolution for calculating Gaussian mean values for generating next generation of continuous ant colony population.

3.5 Literature Review Synthesis

Applying evolutionary algorithms to solve optimization problems is a challenging task. Different algorithms have been used to solve difficult problems, but these algorithms mostly focus on solving single optimization problems, rather than solving multi-modal optimization problems. Because these algorithms are based on global learning and updating strategies which usually tries to drive the whole population towards single optimal solution. Therefore, niching methods were incorporated to solve multi-modal optimization problems. Existing niching strategies along with evolutionary algorithms were used to form new stable niches in every generation and updating strategies were also proposed for existing algorithms. Multi-modal optimization problems were also solved via multi-objective optimization techniques. The transformation techniques were proposed to transform multi-modal optimization problem to multi-objective optimization and is then solved using multi-objective optimization technique. Most of the work was focused on to solve continuous optimization problems using ACO. Most of the proposed approaches have tried to improve the effectiveness of ACO_R. Much limited work is put forward to solve mixed-variable optimization problems. Even few of the hybrids methods were proposed which uses existing evolutionary algorithms along with ACO to solve optimization problems with promising results.

3.6 Comparative Analysis

Reference	Objective	Approach	Limitations
Qu et al. (2012)	Solve MMO problem using EA	Used DE algorithm with proposed mutation operator	Mixed variables not handled. Dimension and complexity dependent
Gao et al. (2014)	Solve MMO problem using EA	Used cluster based DE with adaptive parameter settings	Mixed variables not handled. Dimension and complexity dependent
Li and Tang (2015)	Solve MMO problem using EA and niching technique	History based topological parameter free speciation method	Mixed variables not handled. Dimension and complexity dependent
Fieldsend (2014)	Solve MMO problem using PSO and niching technique	niching based migratory multi swarm optimizer	No adaptive parameter settings and mixed variables not handled
Qu et al. (2013)	Solve MMO problem using PSO and niching technique	Distance based locally informed PSO using niching and adaptive parameter settings	Mixed variables not handled. Dimension and complexity dependent
Yang et al. (2017)	Solve MMO using ACO for continuous problems	ACO_R Integrated with niching and differential evolution	Mixed variables not handled. Dimension and complexity dependent
Lin et al. (2016)	Propose scheme for discrete optimization problems	A hybrid approach using PSO and ACO	multi-modal and mixed-variable not handled

Table 3.1: Comparative Analysis (Part 1)

Reference	Objective	Approach	Limitations
Basak et al. (2013)	Solve MMO using Multi-objective optimization	Bi-objective approach based on DE	Mixed variables not handled.Dimension and complexity dependent
Liao et al. (2014)	Solve mixed-variable optimization problem using Ant colony optimization	Improved ACO_R proposed by Socha and Dorigo (2008)	Cannot handle multi-modal optimization problem
Liao et al. (2012)	Solve mixed-variable optimization problem using ACO_R	Integrated ACO_R with DE and local search scheme	Mixed variables not handled. Dimension,parameter and complexity dependent
Liao et al. (2014)	Propose hybrid ACO model to solve optimization problems	Combined ACO_R with $DACO_R$ and $IACO_R - LS$	Multi-modal and mixed-variable optimization problems not handled
Liao et al. (2011)	Modify ACO_R to improve its performance	Integrated ACO_R with local search scheme	Multi-modal and mixed-variable not handled
Xiao-Min et al. (2010)	Propose SamACO to solve continuous optimization problems	Used sampling method to discretize the continuous search space for traditional ACO	multi-modal and mixed-variable optimization problems not handled
Socha and Dorigo (2008)	Solve continuous optimization problems using ACO	Proposed ACO_R to solve continuous optimization problems	multi-modal and mixed-variable optimization problems not handled
Chen et al. (2017)	Solve continuous optimization using ACO	Robust ACO with adaptive parameter settings	Mixed variables and multi modal not handled.

Table 3.2: Comparative Analysis (Part 2)

Reference	Objective	Approach	Limitations
Wang et al. (2015)	Solve MMO using Multi-objective optimization	Converted MMO to MOO with conflicting objectives	Mixed variables not handled. Dimension and parameter dependent
Gong et al. (2015)	Integrate GA crossover operator in ACO_R	Integrated $BLX-\alpha$, $UNDX$ and PNX in ACO_R	Mixed variables and multi modal not handled
Xiao and Li (2011)	Solve continuous optimization problems using hybrid ACO	Incorporated differential evolution to generate Gaussian mean values for new population	Mixed variables and multi modal optimization problems not handled
Ciornei and Kyriakides (2012)	Solve continuous optimization problem	Hybrid approach based on ACO and GA	Mixed variables not handled. Dimension and parameter dependent

Table 3.3: Comparative Analysis (Part 3)

3.7 Critical Summary

Most of the adopted methods to solve multi-modal optimization problems Li et al. (2017),Dasa et al. (2011),Gao et al. (2014),Qu et al. (2012),Li and Tang (2015) show effectiveness on tested problems, but these methods experience various issues, such as, they suffer from low performance on ragged multi-modal fitness landscapes. These techniques are mostly influenced by sensitive parameter settings and perform well on particular fitness landscapes. Furthermore, reduction in the performance of these evolutionary algorithms is observed as the dimensions and the complexity of the problem increases. Even more, these techniques are either at the loss of fitness evaluations or consume much of memory space. Furthermore, their performance deteriorates if niches have imbalanced number of individuals. Moreover, most of the real world problems are modeled using both discrete and continuous decision variables, whereas proposed Ant colony based multi-modal techniques Yang et al. (2017),Liao et al. (2012) are developed to tackle either discrete or continuous optimization problems only, they fail to handle mixed-variable multi-modal optimization problems. To the best of our understanding there is no previous work done to solve mixed-variable multi-modal optimization problems using ant colony optimization. Therefore, the above mentioned drawbacks motivate us to present Mixed-Variable Multi-modal ACO (MV-MACO) for solving mixed-variable multi-modal optimization problems.

.

Chapter 4

Methodology

In this chapter, the general details of the Mixed variable optimization problems and the design variables associated with the mixed variable optimization problems are discussed and then explained the proposed $MV - MACO$ algorithm.

4.1 Mixed Variable Optimization Problems

These optimization problems are an amalgamation of both continuous and discrete optimization problem and decision variables are an amalgam of both categorical and continuous variables.

These types of problems are usually tough to handle because they possess the difficulty of both continuous and discrete optimization problems. Therefore we proposed an approach which efficiently handles multimodal mixed variable optimization problems.

The proposed approach $MV - MACO$ is building on the foundations provided by Adaptive multi-modal continuous ant colony optimization by Yang et al. (2017). The details of the proposed technique are discussed in detail in the following sections.

4.2 Proposed Technique ($MV - MACO$)

For solving mixed variable multi-modal optimization problem the solution archive is divided into two parts. One part represents continuous decision variables and the other part represents discrete decision variables. The solution archive S contains k

complete solutions of the problem and this solution archive S is sorted according to the fitness values (best to worst) as shown in figure 4.1.

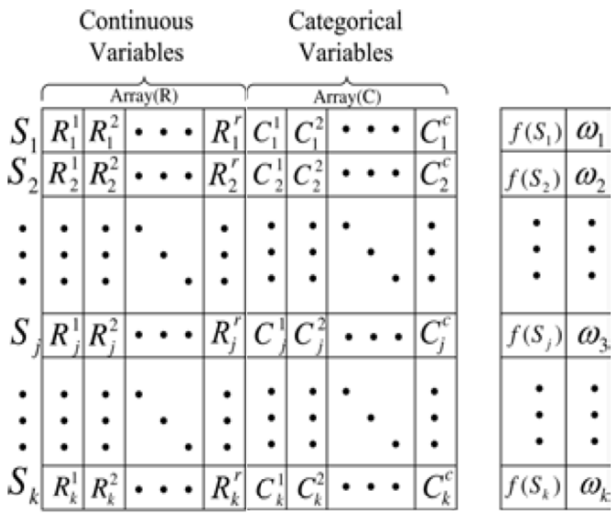

Figure 4.1: The arrangement of the solution archive S used by MV-MACO

K sized population will be initialized randomly, categorical attributes will be chosen randomly from the set of available values, whereas for continuous attributes, specific domain range will be initially assigned to each attribute and they will be randomly chosen from with in this range.

A weight w_j is associated with each solution S_j in the solution archive. Weight value is calculated via a Gaussian function defined in equation 2.5 of chapter 2 in section 2.3.3.

Instead of managing whole solution archive as in traditional ACO_R, $MV-MACO$ operates on the clusters by integrating crowding method. Thus, prior to solution construction by ants, the available solutions in the maintained archive are partitioned into several clusters according to the proposed clustering strategy in algorithm 3.
Each ant probabilistically constructs a solution for the continuous and categorical decision variables. Firstly an ant probabilistically selects the solution from the maintained solution archive S using probability defined in equation 2.6 of chapter 2 in section 2.3.3. Then each ant iteratively constructs the solution for continuous variables using

Algorithm 3 Crowding Based Clustering

Input: Archive k , Cluster Size M

Step1: Generate a reference point REF and calculate its distance to all the population in archive k;

while Archive k is not empty **do**

  Select an individual $INDIV$ near to REF in archive k;

  Create a cluster by combining $INDIV$ and $M-1$ individuals nearest to it in the archive k;

  Remove these M individuals for the archive k;

end while

Output: Return a set of Clusters;

the equation 2.3 of chapter 2 in section 2.3.3, while for each categorical attribute ant chooses one of the available values probabilistically using the formula in equation 4.1:

$$p_T^i = \frac{w_T}{\sum_{j=1}^{\mathrm{t_i}} w_j} \tag{4.1}$$

where w_T is calculated using equation 4.2 below:

$$w_T = \begin{cases} \frac{w_{jT}}{U_T^j} + \frac{q}{\eta} & U_T^j > 0 \text{ and } \eta > 0 \\ \frac{w_{jT}}{U_T^j} & U_T^j > 0 \text{ and } \eta = 0 \\ \frac{q}{\eta} & U_T^j = 0 \text{ and } \eta > 0 \end{cases} \tag{4.2}$$

Where η is the number of available values that are not being used by the solutions in the maintained solution archive S, w_{jT} is the solution having highest quality with value T for the categorical attribute j, U_T^j is the number of solutions in S that use value T for the categorical variable j and q is the same variable used in equation 2.5 of chapter 2 in section 2.3.3. A pseudo-code algorithm for probabilistic solution construction of categorical attributes is given in algorithm 4. After each ant constructs its solution, the new and old solutions will be sorted according to the fitness values (best to worst) and k best solutions will be selected for the next iteration. But before passing these solutions to the next iteration, they are passed on to a diversity enhancement function given in algorithm 5.

Diversity is essential for locating the multiple optima. The proposed approach needs to find all the optima simultaneously, whereas the computational resources are very limited. Therefore, there is a need to incorporate the diversity enhancement operation which aid to avoid the unnecessary search efforts.

Algorithm 4 Probabilistic Solution Construction for Categorical Attributes

Input: Archive k , Categorical variable j

Step1: Calculate η = number of available values that are not used by the solutions in the solution archive k;

for all T for available values of j **do**

Get weight w_{jT} = solution having highest quality with value T for the categorical attribute j;

Calculate U_T^j = number of solutions in k that use value T for the categorical variable j;

if $U_T^j = 0$ and $\eta > 0$ **then**

Weight of value T = q/η, where q is the same variable used in equation 2.5;

end if

if $U_T^j > 0$ and $\eta = 0$ **then**

Weight of value T = w_{jT}/U_T^j;

end if

if $U_T^j > 0$ and $\eta > 0$ **then**

Weight of value T = $w_{jT}/U_T^j + q/\eta$, where q is the same variable used in equation 2.5;

end if

end for

Output: Return weights C_j;

For each cluster when individuals completely converge to the optima, further exploitation in that region is useless and could be a waste of available computational resources and may result in unproductive search efforts. At this moment the enhance diversity operation will save the best individual in the dynamic archive and will reinitialize the cluster for the next iterations. At the beginning of each iteration, the centre of each cluster C_i will be calculated. A distance will be calculated, which is the Euclidean distance between the centre of a cluster C_i and a random individual in the cluster. Suppose, if this distance is less than or equal to ϵ then the population in the cluster C_i has converged. At this moment the enhance diversity operation will add the best individual from the cluster to the dynamic archive and will reinitialize the population of the cluster C_i, reevaluate its fitness and increment the fitness evaluation count by the number of individuals evaluated.
Furthermore, if two clusters overlap, the enhance diversity operation will merge the clusters hence, avoiding unnecessary computation cost and saving a lot of available resources. For merging two clusters, the Euclidean distance between the centres of the two clusters is calculated. If the distance between the centre of C_i and centre of C_j is less than d_0 which is the overlapping threshold then the clusters C_i and C_j can be merged. Suppose the best and the worst individuals in the clusters C_i and C_j are represented as $best_i$,$worst_i$ and $best_j$, $worst_j$ respectively. The two clusters will be merged with the worst best fitness into the better cluster. Assuming, if $f(best_i) \geq f(best_j)$ and $f(best_j)$ $\geq f(worst_i)$ then $worst_i$ will be replaces by the $best_j$ and the cluster C_j will be randomly reinitialized. Otherwise, cluster C_i will dominate cluster C_j and no replacement takes place. Every time when the population in a cluster is reinitialized, it's fitness will be evaluated and fitness evaluation count will be incremented.

The whole process of diversity preservation is described in algorithm 5. After this process, these diversified clusters are feed into the $MV - MACO$ for further optimizing the results. This process will continue until maximum fitness evaluations are reached.

Algorithm 5 Diversity Enhancement

SET $S = empty$
Input: Clusters C
for each i in C **do**
  if $i \in SET$ **then**
    Increment i
  end if
  if dist $\|$ $Center$ of $C_i, Rand \in C_i$ $\| <= \epsilon$ **then**
    Record the best individual of C_i in a dynamic archive
    Reinitialize all the individuals in the cluster C_i
    Evaluate these new individuals
    Append this cluster in to set S
  end if
  for each $j = i + 1$ in C **do**
    if $j \in SET$ or $\|$ $Center$ of $C_i, Center$ of C_j $\| >= d_0$ **then**
      Increment j
    end if
    Get best of i,worst of i,best of j,worst of j
    if f(best of i) $>=$ f(best of j) **then**
      if f(best of j)$>=$ f(worst of i) **then**
        worst of i= best of j
      end if
      Reinitialize all individuals in cluster j
      Evaluate these new individuals
      Append this cluster in to set S
    end if
    if f(best of i) $>=$ f(worst of j) **then**
      worst of j= best of i
    end if
    Reinitialize all individuals in cluster i
    Evaluate these new individuals
    Append this cluster in to set S
    BREAK;
  end for
end for
Output: Return a set of Clusters;

4.3 Overall Strategy for $MV - MACO$

To summarize, the $MV - MACO$ algorithm extends the ACO_R algorithm to solve multimodal mixed variable optimization problems. $MV - MACO$ incorporates the ACO_R and the categorical optimizer proposed in section 4.2 combined with crowding based clustering and diversity enhancement procedure to solve multimodal mixed variable optimization problems. The diversity enhancement procedure is embedded into $MV - MACO$ for two purposes: to enhance diversity and to increase search efficiency. Algorithm 6 exhibit the pseudo-code of the approach presented in section 4.2.

Algorithm 6 Pseudo-code for $MV - MACO$

```
Initialize Parameters;
Randomly Initialize the Solution Archive;
Partition the Solution Archive in to clusters;
while termination contiditon is not satisfied do
   Call Enhance Diversity
   for each cluster do
      Sort Solutions according to their fitness;
      Initialize Weights;
      for each ant construct ant Solution do
         Probabilistic solution construction for mixed-variable attributes;
      end for
      Sort solutions and select k best solutions;
   end for
end while
```

4.4 Flow Chart of the Proposed Methodology

This section explains the overall working of the algorithm $MV - MACO$ and explains its working in the form of flowchart 4.2.

.

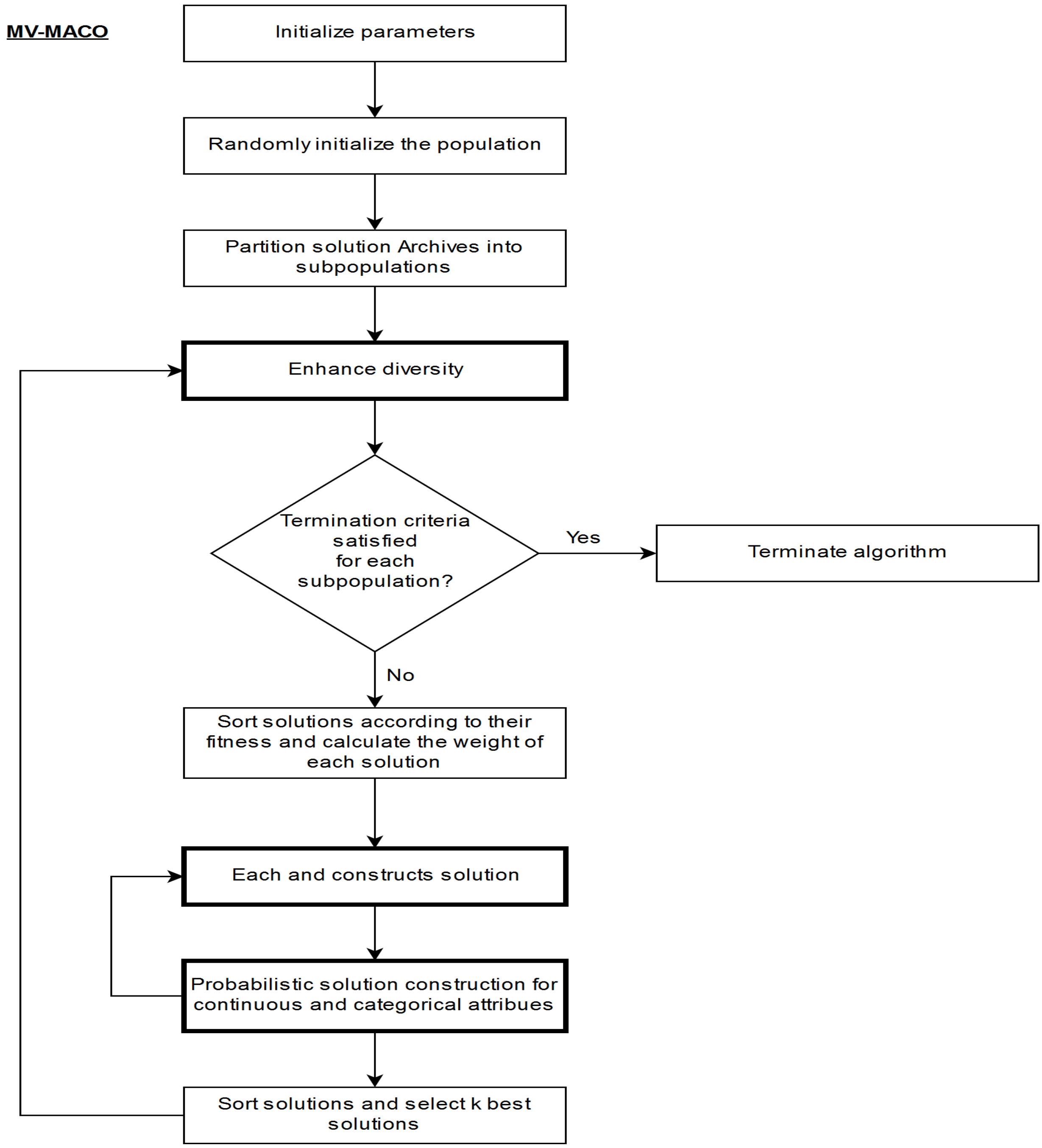

Figure 4.2: Flowchart of the Proposed Technique

Chapter 5

Results and Discussion

The validity of the designed approach is checked on different multi-modal benchmark problems and compared the results of $MV - MACO$ with already proposed algorithms in the literature. Furthermore, the effect of a different number of discrete points for the categorical attributes on the performance of the designed approach is also investigated in this section. For the validity of the proposed methodology, the experiments were conducted independently for each benchmark.

5.1 Selection of Test Problems

Our experiments were comprised of multi-modal benchmark problems found in the literature Lin et al. (2017); Choi and Ahn (2014); Wang et al. (2011). These problems used to investigate the performance of $MV - MACO$ mentioned in section 4.2 and these problems are defined in table 5.1.

Problem	Function	Dimensions
Ackleys Function	$f1$	6D, 10D, 20D
Griewangks Function	$f2$	6D, 10D, 20D
Rastrigins Function	$f3$	6D, 10D, 20D
Levy Function	$f4$	6D, 10D, 20D
Schwefel Function	$f5$	6D, 10D, 20D
Rastrigin Shift Function	$f6$	6D, 10D, 20D
Griewangks Shift Function	$f7$	6D, 10D, 20D
Ackley Shift Function	$f8$	6D, 10D, 20D
Schaffer Shift Function	$f9$	6D, 10D, 20D

Table 5.1: Multi-modal Benchmark Problems

5.2 Evaluation Criteria

5.2.1 Experimental Settings

For evaluating the designed approach following experimental settings were used as described in Lin et al. (2017).
Benchmark Problems: 9 minimization problems
Problem Dimensions: Results for dimensions 6, 10 and 20 were compiled for evaluation.
No of Runs: For each benchmark problem with 6, 10 and 20 dimensions 50 independent runs were conducted.
Maximum Fitness Evaluations: For each benchmark problem 10000 maximum fitness evaluations were allowed.
No of continuous and categorical attributes: Half of the dimensions of the problem were treated as continuous and other half as categorical for evaluation.
Problem Search Range: $[-100, 100]^D$
Population Initialization: Initilal population was uniformly randomly initialized inside the problem search space.
Algorithm Termination: Algorithm terminates when available number of fitness evaluations ends.

Parameters	Symbol	Value
Significance of best quality solution	q	0.06
Spread of the search space	ξ	0.68
Archive Size	S	500
Sub population size	M	25
Neighborhood radius	d_0	1
Convergence Threshold	ϵ	1.0E-01

Table 5.2: Parameters Settings for $MV - MACO$

5.2.2 Experimental Environment

All the experiments were performed on Windows 10 system with Intel i7 3630QM processor (2.4Ghz) 8GB RAM, and a Nvidia Geforce GT 630M and all the program codes were written in Python.

5.2.3 Parameter Settings

The parameter settings for the evaluation of the designed technique are shown in table 5.2. We used these parameters settings to analyse the effectiveness of $MV - MACO$ on multi-modal benchmark problems defined in table 5.1. Moreover, using these parameter settings the performance of $MV - MACO$ is evaluated on a different number of discrete points in the discretization step using these benchmark problems. Moreover, we used these parameter settings for the final validation of our designed $MV - MACO$ algorithm by comparing it with other variants of mixed variable optimizers and with recent algorithms, which were used in the literature for the testing of multimodal optimization algorithms.

5.3 Performance Measure

For evaluating the performance of the designed methodology, fifty independent runs were conducted for each problem having 6,10 and 20 dimensions respectively. Half of the continuous attributes were transformed into categorical variables, their continuous domains were discretized into a series of different discrete points. Each categorical variable was discretized in to 10,20,30,40,60,80,100 points respectively. Hence for the evaluation half of the variables were continuous and the other half were categorical. For the designed approach, we measured the mean value and standard deviation for the different settings of the no of discrete points and dimensions using 50 independent runs. Moreover, mean value, best value, worst value and standard deviation were recorded for the validation of the designed approach with respect to recent approaches in the literature. For each algorithm run the maximum 10000 function evaluations were allowed.

5.4 Results

Our designed algorithm was evaluated on 9 benchmark functions having 6, 10, 20 dimensions respectively, with a varying number of discrete points. This results in 27 group of experiments (9 benchmark functions and 3 dimensions). Furthermore, to the best of our knowledge the multimodal benchmark functions $f1$, $f2$ and $f3$ have only been used in the literature for the evaluation of the mixed variable optimizers in Lin et al. (2017). Hence we compared our designed approach $MV - MACO$ with other variants of the mixed variable optimizers $L - SHADE_{ACO}$ and ACO_{MV} with respect to the mentioned benchmark functions. Moreover, to further validate our approach, we compared it with recent algorithms for multimodal optimization, which were used for testing the multimodal optimizers in the literature.

5.4.1 Performance of MV-MACO on Various Number of Discrete Points

The experimental results of $MV - MACO$ with different number of discrete points on functions $f1$- $f9$, having dimensions 6,10 and 20 are reported in table 5.3.We measured the mean error value over the 50 runs of the designed approach and compute its standard deviation. These results are summarized in table 5.3. Furthermore, to make comparison easier, a graphical representation of the results reported in table 5.3 are shown in figure 5.1.

From table 5.3 it can be seen that $MV - MACO$ has reached the lower mean error value for all the functions and dimensions $D = 6$,10 and 20. The only exception is for the functions $f6$ on dimension 6,10 and 20 with 40 number of discrete points whereas, $MV - MACO$ has shown significant performance on function $f6$ as the number of discrete points and dimensions increases. For all the dimensions proposed algorithm has reached the most minimum error which indicates that it has successfully converged to the optima. Furthermore, for the function $f9$ on dimension 6 and 20 the performance of $MV - MACO$ remains almost consistent as the number of discrete points increase gradually but it has shown a remarkable decrease in error as the number of discrete points reached 100 on a 10 dimensional Schaffer Shift Function. In fact, the observed differences are also significant in figure 5.1. However, it is clear that $MV - MACO$ has performed much better on higher dimensional problems with a large number of discrete points. This may be because of the more explorative behaviour of the $MV - MACO$ due to the diversity enhancement operation which helps to avoid unproductive search efforts and efficiently utilizes available fitness evaluations.

Table 5.3: Experimental Results(Mean and Std Dev) of MV-MACO on Different Number of Discrete Points for Functions $f1$-$f9$ Having Dimensions 6,10 and 20

		Mean(Std Dev)		
		Dimensions		
Func	**No of Discrete Points**	**6**	**10**	**20**
$f1$	**10**	1.74E-14±1.58E-14	1.19E-07±2.13E-08	2.43E-04±5.78E-05
	20	6.34E-14±8.02E-15	1.64E-13±1.25E-14	2.64E-13±8.85E-15
	30	5.54E-14±4.35E-15	1.45E-13±6.36E-15	2.89E-13±4.63E-15
	40	9.22E-14±6.22E-15	2.25E-13±1.05E-14	4.29E-13±8.18E-15
	60	3.11E-14±3.16E-15	2.42E-12±3.89E-01	8.88E+00±6.11E-01
	80	5.58E-14±5.71E-15	1.45E-13±6.26E-15	2.79E-13±6.36E-15
	100	1.41E-13±6.65E-13	2.59E-13±6.71E-15	4.301E-13±9.858E-15
$f2$	**10**	1.11E-16±0.00E+00	1.78E-12±7.29E-13	8.73E-06±4.42E-06
	20	4.93E-04±2.20E-03	1.47E-13±5.83E-13	4.31E-15±4.47E-16
	30	4.93E-04±2.20E-03	1.25E-15±2.07E-16	3.31E-15±1.65E-16
	40	5.32E-15±1.37E-15	4.93E-04±2.20E-03	7.22E-15±1.15E-15
	60	1.36E-03±3.30E-03	2.82E-02±6.85E-02	2.67E-01±4.34E-01
	80	5.38E-16±2.53E-16	1.20E-15±2.34E-16	3.46E-15±3.08E-16
	100	6.78E-04±2.28E-03	8.66E-16±8.76E-17	3.175E-15±6.081E-17
$f3$	**10**	1.53E-03±3.31E-07	5.00E-03±7.38E+01	1.10E-04±1.11E-03
	20	3.28E-02±4.91E-02	3.31E+00±2.15E+00	8.86E-04±3.95E-03
	30	3.55E-09±7.77E-09	5.69E-03±6.51E-02	1.05E-03±6.00E-03
	40	4.41E-08±8.61E-09	1.02E-08±7.40E-09	2.02E-05±1.04E-05
	60	2.66E-11±1.69E-11	3.31E-09±8.22E-09	2.64E-11±3.01E-10
	80	2.02E-13±1.67E-13	5.69E-13±5.67E-13	3.56E-12±2.01E-12
	100	2.05E-13±1.03E-13	4.09E-12±6.43E-13	1.56E-11±2.07E-12
$f4$	**10**	5.98E-10±1.76E-09	2.09E-11±3.91E-11	1.11E-09±2.47E-09
	20	3.58E-31±6.87E-31	4.03E-31±6.28E-31	3.44E-31±2.83E-31
	30	6.36E-31±6.39E-31	1.22E-30±1.64E-30	2.72E-30±2.83E-31
	40	3.83E-29±5.67E-30	2.93E-31±7.07E-31	2.72E-30±1.97E-30
	60	2.61E-28±1.46E-27	2.02E-28±3.11E-28	1.77E-31±1.50E-31
	80	1.19E-27±2.44E-26	5.83E-26±1.27E-26	2.61E-28±1.46E-27
	100	2.56E-26±1.03E-27	6.52E-26±1.39E-27	1.33E-13±2.37E-12
$f5$	**10**	3.37E-10±1.79E-11	1.57E-09±3.07E-09	1.19E-12±2.16E-12
	20	2.62E-09±1.63E-09	1.13E-09±1.17E-09	1.64E-13±1.25E-14
	30	2.55E-10±2.04E-09	7.43E-10±6.36E-09	1.45E-13±6.36E-15
	40	3.31E-11±1.67E-11	5.25E-10±3.42E-10	2.25E-13±1.05E-14

Table 5.3 continued from previous page

		Mean(Std Dev)		
		Dimensions		
	60	2.66E-11±1.89E-11	7.62E-09±6.89E-10	2.42E-13±3.89E-12
	80	3.51E-10±6.53E-10	1.07E-10±2.01E-10	1.43E-13±6.26E-15
	100	4.03E-10±1.56E-10	1.02E-09±1.65E-11	2.59E-13±6.71E-15
$f6$	**10**	1.08E+02±1.67E-10	1.33E+02±3.42E-12	1.10E+02±3.67E-13
	20	1.23E+02±2.32E-10	1.52E+02±4.41E-12	1.37E+02±4.80E-14
	30	1.67E+02±2.39E-11	1.17E+02±1.93E-13	1.54E+02±5.10E-14
	40	1.91E+02±1.98E-13	2.01E+02±5.32E-12	3.09E+02±4.07E-14
	60	-2.19E+02±1.74E-13	-3.29E+02±4.89E-13	-3.36E+02±3.38E-13
	80	-3.25E+02±1.23E-13	-3.96E+02±5.74E-14	-3.07E+02±4.32E-13
	100	-3.13E+02±1.61E-13	-3.53E+02±6.83E-14	-3.53E+02±5.68E-14
$f7$	**10**	-1.87E+02±5.84E-14	-1.82E+02±1.18E–09	-1.75E+02±1.23E-09
	20	-1.83E+02±4.34E-09	-1.80E+02±2.28E-09	-1.75E+02±1.78E-09
	30	-1.83E+02±4.38E-09	-1.83E+02±3.52E-08	-1.78E+02±2.36E-10
	40	-1.84E+02±2.75E-03	-1.81E+02±2.89E-10	-1.78E+02±3.30E-13
	60	-1.84E+02±2.39E-02	-1.81E+02±3.45E-14	-1.79E+02±4.03E-14
	80	-1.83E+02±3.14E-03	-1.81E+02±3.52E-14	-1.79E+02±3.07E-14
	100	-2.83E+02±3.87E-03	-1.83E+02±3.42E-14	-1.79E+02±4.01E-14
$f8$	**10**	-1.43E+02±4.47E-14	-1.43E+02±2.18E-08	-1.43E+02±7.53E-11
	20	-1.59E+02±5.09E-14	-1.49E+02±5.98E-11	-1.44E+02±1.21E-11
	30	-1.57E+02±2.07E-14	-1.48E+02±3.50E-11	-1.45E+02±4.08E-10
	40	-1.57E+02±3.80E-14	-1.48E+02±7.94E-12	-1.41E+02±5.68E-09
	60	-1.53E+02±1.02E-14	-1.49E+02±2.34E-12	-1.44E+02±3.24E-09
	80	-1.51E+02±6.13E-14	-1.47E+02±3.42E-12	-1.41E+02±8.76E-09
	100	-1.48E+02±7.24E-14	-1.47E+02±4.92E-12	-1.44E+02±4.49E-09
$f9$	**10**	9.77E-01±2.44E-02	4.71E+00±3.53E+00	1.48E+01±5.04E+00
	20	5.74E+00±1.18E+01	5.89E+00±8.40E+00	5.40E+01±3.73E+00
	30	6.08E+01±6.83E+00	6.05E+00±1.50E+02	6.33E+01±1.66E+02
	40	6.43E+01±4.99E+01	5.62E+01±7.47E+01	6.60E+01±8.05E+00
	60	5.700E+01±8.03E+00	6.00E+01±1.89E+01	6.14E+01±2.33E+02
	80	5.64E+01±1.23E+01	5.54E+01±3.81E+01	5.71E+01±8.02E+01
	100	6.78E+01±1.23E+01	6.25E+00±2.04E+01	5.34E+01±1.47E+01

5.4.2 Comparison of Solution Quality with Existing Variants of ACO

In the existing literature, there exists two variants of ACO, $L - SHADE_{ACO}$ and ACO_R for solving mixed variable optimization problems. Hence, in this section, we compared the results of $MV - MACO$ at dimensions 6 and 10 with varied discrete points for each categorical variable. In Lin et al. (2017)these number of points is taken as 100 and results were reported for $L - SHADE_{ACO}$ and ACO_R on benchmark functions $f1$, $f2$ and $f3$. As stated before, the mean error value is calculated over the 50 independent runs of the algorithms for dimensions 6 and 10. For algorithms $L - SHADE_{ACO}$ and ACO_R the values below 0.00E-10 are approximated to 0.00E+00 in the existing literature.

From table 5.4, it can be seen that $MV - MACO$ has found the optimal value for dimension equal to 6 for the functions $f1$, $f2$ and $f3$. It can also be seen that the same level of performance is also achieved by $L - SHADE_{ACO}$ for the functions $f2$ and $f3$. Similarly, the same level of performace is also achived by ACO_R for the the functions $f1$ and $f2$. The results of the functions $f1$, $f2$ and $f3$ for the higher dimension 10 are recorded in table 5.5. It can be seen that $MV - MACO$ has achieved the lowest mean error value for all the functions as compared to $L - SHADE_{ACO}$ and ACO_R. $L - SHADE_{ACO}$ has achieved better mean value for the function $f1$ and $f3$ whereas for functions $f2$, ACO_R has performed better. From the results, we can observe that $MV - MACO$ appears to be more robust and effective when applied to multimodal mixed variable optimization problems from the fact that it achieves the lowest mean error value among all algorithms mentioned previously.

Table 5.4: Experimental Results Of MV-MACO, $L - SHADE_{ACO}$ and ACO_R On Benchmarks f1, f2 and f3 with Dimension=6

Algorithm	**Measure**	$f1$	$f2$	$f3$
$L - SHADE_{ACO}$	Mean	7.15E-01	0.00E+00	0.00E+00
ACO_R	Mean	0.00E+00	8.41E-04	2.10E+00
$MV - MACO$	Mean	1.66E-13	7.40E-04	2.61E-13

Table 5.5: Experimental Results Of MV-MACO,$L - SHADE_{ACO}$ and ACO_R On Benchmarks f1, f2 and f3 with Dimension=10

Algorithm	**Measure**	$f1$	$f2$	$f3$
$L - SHADE_{ACO}$	Mean	1.17E+00	5.10E-01	5.07E-03
ACO_R	Mean	0.00E+00	4.52E-03	1.03E+01
$MV - MACO$	Mean	2.59E-13	8.66E-16	4.09E-12

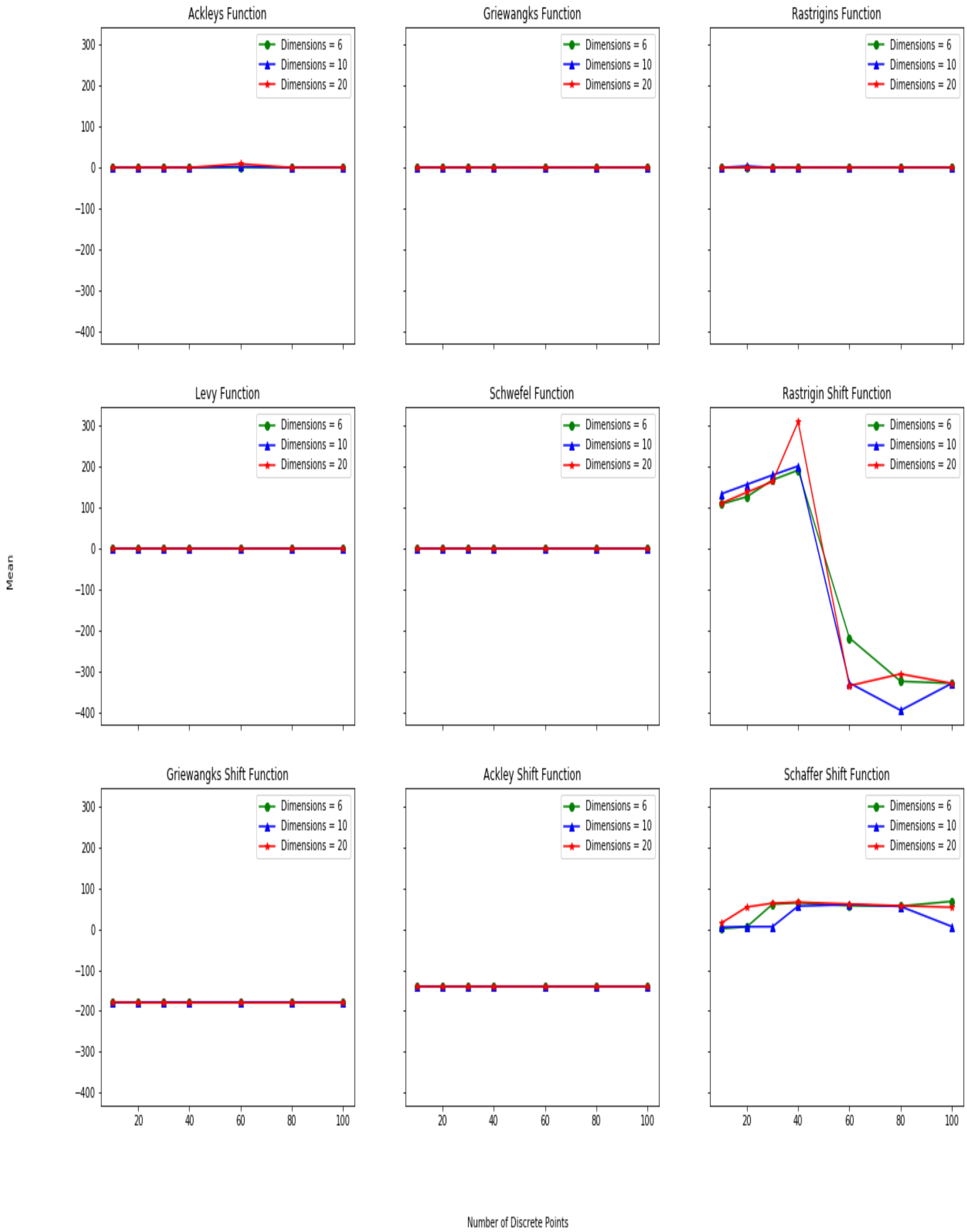

Figure 5.1: Graphs Representing of the Experimental Results of MV-MACO on Different Number of Discrete Points for Functions $f1$-$f9$ Having Dimensions 6,10 and 20

5.4.3 Comparison with Recently Proposed Algorithms

To evaluate our designed methodology we compared our algorithm $MV - MACO$ with three recently proposed algorithms $EPUS - PSO$,$CoDE$ and $jDEdynNP - F$ over 4 performance metrics based on error Fieldsend (2014); Qu et al. (2013); Qin et al. (2009). These performance metrics include best, mean worst and the standard deviation of the error.

Table 5.6 shows the results obtained by the algorithms $EPUS - PSO$,$CoDE$ and $jDEdynNP - F$ on 6-dimensional problems and a detailed comparison of the performance of these algorithms using best, mean, worst and standard deviation of the error with our designed $MV - MACO$ algorithm is presented. From the metrics we can judge that $MV - MACO$ has performed better on $f1$,$f3$,$f4$,$f5$ and $f8$ as compared to $EPUS - PSO$,$CoDE$ and $jDEdynNP - F$. For $f1$ the performance of $MV - MACO$ and $CoDE$ is relatively similar as shown by the mean and best error values of both the algorithms. Moreover, for function $f6$ $jDEdynNP - F$ outperforms $MV - MACO$. Similarly, for the function $f7$ $EPUS - PSO$ has better performance with respect to its mean and standard deviation as compared to other algorithms. For a function, $f9$ both $EPUS - PSO$ and $MV - MACO$ didn't achieve the satisfactory results whereas, $CoDE$ and $jDEdynNP - F$ are somewhat near the optimal solution.

Furthermore, table 5.7 shows the results obtained by $MV - MACO$,$EPUS - PSO$,$CoDE$ and $jDEdynNP - F$ on 10 dimensional problems. As the dimensions of the problem increase gradually the complexity of the problem also increases which negatively affects the performance of the population-based algorithms. $MV - MACO$ outperforms on the functions $f1$, $f3$,$f5$, $f6$ and $f9$ as compared to $EPUS - PSO$,$CoDE$ and $jDEdynNP - F$ algorithms in the literature. For function, $f2$ and $f9$, $MV - MACO$,$EPUS - PSO$ and $CoDE$ produced almost similar results with respect to mean, best and standard deviation error values. Similarly, for the function $f7$ the performance of all the algorithms is almost identical with respect to their mean error values. Furthermore, for the function $f9$ all the algorithms have achieved almost the lowest error values for all the metrics hence, all have converged approximately close to the optima.

Moreover, the performance of all the mentioned algorithms is also evaluated on higher dimensional problems having dimensions equal to 20. All the mentioned algorithms in the table 5.8 have performed really well on higher dimensional problems. For functions $f1$ and $f5$ $MV - MACO$ and $EPUS - PSO$ have achieved almost similar results but, $EPUS - PSO$ supersede $MV - MACO$ with a slight margin whereas, in the case of the function $f2$ $MV - MACO$ takes the lead. Furthermore, $MV - MACO$ outperforms for the functions $f3$, $f4$ and $f6$ by having the lowest possible error as compared to $EPUS - PSO$,$CoDE$ and $jDEdynNP - F$ which clearly states that $MV - MACO$ is more robust and effective on higher dimensional problems because of its more explorative behaviour due to embedded diversity enhancement operation which helps to avoid premature convergence and helps to escape the population from trapping in to local optima. For functions, $f7$ and $f8$ the performance of all the algorithms is almost identical with respect to their mean error values. Moreover, for the function $f9$, $jDEdynNP - F$ performed better than all the algorithms mentioned. From the results it can be observed that $jDEdynNP - F$ is able to perform better on higher dimensional problems as compared to $MV - MACO$,$EPUS - PSO$ and $CoDE$ considering the function $f9$ to be highly multimodal and complex.

From tables 5.6,5.7 and 5.8 we can judge that $MV - MACO$ outperforms the mentioned state of the art algorithms under most conditions. The results on the algorithms were generated using 6, 10 and 20 dimensions and error values were reported using 50 independent runs while $MV - MACO$ was also evaluated with a number of discrete points of each categorical variable equal to 100 since it is the only algorithm that can handle discrete variables than those mentioned in this study.

Table 5.6: Comparison of MV-MACO with Recently Proposed Algorithms for the Dimension=6

Algorithm	Function	Best	Mean	Worst	SD
$MV-MACO$	$f1$	*1.66E-13*	*1.41E-13*	*1.73E-13*	*6.65E-13*
$EPUS-PSO$		1.37E+01	1.80E+01	1.81E+01	1.58E-01
$CoDE$		1.17E-13	2.61E-13	7.92E-13	1.63E-13
$jDEdynNP-F$		4.02E+00	5.80E+00	6.91E+00	8.52E-01
$MV-MACO$	$f2$	5.64E-04	6.78E-04	7.51E-04	2.28E-03
$EPUS-PSO$		*1.13E-15*	*1.34E-04*	*1.03E-01*	*3.23E-02*
$CoDE$		0.00E+00	7.56E-04	1.70E-03	3.75E-03
$jDEdynNP-F$		0.00E+00	2.04E-01	7.58E-01	1.01E-01
$MV-MACO$	$f3$	*2.61E-13*	*2.05E-13*	*3.01E-13*	*1.03E-13*
$EPUS-PSO$		0.97E+02	1.72E+03	4.21E+03	2.09E+02
$CoDE$		3.74E+02	1.24E+02	5.31E+02	3.64E+01
$jDEdynNP-F$		1.98E+01	3.05E+01	5.77E+01	7.03E+00
$MV-MACO$	$f4$	*2.42E-26*	*2.56E-26*	*2.99E-26*	*1.03E-26*
$EPUS-PSO$		3.45E+02	3.84E+02	4.47E+02	2.90E+01
$CoDE$		3.58E01	1.73E+00	3.62E+00	8.78E-01
$jDEdynNP-F$		5.29E+00	9.41E+00	1.46E+01	2.53E+00
$MV-MACO$	$f5$	*3.47E-10*	*4.03E-10*	*4.78E-10*	*1.56E-10*
$EPUS-PSO$		3.24E+03	3.48E+03	3.91E+03	1.71E+03
$CoDE$		9.58E+01	1.25E+02	1.51E+02	1.03E+02
$jDEdynNP-F$		1.72E+03	2.01E+03	3.54E+03	6.15E+02
$MV-MACO$	$f6$	-4.01E+02	-3.13E+02	-3.02E+02	1.61E+01
$EPUS-PSO$		1.75E+03	2.23E+03	3.14E+03	2.51E+02
$CoDE$		5.17E+01	8.89E+01	1.06E+02	2.20E+01
$jDEdynNP-F$		*-8.42E+01*	*-1.40E+02*	*-2.00E+02*	*2.20E+01*
$MV-MACO$	$f7$	-3.01E+02	-2.83E+02	-0.83E+02	3.87E-03
$EPUS-PSO$		*-3.10E+02*	*-1.99E+02*	*-1.81E+02*	*6.95E-13*
$CoDE$		-1.84E+02	-1.81E+02	-1.80E+02	1.23E-14
$jDEdynNP-F$		-1.81E+02	-1.80E+02	-1.80E+02	4.14E-01
$MV-MACO$	$f8$	*-1.79E+02*	*-1.48E+02*	*-1.31E+02*	*7.24E-14*
$EPUS-PSO$		-1.21E+02	-1.21E+02	-1.21E+02	3.33E-02
$CoDE$		-1.40E+02	-1.40E+02	-1.40E+02	9.05E-13
$jDEdynNP-F$		-1.37E+02	-1.36E+02	-1.34E+02	8.27E-01
$MV-MACO$	$f9$	9.56E+01	8.78E+01	9.89E+01	1.23E+01
$EPUS-PSO$		1.72E+03	1.82E+03	1.88E+03	3.59E+01
$CoDE$		*6.51E+01*	*9.58E+01*	*1.51E+02*	*2.52E+01*
$jDEdynNP-F$		2.30E+02	3.01E+02	3.60E+02	3.65E+01

Table 5.7: Comparison of MV-MACO with Recently Proposed Algorithms for the Dimension=10

Algorithm	Function	Best	Mean	Worst	SD
$MV-MACO$	$f1$	*2.53E-13*	*2.59E-13*	*2.71E-13*	*6.71E-15*
$EPUS-PSO$		188E+01	1.91E+01	1.93E+01	8.80E-02
$CoDE$		2.15E+00	2.29E+00	3.25E+00	3.32E-01
$jDEdynNP-F$		7.99E+00	9.55E+00	1.33E+01	3.19E-01
$MV-MACO$	$f2$	7.11E-16	8.66E-16	9.09E-16	8.76E-17
$EPUS-PSO$		*6.61E-16*	*7.14E-16*	*8.78E-16*	*1.12E-16*
$CoDE$		1.01E-16	3.27E-02	1.77E-01	6.24E-02
$jDEdynNP-F$		1.04E-011	1.25E-01	4.32E-01	1.42E-01
$MV-MACO$	$f3$	*2.73E-012*	*4.09E-12*	*4.55E-12*	*6.43E-13*
$EPUS-PSO$		2.20E+03	2.74E+03	3.13E+03	1.67E+02
$CoDE$		5.27E+02	6.12E+02	7.08E+02	8.51E+01
$jDEdynNP-F$		2.12E+02	2.73E+02	3.95E+02	2.08E+01
$MV-MACO$	$f4$	*6.31E-26*	*6.52E-26*	*6.86E-26*	*1.39E-27*
$EPUS-PSO$		9.28E+02	9.76E+02	1.05E+03	4.12E+01
$CoDE$		1.24E+01	1.54E+01	2.03E+01	3.04E+00
$jDEdynNP-F$		1.87E+01	3.19E+01	4.34E+01	7.38E+00
$MV-MACO$	$f5$	*9.80E-10*	*1.02E-09*	*1.75E-09*	*1.65E-11*
$EPUS-PSO$		7.84E+04	8.45E+04	9.12E+04	2.86E+03
$CoDE$		1.13E+04	1.44E+04	2.41E+04	3.35E+03
$jDEdynNP-F$		2.03E+04	2.42E+04	2.78E+04	2.72E+03
$MV-MACO$	$f6$	*-3.86E+02*	*-3.53E+02*	*-3.30E+02*	*-1.61E-13*
$EPUS-PSO$		3.34E+03	4.01E+03	4.82E+03	4.41E+02
$CoDE$		4.08E+02	5.33E+02	7.11E+02	8.86E+01
$jDEdynNP-F$		8.42E+02	1.02E+03	1.39E+03	1.79E+02
$MV-MACO$	$f7$	*-2.09E+02*	*-1.83E+02*	*-1.73E+02*	*3.42E-14*
$EPUS-PSO$		-1.80E+02	-1.80E+02	-1.80E+02	2.34E-03
$CoDE$		-1.80E+02	-1.80E+02	-1.80E+02	3.32E-02
$jDEdynNP-F$		-1.80E+02	-1.80E+02	-1.80E+02	1.34E-01
$MV-MACO$	$f8$	*-1.54E+02*	*-1.47E+02*	*-1.41E+02*	*4.92E-14*
$EPUS-PSO$		-1.21E+02	-1.21E+02	-1.20E+02	1.27E-02
$CoDE$		-1.38E+02	-1.38E+02	-1.35E+02	2.23E-01
$jDEdynNP-F$		-1.32E+02	-1.30E+02	-1.28E+02	4.12E-01
$MV-MACO$	$f9$	*2.12E+02*	*2.25E+02*	*2.59E+02*	*2.14E+01*
$EPUS-PSO$		4.27E+03	4.57E+03	4.74E+03	7.81E+01
$CoDE$		4.66E+02	4.96E+02	5.26E+02	2.67E+01
$jDEdynNP-F$		6.01E+02	7.50E+02	8.08E+02	1.22E+02

Table 5.8: Comparison of MV-MACO with Recently Proposed Algorithms for the Dimension=20

Algorithm	**Function**	**Best**	**Mean**	**Worst**	**SD**
$MV-MACO$	$f1$	4.166E-013	4.301E-13	4.414E-15	9.858E-15
$EPUS-PSO$		*1.88E-13*	*1.88E-13*	*2.01E-13*	*6.42E-13*
$CoDE$		6.79E-09	7.17E-09	7.59E-09	3.18E-08
$jDEdynNP-F$		1.25E-03	1.29E-03	1.28E-03	2.13E-04
$MV-MACO$	$f2$	3.10E-15	3.17E-15	3.22E-15	6.08E-17
$EPUS-PSO$		*2.33E-15*	*2.46E-15*	*2.55E-15*	*9.28E-17*
$CoDE$		1.80E-05	2.26E-01	1.02E+00	4.47E-01
$jDEdynNP-F$		1.34E-05	2.19E-01	6.67E-01	2.71E-01
$MV-MACO$	$f3$	*1.27E-01*	*1.56E-11*	*1.81E-11*	*2.07E-12*
$EPUS-PSO$		4.63E+03	4.76E+03	4.93E+03	1.07E+02
$CoDE$		8.93E+02	9.27E+02	9.64E+02	2.88E+01
$jDEdynNP-F$		6.30E+02	6.78E+02	7.33E+02	4.06E+01
$MV-MACO$	$f4$	*1.30E-25*	*1.33E-25*	*1.36E-25*	*1.27E-27*
$EPUS-PSO$		1.77E+03	1.83E+03	1.92E+03	5.68E+01
$CoDE$		4.31E+01	5.17E+01	6.23E+01	7.84E+00
$jDEdynNP-F$		8.06E+01	8.98E+01	1.08E+02	1.07E+01
$MV-MACO$	$f5$	2.32E-09	2.59E-09	2.38E-08	6.71E-15
$EPUS-PSO$		*1.51E-11*	*1.60E-11*	*1.73E-11*	*5.06E-11*
$CoDE$		5.52E+04	5.75E+04	5.95E+04	2.03E+03
$jDEdynNP-F$		8.81E+04	9.15E+04	9.44E+04	3.14E+03
$MV-MACO$	$f6$	*-3.30E+02*	*-3.31E+02*	*-3.30E+02*	*5.68E-14*
$EPUS-PSO$		5.61E+03	6.17E+03	6.73E+03	3.84E+02
$CoDE$		1.66E+03	1.84E+03	2.06E+03	1.17E+02
$jDEdynNP-F$		4.20E+03	4.52E+03	5.05E+03	2.72E+02
$MV-MACO$	$f7$	*-1.97E+02*	*-1.81E+02*	*-1.18E+02*	*4.01E-14*
$EPUS-PSO$		-1.19E+02	-1.18E+02	-1.17E+02	2.32E-03
$CoDE$		-1.81E+02	-1.75E+02	-1.65E+02	5.59E+00
$jDEdynNP-F$		-1.81E+02	-1.79E+02	-1.79E+02	2.23E-01
$MV-MACO$	$f8$	*-1.40E+02*	*-1.37E+02*	*-1.39E+02*	*4.49E-09*
$EPUS-PSO$		-1.21E+02	-1.21E+02	-1.21E+02	1.13E-02
$CoDE$		-1.33E+02	-1.31E+02	-1.30E+02	9.07E-01
$jDEdynNP-F$		-1.28E+02	-1.28E+02	-1.28E+02	3.36E-01
$MV-MACO$	$f9$	3.08E+02	3.34E+02	3.49E+02	1.47E+01
$EPUS-PSO$		9.19E+03	9.27E+03	9.31E+03	8.61E+01
$CoDE$		4.02E+03	4.32E+03	4.51E+03	1.42E+02
$jDEdynNP-F$		*4.06E+01*	*4.69E+01*	*5.12E+01*	*3.88E+01*

Chapter 6

Conclusion

6.1 Conclusion

For solving real-world problems, decision makers prefer to have multiple solutions to a problem before the final decision can be made. This is important because if one solution is not applicable, then the other solution can be adopted immediately. However, these real-world problems are modelled using mixed decision variables. Existing algorithms mostly focus on solving continuous or discrete optimization problems, rather than solving mixed variable optimization problems. This paper has focused on solving multi-modal mixed-variable optimization problems using ant colony optimization algorithm. The solution construction step of the ants in ACO and the structure of the solution archive is modified to construct solutions for mixed decision variables. Furthermore, to achieve multiple optimal solutions clustering method is used for distributing search space in to different clusters, which ultimately increases the chances of finding global optimal solutions respectively. Furthermore, in order to have the robust and efficient performance of $MV - MACO$ on higher dimensional problems a diversity enhancement procedure is incorporated into $MV - MACO$ which has eventually enhanced it's explorative behaviour and thus, helps to escape the population into local optima. Moreover, a detailed analysis is performed for the validity of the proposed approach and compared the results with already proposed algorithms in the literature.

The $MV - MACO$ solution archive gives a flexible and adaptive framework for solving constrained multimodal optimization problems. Therefore we are passionate to extend our work by developing and integrating an efficient constraint handling mechanism into $MV - MACO$ and to further boost the performance of $MV - MACO$ the local search strategy will also be incorporated which will eventually help to achieve better solutions in less number of function evaluations.

References

Basak, A., S. Das, and K. C. Tan (2013). Multimodal optimization using a biobjective differential evolution algorithm enhanced with mean distance-based selection. *IEEE Trans. Evol. Comput. 17*(5), 666–685.

Chen, Z., S. Zhou, and J. Luo (2017). A robust ant colony optimization for continuous functions. *Expert Syst. Appl. 81*(April), 309–320.

Cheng, R., M. Li, K. Li, and X. Yao (2017). Evolutionary Multiobjective Optimization Based Multimodal Optimization: Fitness Landscape Approximation and Peak Detection. *IEEE Trans. Evol. Comput. XX*(X), 1–15.

Choi, T. J. and C. W. Ahn (2014). An adaptive cauchy differential evolution algorithm with bias strategy adaptation mechanism for global numerical optimization. *JCP 9*(9), 2139–2145.

Ciornei, I. and E. Kyriakides (2012). Hybrid ant colony-genetic algorithm (GAAPI) for global continuous optimization. *IEEE Trans. Syst. Man, Cybern. Part B Cybern. 42*(1), 234–245.

Dasa, S., S. Maity, B. Y. Qu, and P. N. Suganthan (2011). Real-parameter evolutionary multimodal optimization-A survey of the state-of-the-art. *Swarm Evol. Comput. 1*(2), 71–78.

Fieldsend, J. E. (2014). Running Up Those Hills: Multi-modal search with the niching migratory multi-swarm optimiser. *Proc. 2014 IEEE Congr. Evol. Comput. CEC 2014*, 2593–2600.

Gao, W., G. G. Yen, and S. Liu (2014). A cluster-based differential evolution with self-adaptive strategy for multimodal optimization. *IEEE Trans. Cybern. 44*(8), 1314–1327.

Gong, M., L. Pan, T. Song, K. Tang, and X. Zhang (2015). Ant Colony Optimization with Different Crossover Schemes for Continuous Optimization. *Bio-inspired computing - theories and applications: 10th international conference, bic-ta 2015 hefei, China, september 25-28, 2015 proceedings 562*, 56–62.

Li, L. and K. Tang (2015). History-based topological speciation for multimodal optimization. *IEEE Trans. Evol. Comput. 19*(1), 136–150.

Li, X., M. G. Epitropakis, K. Deb, and A. Engelbrecht (2017, August). Seeking multiple solutions: An updated survey on niching methods and their applications. *IEEE Transactions on Evolutionary Computation 21*(4), 518–538.

Liao, T., M. A. Montes de Oca, D. Aydin, T. Stützle, and M. Dorigo (2011). An incremental ant colony algorithm with local search for continuous optimization. *Proc. 13th Annu. Conf. Genet. Evol. Comput. - GECCO '11*, 125.

Liao, T., K. Socha, M. A. M. de Oca, T. Stützle, and M. Dorigo (2014). Ant colony optimization for mixed-variable optimization problems. *IEEE Trans. Evolutionary Computation 18*(4), 503–518.

Liao, T., K. Socha, M. A. M. D. Oca, and T. St (2014). Ant Colony Optimization for Mixed-Variable Optimization Problems. *18*(4), 503–518.

Liao, T., T. Stützle, M. A. Montes De Oca, and M. Dorigo (2014). A unified ant colony optimization algorithm for continuous optimization. *Eur. J. Oper. Res. 234*(3), 597–609.

Liao, T. W., R. J. Kuo, and J. T. Hu (2012). Hybrid ant colony optimization algorithms for mixed discrete-continuous optimization problems. *Appl. Math. Comput. 219*(6), 3241–3252.

Lin, Y., W. Du, and T. Stützle (2017). Three L-SHADE based algorithms on mixed-variables optimization problems. In *2017 IEEE Congress on Evolutionary Computation, CEC 2017, Donostia, San Sebastián, Spain, June 5-8, 2017*, pp. 2274–2281.

Lin, Y. C., M. Clauß, and M. Middendorf (2016). Simple Probabilistic Population-Based Optimization. *IEEE Trans. Evol. Comput. 20*(2), 245–262.

Qin, A. K., V. L. Huang, and P. N. Suganthan (2009). Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evolutionary Computation 13*(2), 398–417.

Qu, B. Y., P. N. Suganthan, and S. Das (2013). A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Trans. Evol. Comput. 17*(3), 387–402.

Qu, B. Y., P. N. Suganthan, and J. J. Liang (2012). Differential evolution with neighborhood mutation for multimodal optimization. *IEEE Trans. Evol. Comput. 16*(5), 601–614.

Socha, K. and M. Dorigo (2008). Ant colony optimization for continuous domains. *European journal of operational research 185*(3), 1155–1173.

STUTZLEz, T., M. DORIGO, IRIDIA, U. L. de Bruxelles, Belgium, ftstutzle, and mdorigog@ulb.ac.be (1999). 1 aco algorithms for the traveling salesman problemy.

Wang, Y., Z. Cai, and Q. Zhang (2011). Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evolutionary Computation 15*(1), 55–66.

Wang, Y., H. X. Li, G. G. Yen, and W. Song (2015). MOMMOP: Multiobjective Optimization for Locating Multiple Optimal Solutions of Multimodal Optimization Problems. *IEEE Trans. Cybern. 45*(4), 830–843.

Xiao, J. and L. Li (2011). A hybrid ant colony optimization for continuous domains. *Expert Syst. Appl. 38*(9), 11072–11077.

Xiao-Min, H., Z. Jun, H. S. H. Chung, L. Yun, and L. Ou (2010). SamACO: Variable Sampling Ant Colony Optimization Algorithm for Continuous Optimization. *Syst. Man, Cybern. Part B Cybern. IEEE Trans. 40*(6), 1555–1566.

Yang, Q., W. N. Chen, Z. Yu, T. Gu, Y. Li, H. Zhang, and J. Zhang (2017, April). Adaptive multimodal continuous ant colony optimization. *IEEE Transactions on Evolutionary Computation 21*(2), 191–205.

Zhang, Y., Y. Gong, H. Zhang, T. Gu, and J. Zhang (2017). Toward fast niching evolutionary algorithms: A locality sensitive hashing-based approach. *IEEE Trans. Evolutionary Computation 21*(3), 347–362.