University of Windsor
Scholarship at UWindsor

Electronic Theses and Dissertations

12-9-2015

DOMINANCE IN MULTI-POPULATION CULTURAL ALGORITHMS

Santosh Upadhyayula
University of Windsor

Recommended Citation

Upadhyayula, Santosh, "DOMINANCE IN MULTI-POPULATION CULTURAL ALGORITHMS" (2015). *Electronic Theses and Dissertations.* Paper 5674.

DOMINANCE IN MULTI-POPULATION CULTURAL ALGORITHMS

By

SANTOSH UPADHYAYULA

A Thesis

Submitted to the Faculty of Graduate Studies

Through the School of Computer Science

In Partial Fulfillment of the Requirements for

the Degree of Master of Science at the

University of Windsor

Windsor, Ontario, Canada

2015

Dominance in Multi-Population Cultural Algorithms

by

Santosh Upadhyayula

APPROVED BY:

Kemal Tepe, External Reader
Department of Electrical and Computer Engineering

Dan Wu, Internal Reader
School of Computer Science

Ziad Kobti, Advisor
School of Computer Science

November 20, 2015

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

We propose a new approach that can be used for solving the knowledge migration issue in multi-population cultural algorithms (MPCA). In this study we introduce a new method to enable the migration of individuals from one population to another using the concept of complete dominance applied to MPCA. The MPCA's artificial population comprises of agents that belong to a certain sub-population. In this work we create a dominance multi population cultural algorithm (D-MPCA) with a network of populations that implements a dominance strategy. We hypothesize that the evolutionary advantage of dominance can help improve the performance of MPCA in general optimization problems. Three benchmark optimization functions are used to calculate the fitness value of the individuals. The proposed D-MPCA showed improved performance over the traditional MPCA. We conclude that dominance helps in improving the efficiency of knowledge migration in MPCA.

DEDICATION

To my loving family:

Father: U L N Chainulu

Mother: U Aruna Kumari

Brother: Upadhyayula Naresh

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Ziad Kobti, for giving me the opportunity to get an exposure of the research field. It has been a great experience working under his guidance. His constant motivation, support and faith guided me in successful completion of my thesis. I would also like to appreciate him for all the financial support he has given, which helped me takecare of all my expenses and focus on my research.

I would like to thank my internal reader Dr. Dan Wu for his support and positive attitude towards my approach. I am also thankful to my external reader Dr. Kemal Tepe for his suggestions and interest in the approach.

I give my sincere thanks to Mrs. Gloria Mensah, the secretary to the director and Mrs. Karen Bourdeau, the graduate secretary, who always supported and helped me when ever I needed any assistance in various academic issues and in setting up last minute meetings.

Lastly, I would like to thank my family and friends for their utmost faith in me and for supporting me through the ups and downs of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF ACRONYMS & DEFINATIONS

EA- Evolutionary Algorithms

CA- Cultural Algorithms

GA-Genetic Algorithms

MPCA-Multi-Population Cultural Algorithms

D-MPCA-Dominance Multi-Population Cultural Algorithm

Haploidy: Single set of a chromosome

Diploidy: Two sets of chromosomes of the same gene

H-MPCA: Heterogeneous Multi-Population Cultural Algorithm

MCAKM- A Novel Multi-Population Cultural Algorithm Adopting Knowledge Migration

TAMPCA- Transfer Agent based Multi-Population Cultural Algorithm

1 Introduction

1.1 Evolutionary Computations

Evolutionary Computations is a branch of Artificial Intelligence. The algorithms that come under this section adopt Darwin's principles of Evolution; hence, they are called Evolutionary Algorithms. Technically speaking these algorithms can be considered as Global optimization problems Kybernetes [1998]. There are different algorithms that come under evolutionary computations, such as :

1. Cultural Algorithms
2. Ant Colony Optimization Algorithms
3. Genetic Algorithms
4. Differential Evolution Algorithms
5. Swarm Intelligence

Evolutionary algorithms also come under the evolutionary computations, and Multi-Population Cultural Algorithms (MPCA) are a subset of evolutionary computations and therefore are also considered optimization algorithms.

1.2 Dominance

Dominance, according to Burger [2008], is the process of elimination of redundancy in selecting the gene from a group of genes. When one allele is expressed over another allele at the same location of a gene, the first allele is said to be dominant over the second one [Hunter, 1996]. Clarke [1997] states that dominance can be useful for the selection procedure as a factor of population hierarchy. This is necessary for both the selection and for finding the origin of the allele and its features. Dominance is a feature in biological systems with evolutionary advantages. It assists the survival of the population by allowing them to adapt to the changes in the environment. It is a vital aspect in Evolutionary algorithms.

> “In the struggle for survival, the fittest win out at the expense of their rivals because they succeed in adapting themselves best to their environment.”
> –Charles Darwin

> “The value and utility of any experiment are determined by the fitness of the material to the purpose for which it is used, and thus in the case before us it cannot be immaterial what plants are subjected to experiment and in what manner such experiment is conducted” –Gregor Mendel

1.3 Research Motivation

The main motivation for the research comes from observing different optimization problems. We found that many different algorithms have been proposed to optimize knowledge sharing in cultural and multi-population cultural algorithms, but most of them were more problem specific. So, we want to try to optimize MPCA by implementing dominance. We wanted to use dominance as motivated from the work of Charles Darwin (1809-1882) and Gregor Mendel (1882-1884). Their unparalleled work on evolution and dominance respectively led us to implement dominance in MPCA. While working on Cultural algorithms we observed that multi-population cultural algorithms were a new field of study and that the research to be done on them was huge also its implications are also vast. In our thesis we mostly emphasize on implementing complete dominance on MPCA.

1.4 Thesis Contribution

In our work, we show the implementation of artificial dominance function on an MPCA network and how dominance can play a key role in the knowledge migration in between different sub-populations. In our study we hypothesize that when individuals migrate from one sub-population to another sub-population and they will affect the whole population. We started our research on the hypothesis that dominance can affect the population on a large scale and that is what we wanted to test. We created our own MPCA framework based on Guo [2011] and implemented dominance on the populations by migrating the individuals from one population to the other in order to share information among the sub-populations. Various benchmark functions (CEC 2010, CEC 2013) were used to test the MPCA with dominance. Dominance was tested on different population sizes and multiple sub-populations (Population Size 5, 10, 20). The functions were also tested on both uni modal and basic multi-modal functions.

1.5 Thesis Outline

In the first chapter of this thesis book we explained about our motivation and contribution, the research is divided into the remaining seven chapters. Chapter 2 consists of the related work done on dominance and MPCA. This chapter details all the relevant work done on dominance and multi-population cultural algorithms. This section consists of 8 papers of which 5 are related to dominance and 3 are related to MPCA. In chapter 3, we explain dominance and the MPCA in detail. Chapter 4 consists of a detailed explanation of our framework and our approach. Chapter 5 explains our experimental setup and the parameters that were used in the experiments. Chapter 6 outlines the benchmark functions that we used in our framework. Chapter 7 consists of all the results of our experiments and we conclude the thesis in chapter 8.

2 Related Work

This chapter consists of the all the related work used to build the fundamental concepts and architecture of our thesis work. In this study we explain about the research motivation that we got form the papers in detail. And the literature related to dominance and multi-population cultural algorithms. The first section consists of papers related to Dominance and the second section consists of papers related to MPCA.

2.1 Dominance

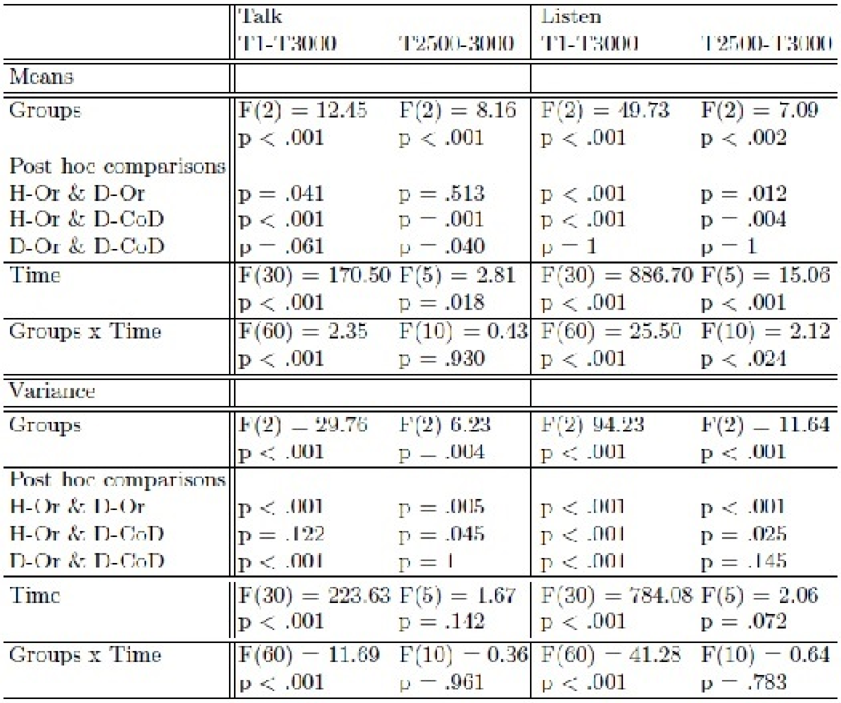
2.1.1 Non-stationary Function Optimization Using Genetic Algorithms with Dominance and Diploidy

Goldberg et al [1987] first implemented dominance in genetic algorithms. The authors tried to improve the performance of Genetic Algorithms in varying environments over time using the dominance mechanism. The authors referred to Hollstien [1971] and De Jong [1975]. They found it abnormal that previous researchers who used genetic algorithms have not applied the dominance operator and that the mechanisms were tested on haploid genetics exclusively. The authors have proposed a new idea of using diploidy (2 chromosomes) genetics and dominance together. In this approach the authors tried to protect the gene memory by shielding the genes. To study the performance of genetic algorithms (GA) with dominance the authors compared three different schema and conducted their experiments on the Blind-Knapsack Problem. The authors claimed that diploidy was superior to haploidy in Non stationary Knapsack Problem. They also claimed that dominance helped to provide an extra shield to the gene memory. The authors claim that diploidy was superior to haploidy in the Non-stationary Knapsack Problem with an oscillating weight constraint. They also claim that using the new algorithm could achieve faster response to Non stationary environments.

2.1.2 Preserving Variability in Sexual Multi-agent Systems with Diploidy and Dominance

Bowers et al [2006] tried to implement a dominance mechanism in preserving the variability of the sex gene in multi-agent systems. They referred to the works of Lin et al. [1994], Mauldin et al. [1984], and Potts et al. [1994]. The authors state that there was no specific function for evaluating the fitness for evolutionary strategies. They also observed that the fitness values are not generated continuously in evolution strategies but it was different in Genetic Algorithms. They introduced a new mechanism to preserve the variability of the sex gene using dominance and preventing convergence. The authors tried to solve this problem by using two types of dominance's i.e. Complete Dominance (Mendelian Dominance) and Co-Dominance in the algorithm. The authors used VUScape, which is an environment in the JAVA Artificial Worlds and Agent Societies (JAWAS) Framework. The tests were conducted on a 2-D grid which is populated by virtual agents. The authors tried to measure the average values of each dominance trait and to observe the change in the value of variance overtime. They claimed that diploidy preserved the variability and that Medallion Dominance or Complete Dominance performed better than Co-Dominance. They also observed that in all the cases, the algorithms failed to perform well in the first 1000 runs; this concludes that the populations lost their variability. However, an increase in the number of generations showed an increase in the variability. In Figure 1 the authors analyze the mean and variance on two different tests.

The authors claimed that complete dominance preserved variability better than co-dominance. In both single and multi-chromosomes, dominance helped in preserving the variability which, in turn, helped in the evolution of the sex gene.



	Talk		Listen	
	T1-T3000	T2500-3000	T1-T3000	T2500-T3000
Means				
Groups	F(2) = 12.45 p < .001	F(2) = 8.16 p < .001	F(2) = 49.73 p < .001	F(2) = 7.09 p < .002
Post hoc comparisons				
H-Or & D-Or	p = .041	p = .513	p < .001	p = .012
H-Or & D-CoD	p < .001	p − .001	p < .001	p − .004
D-Or & D-CoD	p − .061	p − .040	p − 1	p − 1
Time	F(30) = 170.50 p < .001	F(5) = 2.81 p = .018	F(30) = 886.70 p < .001	F(5) = 15.06 p < .001
Groups x Time	F(60) = 2.35 p < .001	F(10) = 0.43 p = .930	F(60) = 25.50 p < .001	F(10) = 2.12 p < .024
Variance				
Groups	F(2) − 29.76 p < .001	F(2) 6.23 p − .004	F(2) 94.23 p < .001	F(2) − 11.64 p < .001
Post hoc comparisons				
H-Or & D-Or	p < .001	p = .005	p < .001	p < .001
H-Or & D-CoD	p = .122	p = .045	p < .001	p = .025
D-Or & D-CoD	p < .001	p = 1	p < .001	p = .145
Time	F(30) = 223.63 p < .001	F(5) = 1.67 p = .142	F(30) = 784.08 p < .001	F(5) = 2.06 p = .072
Groups x Time	F(60) − 11.69 p < .001	F(10) − 0.36 p − .961	F(60) − 41.28 p < .001	F(10) − 0.64 p − .783

Figure 1: Analysis of variance [Bowers et al. 2006]

2.1.3 Haploid Genetic Programming with Dominance

Vekaria et al [1997] used the dominance mechanism on haploid structures to solve the genetic crossover issue. The authors referred to the work of Dawkins et al [1989], Vekaria et al. [1997] and Koza et al. [1992] in this paper and state the way they tried to use dominance in solving the problems related to genetic programming. The authors stated that dominance was not considered as an evolutionary factor during crossover in genes. The new idea that the authors proposed was the use of dominance in genetic programming. The authors used the dominance mechanism to improve the genetic programming. Given a parse tree, the nodes which were more dominant are used to produce/generate the population. The selection for the crossover mechanism is done based on the breadth first search method. The authors implemented 2 types of dominance crossovers:

1. Single Node Dominance Crossover (SNDC)

2. Sub-tree Dominance Crossover (STDC)

They found that in SNDC the population converged too early and at times it also failed in producing a solution. The authors explained that this phenomena occurred because the trees were not allowed to grow. The results were not as per their expectations. The functions that were used had different number of arguments they distorted the shape of the genetic programming tree. They also observed that in STDC the trees bloated to maximum size but the results generated were not as per expectations. Also they noted a repetition of sub-trees in the tree. The authors concluded that dominance crossover was not an appropriate operator to be used with their genetic programming structure. They also claim that breadth first search method failed to solve their problem.

2.1.4 Learning the Dominance in Diploid Genetic Algorithms for Changing Optimization Problems

Yang et al. [2007] stated that in genetic algorithms there were many problems that needed answers like dynamic optimization problems. They referred to the works of Branke et al. [2002] and uyar et al. [2005]. The researchers state that in dynamic optimization problems (DOP), the goal of GAs has shifted from locating a single optimal solution quickly and precisely to track the motion of the optimum over time. In this paper, the authors proposed a new adaptive dominance scheme for diploid GAs that helped in solving dynamic optimization problems using different DOP generators that they constructed. A series of DOP were used as a test bed and they conducted their experiments. The authors compared their new algorithm with two different dominance schemes Ng-wong and the addictive dominance scheme with dominance change for dynamic GAs. Figure 2 gives the representation and evaluation of the individual in genetic algorithms and the process in which the fitness value is evaluated.

The authors stated that in their experiments their algorithm outperformed the

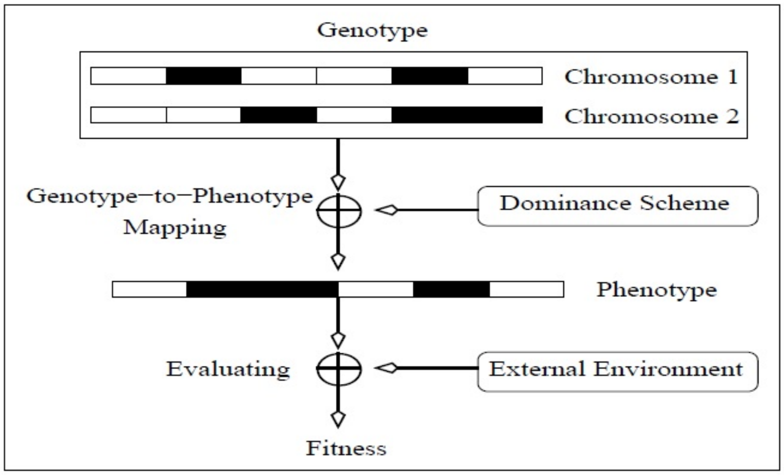


Figure 2: Representation and evaluation of an individual in dynamic genetic algorithms [Yang 2007]

other algorithms and this was proven by plotting a table with the data for best-of-generation fitness against generations on a dynamic One Max problem. The authors claim that their dominance scheme was effective in improving the performance of dynamic genetic algorithms over dynamic environments. They also claimed that their method outperformed Ng-wong scheme for dynamic environments.

2.1.5 A New Real-Valued Diploid Genetic Algorithm for optimization in Dynamic Environments

Amineh et al. [2014] stated that genetic algorithms with dynamic changing environments had many problems, they tried to solve one such problem, the optimization in dynamic environments using a new dominance mechanism and diploidy together. They refer to the work of Goldberg et al. [1981], kominami et al. [2010] and Yang et al. [2007]. The problem that the authors found in the previous work was that in dynamic environments the Genetic Algorithms converge to a single solution and the populations loses its diversity, due to which they cannot adapt to the changes in the environment. The authors also explained about the work of Lewis et al. [1998] who did a comparative study and observed that simple dominance scheme was not

Parameter	Value
Number of Peak Variable between	1 to 100
Change frequency Variable between	500 to 1000
Height change	7.0
Width change	1.0
Peaks shape	Cone
Basic Function	NO
Shift Length	1.0
Number of dimension	5
Correlation coefficient	0
Peaks Location range	[0-100]
Peak height	[30.0-70.0]
Peak width	[1-12]
Initial value of peaks	50.0

Figure 3: Moving Peak Parameter Setting [Amineh et al 2014]

enough to get the optimal value in genetic algorithms. The authors proposed a new real-valued diploid genetic algorithm. In this algorithm each individual is considered as a diploid chromosome and a global domination map was assigned which they used to determine the phenotype of the individuals. The authors evaluated the fitness of all the populations and the values got worse by 20% were selected and distributed randomly among the populations. The authors have conducted their experiments on Moving Peak Benchmark as it was both dynamic and continuous. They also compared their model with different genetic problems. The experimental setup is specified in Figure 3. All the parameters and the range of the values of the experiments are specified in the following table.

The experiments were ran 30 runs on each GA for Moving peak benchmark and the averages were calculated. The authors, then compared Ng-Wong –Dominance mechanism, addictive with dominance change mechanism and observed the effects carefully. The authors state that based on their results their new algorithm performed better than the others on different frequency i.e. no of peaks. The authors also claimed that their algorithm gave homogeneous outputs. They also stated that MPB (moving peak benchmark) is the most famous dynamic and changing benchmark and they have compared it with different diploid genetic algorithms. The authors made it clear that there was a significant difference between the performance of their algorithm and others.

2.2 Multi-Population Cultural Algorithms

In this section we discuss about all the literature that are relevant to our research. We explain in detail multi-population cultural algorithms and how they were used by different authors.

2.2.1 Heterogeneous Multi-Population Cultural Algorithm

Authors Raeesi et al. [2013] stated that given a group of sub-populations consisting of different cultural algorithms, they do not communicate directly with each other, so to solve this problem multi-population cultural algorithms (MPCA) was introduced. They refer to the work of Digalakis et al. [2002], Holland et al [1975], Koza et al. [1992] and Reynolds et al. [1994]. They stated that evolutionary algorithms were successfully applied on different optimization problems but had some issues like immature convergence. This was because they couldn't preserve the diversity of population over generations. The authors proposed a new MPCA in which the sub-populations were same and the optimization parameters are divided among the sub-populations. For every sub-population a specific set of partial solutions that were responsible to optimize the parameters exist were assigned. In Figure 4, a detailed architecture of the proposed algorithm (HMP-CA) is presented.

The authors implemented the Heterogeneous multi population cultural algorithm (H-MPCA) in Java platform. In the experiments the population size was considered as 1000 and the number of sub-populations was fixed to 30 with size of 33 in each sub-population. The experiments were run for up to 10000 generations and 10 iterations. The authors were able to find the minimum value for seven numerical optimization

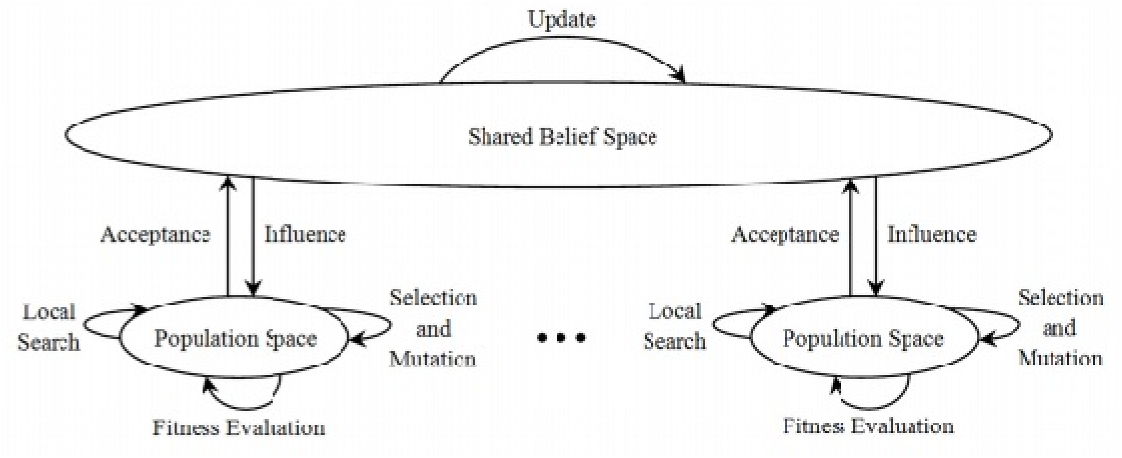


Figure 4: HMP-CA Architecture [Raeesi et al. 2013]

functions out of 8. They could not find the minimum value for one of the functions over the given time. The authors also explained that their algorithm was not able to obtain the minimum value for one specific function in the 10000 generations for 5-sub populations but it was able to do it if given permission to use higher number of generations. The authors claimed to have found the minimal values of the numerical optimization functions and also their model was efficient in both the time and space complexity.

2.2.2 A novel Multi-Population Cultural Algorithm Adopting Knowledge Migration

Guo et al. [2010] stated that in MPCA the information is exchanged in sub-populations but at the individual level and not at the population level. The individuals migrate among sub-populations for exchanging information. But, they do not consist of complete knowledge of the sub-population, due to which they cannot reflect enough information. This limits the evolution performance.

They referred to the works of Reynold [1994], Jin et al. [1999], Bin [2005]. Most researchers did not take into account the implicit knowledge in MPCA. The authors Digalakis and Margaritis [2002] first implemented MPCA. They also stated that the individuals exchange information among the belief-space of the sub-populations, but their method was not entirely clear.

The authors proposed a new novel MPCA adopting knowledge migration. In the new algorithm knowledge is exchanged implicitly among sub-populations instead of individuals. In MPCA there are 'n' number of sub-populations and each sub-population adopts a cultural algorithm and the information is exchanged among them by migrating individuals at regular intervals. Figure 5 reperesnts the architecture of the proposed MCAKM architecture.

In order to justify the new algorithm the authors implemented some high-dimension benchmark functions and the performance of their algorithm was compared and analyzed on different parameters. The tests were run on a population size of 30 and 3 sub-populations with a selection proportion of 0.3 and a mutation probability of 0.08. The experiments were run 20 times and for 100 iterations.

In order to validate their algorithm the authors compared it with general cultural algorithm and other MPCAs adopting influence range. They found that their algorithm MCAKM performed better and also had faster convergence speed along with better solutions. The new MPCA was derived from human cultural interactions. And the knowledge extracted from the evolution process was more efficient

2.2.3 Knowledge Sharing Through Agent Migration with Multi-Population Cultural Algorithm

Hlynka et al. [2013] explained that sharing of knowledge among agents/individuals in an MPCA was always a problem. The authors tried to address this problem. They refer to the work of Reynolds et al. [2003], Kobti et al. [2006], Guo et al. [2010] and Raeesi et al. [2012]. The authors stated that other researchers didn't use knowledge

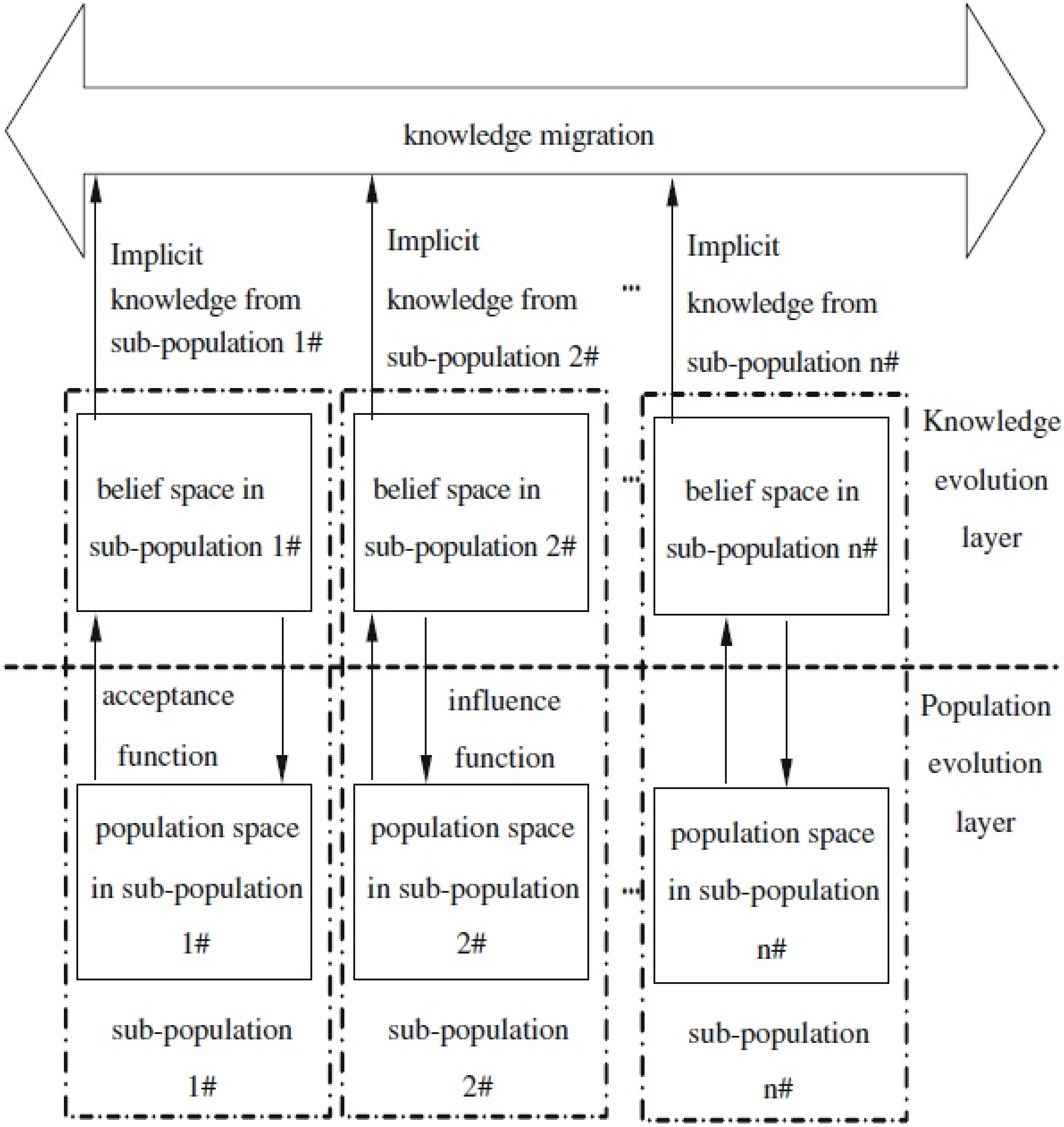

knowledge migration
Implicit knowledge from sub-population 1#
Implicit knowledge from sub-population 2#
...
Implicit knowledge from sub-population n#
belief space in sub-population 1#
belief space in sub-population 2#
...
belief space in sub-population n#
Knowledge evolution layer
acceptance function
influence function
Population evolution layer
population space in sub-population 1#
population space in sub-population 2#
...
population space in sub-population n#
sub-population 1#
sub-population 2#
sub-population n#

Figure 5: MPCA adopting knowledge migration [Guo et al.2011]

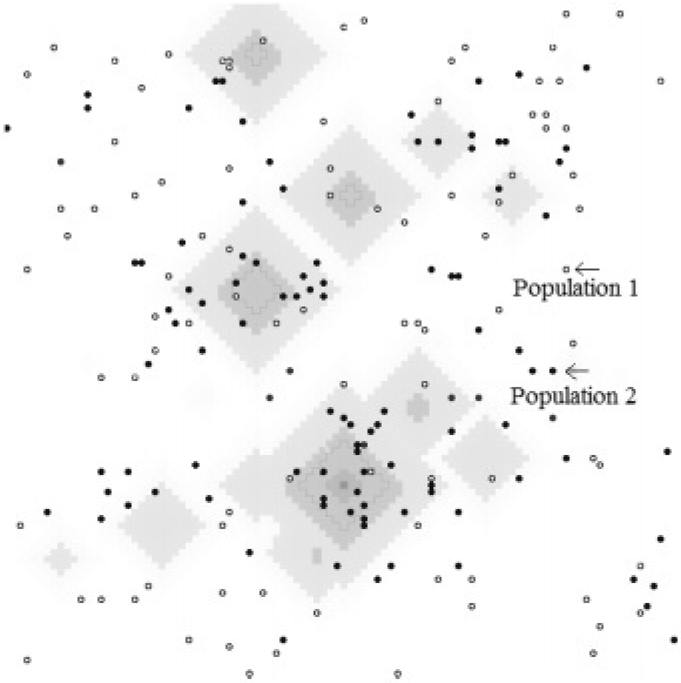


Figure 6: Domain map at beginning of Experiment [Hlynka et al. 2013]

migration for sharing the knowledge and they propose a method for the transfer of knowledge in an MPCA by migrating the agents among the sub-populations. The authors ran their experiments on Repast Simulation tool. They used Moving Peak's cone's world domain to test their algorithm. They calculate the sub-populations performance over time by transferring individual agents from one population to other populations. They ran their experiments by transferring 1%, 5%, 10%, 20%, and 50% of the populations and calculated the changes in the sub-population's performance.

They found the best and worst run times for different knowledge's i.e. Topographic and Situational Knowledge. The authors stated that their algorithm performed better only for 1% of population transfer and the results for other transfers were not as per their expectations. In Figure 6 the multi populations are displayed. There were two sub-populations and both were displayed in the figure.

The authors claimed that their model performed very well when 1% of the sub-populations were being transferred and also claim that the transfer of a small group of agents among the sub-populations could improve the consistency of both the sub-populations.

2.3 Chapter Conclusion

From the above mentioned works we will be implementing the complete dominance feature by Amineh et al.[2014] in Multi-Population Cultural Algorithms. We try to optimize the knowledge migration in MPCAs by implementing the artificial dominance feature.

3 Evolutionary Algorithms

This section consists of a detailed explanation about evolutionary algorithms, types of evolutionary algorithms, genetic algorithms, cultural algorithms and MPCA. In this chapter we try to give a brief overview of the algorithms that are relevant to our field of study.

3.1 Evolutionary Algorithms

Evolutionary algorithms are simple computer simulated programs that try to solve very complex problems by using the principles proposed by Darwin. Jung et al. [2006] In Evolutionary Algorithms it is generally hoped that over generations the best optimal solutions is generated. These are the algorithms that are inspired from natural phenomena like mutation, recombination, selection and reproduction. These algorithms are used to solve population based meta heuristic optimization problems. EAs consist of populations or the individuals that are randomly generated, this population is considered as the initial population or the parent generation. By combining mutation and crossover properties a new population i.e. offspring's are generated from the previous set of populations. In EAs a specific function is used for calculating the fitness of the individuals in the populations this fitness function is useful because it helps in applying mutation, crossover and selection for the next generation of the population Eiben et al. [2003]. Evolutionary algorithms are used for the optimization of population related problems. There is a need for optimizing the problems as the populations cannot preserve or store the diversity over generations. For optimizing these meta heuristics, in order to get the optimal solution evolutionary algorithms are useful.According to Jones et al. [1998] EAs are a group of probabilistic optimization algorithms that are mainly based on similarity between biological systems and artificial systems. There are different types of Evolutionary Algorithms such as:

1. Genetic Algorithms
2. Cultural Algorithms
3. Ant Colony Optimization Algorithms

3.2 Genetic Algorithms

Genetic Algorithms were first created by Holland in the year 1973 but they became famous because of the work of Goldberg [1989]. Genetic Algorithms are a subset of Evolutionary Algorithms hence, they are population based algorithms. These algorithms are mainly used to solve search related and other optimization problems. GAs consist of a group of individuals as the initial population that are randomly generated and these individuals are used to find the optimal solution in a given system. The individuals also exchange information using the operators like mutation, crossover, selection and recombination. In GAs after each generation the best individuals are selected for mutation, crossover and to generate the next population based on their fitness values. According to Coley et al. [1999] GAs are numerical optimization algorithms that are motivated by natural genetics and selection. The GAs are simple and are also easy to code. They are not initiated at a single point but are rather spread out in a search space in order to get the optimal solution. The GAs use three operator's mutation, crossover and selection to direct the population to get an optimal solution.

1. **Selection:**

   This operator attempts to pressurize the populations in a way similar to that of natural selection that is found in biological systems. The individuals that perform better or that are fitter are selected and the weaker individuals are discarded as the fitter individuals have a good chance of transferring the information to the next generations.

2. **Crossover:**

   This operator permits the solutions to share information similar to the process that is followed in the natural phenomena like sexual reproduction. The crossover operator is mainly useful as it helps in the reproduction of the off springs.

3. **Mutation:**

   This operator is used to change or flip the values of the individuals and hence it is very rarely used in Genetic Algorithms.

GAs are helpful in solving constrained and unconstrained optimization problems that are based on natural selection. Genetic Algorithms continuously modify a population of individual solutions and over time populations evolve to generate an optimal solution. Genetic Algorithms have vast applications in the fields of Image Processing, VLSI, Laser Technology, etc. Genetic Algorithms are commonly used to solve stationary optimization problems and based on the history these algorithms performed very poorly in real world problems i.e. dynamic changing environments. Genetic Algorithms are designed to solve the static problems and not dynamic the reason for this is that given a dynamic environment the individuals converge quickly to obtain a solution and due to this the population loses its genetic diversity.

3.3 Cultural Algorithms

Cultural Algorithms were first introduced by Reynolds et al. [1994]. Cultural algorithms are an extension of genetic algorithms. According to Reynolds et al. [1994] it was suggested that cultural evolution provides societies to adapt or evolve to the changes in the environment. Durham [1992] defines culture as a “system of symbolically encoded conceptual phenomena that are socially and historically transmitted within and between populations”. Cultural Algorithms have gained a lot of attention in recent years, many researchers tried to develop the cultural evolution process on the perspectives of traits between individuals and formation of generalized belief space based on the individual experiences for both micro and macro – evolutionary levels respectively. "Cultural Algorithms are a class of computational models of cultural evolution that support dual inheritance perspective". Reynolds et al. [1994] Cultural Algorithms consist of two components

1. Belief Space
2. Population Space

3.3.1 Belief Space

This component consists of different kinds of knowledge relevant to solve the problem. Due to this the belief space is divided into separate categories. These categories contain different domains of knowledge which the population poses of the search space Kobti et al. [2003]. The belief space can be considered as a container that can store the knowledge that is extracted from the population. The belief space is efficiently updated after each iteration by the best individuals of the population. Based on the fitness function the individuals are selected this process is similar to the process used in Genetic Algorithms.

Artificial 'belief spaces' store the knowledge that is gained during the execution of the algorithm and it influences the individuals along with the genetic evolution. There are different types of knowledge Hlynka et al. [2013]. They are:

1. Temporal Knowledge (Historic Knowledge): knowledge of past occurrence
2. Domain Knowledge: relationships and information about the domain objects
3. Spatial Knowledge: topographic knowledge
4. Normative Knowledge: Range of better choices

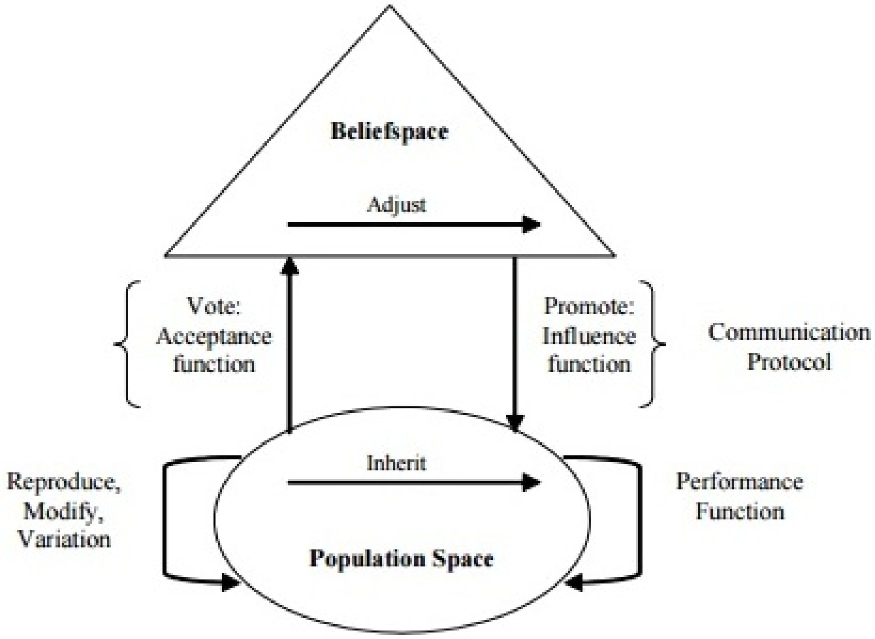


Figure 7: Cultural Algorithm Architecture kobti et al. [2003]

5. Situational Knowledge: successful and unsuccessful instances

3.3.2 Population Component

The population component in CAs is almost same as that of GAs. The population component/space interacts with the belief space by two connections i.e. influence function and acceptance function. The best individuals are sent from the population component into the belief space through the acceptance function. After that the belief space is updates its knowledge and then the population component by the influence function. The population component uses this knowledge to generate the next generation of individuals. In Figure 7 we display the basic architecture of cultural algorithms.

Algorithm 1 Pseudo Code for CA Kobti et al [2004]

```
Initialize Population Space
Initialize BeliefSpace
Repeat until termination criteria is met
Perform actions of the individual populations in the population space
Evaluate each individual using the fitness function
Select the parents to reproduce a new generation of offspring
Let the beliefspace modify the genome of
the descendants using the influence function
Update the beliefspace by using the accept function
```

3.3.3 Communication Interface/Protocol:

The CAs in order to communicate between the Population component and the beliefspace component require an interface. So, the individuals with the best fitness values update the belief space using the Update function. Another function the influence function which effects the population component. Kobti et al. [2004]

3.4 Multi-Population Cultural Algorithms (MPCA)

CAs incorporating the multi-population concept are known as Multi population cultural algorithms. MPCA are an extension of cultural algorithms. These algorithms are mainly useful to solve optimization problems. The MPCA were first designed by Digalakis [2002]. MPCA are mainly used to solve the problem of knowledge sharing/migration. MPCA can be called as a group of smaller Cultural Algorithms that solve similar problems. In the first model of MPCA individual migration was used to share knowledge among different sub-populations to schedule electrical generators. They consists of more number of parameters that are to be optimized, when they are compared to the traditional CAs. For example the parameters like the number of sub-populations, size of each sub-populations and the migration procedure. MPCA was successfully applied in different fields like multi-modal optimization problems Guo et al. [2011], interactive optimization problems Yi-nan et al. [2011] and constraint opti-

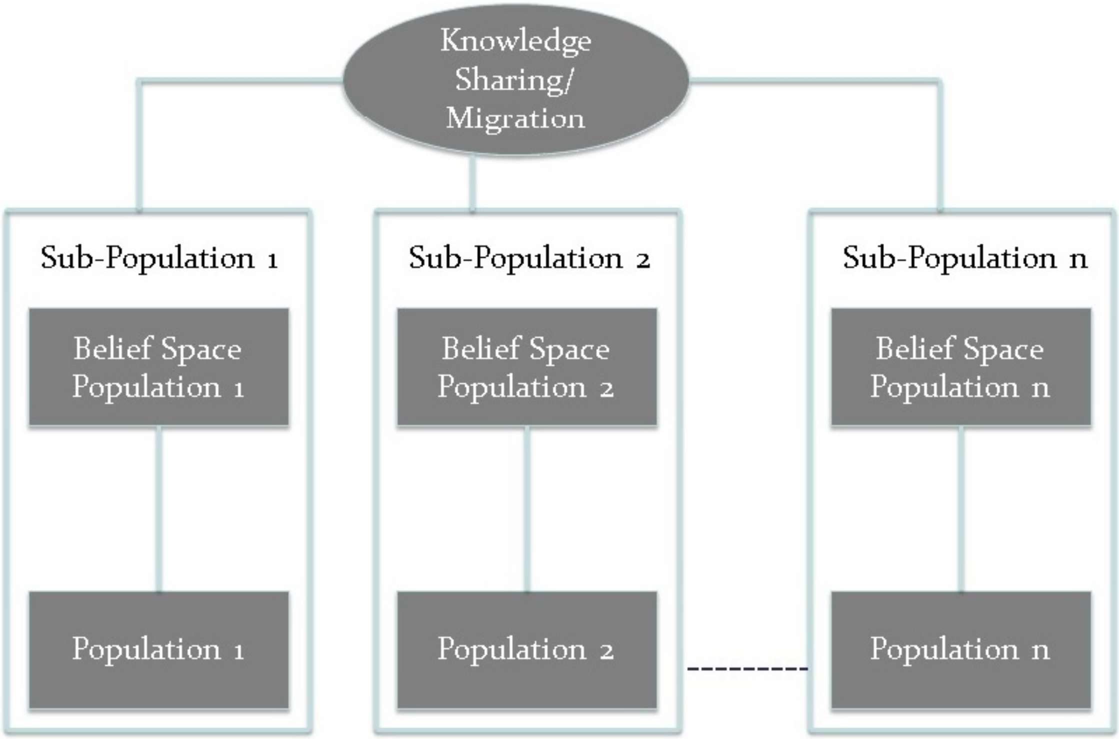

Knowledge Sharing/ Migration
Sub-Population 1
Belief Space Population 1
Population 1
Sub-Population 2
Belief Space Population 2
Population 2
Sub-Population n
Belief Space Population n
Population n

Figure 8: MPCA Architecture

mization problem. MPCA were also used in fields like optimization problems, supply chain management and neurofuzzy inference systems. MPCA were applied on electrical generator maintenance scheduling problem in which the optimization problem is still unsolved. Many different methods were available that offer optimal solutions for small-size problems, but they do not guarantee the optimal solution for all the problems. So, to find an optimal solution for all the problems Digalakis et al. [2002] proposed an MPCA for electrical generator scheduling problems. Many researchers have used MPCA to solve different optimization problems. Figure 8 consists of the MPCA architecture.

There are different versions of multi-population cultural algorithms like the multi-population cultural genetic algorithms (MCGA), Multi-population cultural differential evolution (MCDE) and multi-population cooperative particle swarm cultural algorithm (MCPSCA).

3.4.1 Knowledge Migration

Guo et al [2011] proposed a new method called as knowledge migration for sharing information among different sub-populations. In this method, the knowledge is exchanged among the sub-populations instead of migrating the individuals from one sub-population to the other. This proposed model of migrating knowledge is better because knowledge has more information of the older generations and the direction in which the evolution is taking place. Guo implemented this method on an MPCA network in which the populations are divided into equal sizes of sub-populations and CA is applied on each sub-population separately. The exchange of cultural knowledge occurs for every predefined number of generations.

In every local CA the best individuals are selected and using the acceptance function they update the belief space to store the topographic and normative knowledge. The normative knowledge stores a record of the search space and the topographic knowledge governs the process of finding the best individuals in every sub-population and thus find the optimal solution. The knowledge acquired by the belief space is used by the influence function to make the mutation operator for offspring generation. Guo proposed a strategy for merging both the private knowledge of the belief space and the migrated knowledge of each sub-population. They only considered topographic knowledge for creating this knowledge.

3.5 Chapter Conclusion

This chapter consists of evolutionary algorithms like genetic algorithms, cultural algorithms and multi-population cultural algorithms detail explanation. A complete explanation of knowledge migration in MPCA,the architecture of cultural algorithms and multi-population cultural algorithms.

4 Dominance

Dominance was first introduced by Gregor Mendel (1822-1884). He is considered as the father of modern genetics. Mendel ran experiments on garden peas and discovered that there were two distinct phenotypes in all the plants that were very discrete like the red versus white flowers, round versus wrinkled seeds, tall versus short plants and yellow versus green seeds. He also observed that when separately bread the plants always created the same sets of phenotypes in every generation. But, when these plants were inter bread among multiple phenotypes resulted in the exhibition of more properties of one phenotype over the other in the ration of 1:3. This clearly showed that the plant exhibited the properties of one phenotype over the other and this was called as Dominance, Bernstein et al. [1968].

Dominance is the process of masking the properties of one phenotype over the other. It is also the process in which the gene expression of one allele is expressed over the other. The terms allele, genotype, gene, phenotype and homozygote were not introduced by Gregor Mendel but were added to his work by others. Mendel presented the method of notation of lowercase and capital letters for differentiation recessive alleles and dominant once. This method is still being used. There are Different Types of Dominance, they are:

1. Complete Dominance

2. Incomplete Dominance

3. Co-Dominance

4.1 Complete Dominance

When one allele completely covers or masks the effects of other allele in a heterogeneous genotype then it is known as complete dominance King et al [2013]. In this

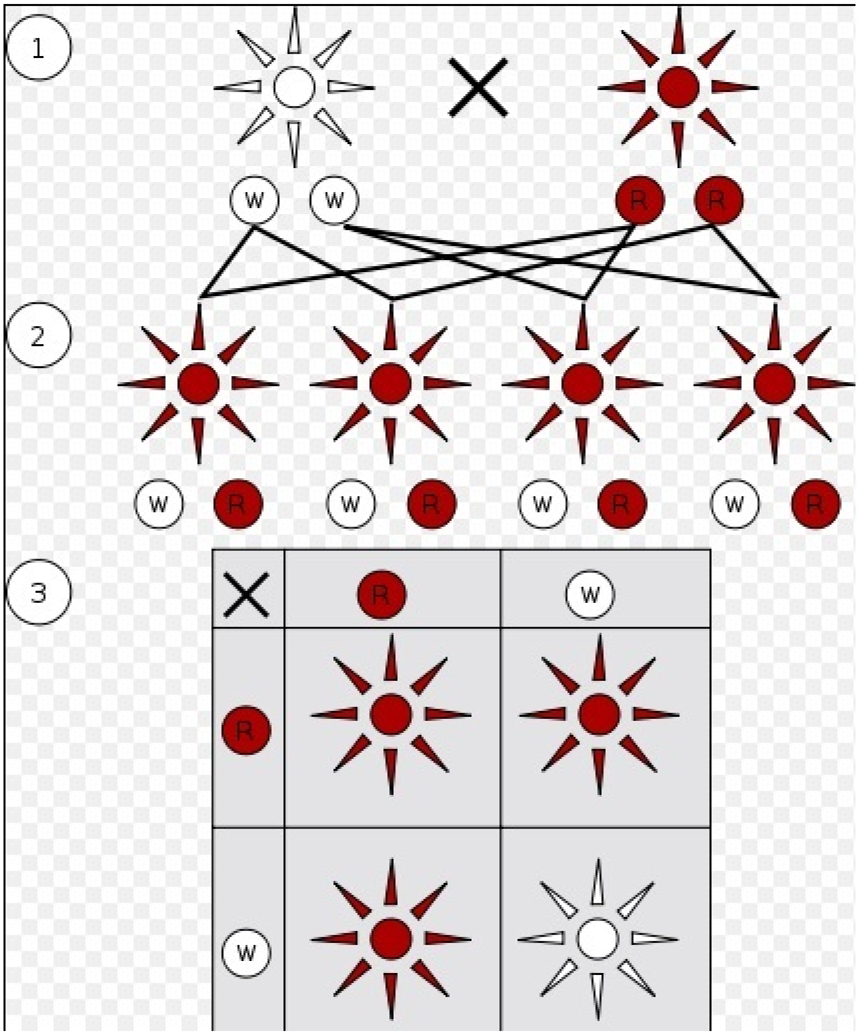


Figure 9: Complete Dominance Mendel et al [1996]

process the allele that is being masked is known as recessive and the former allele is known as dominant. An example for complete dominance could in the color of the eye.

In this Figure 9 we can see that the there are two gene types exhibiting different properties. In step 1, one type exhibiting the color red and the other exhibiting the color white. The red property represents dominance and the white represents recessive. When crossover occurs we expect that the offspring's generated will have equal properties of both the genes but due to dominance we can see that the offspring's generated/produced exhibit the properties of the red gene over white gene when the interaction was between red and white. But when there was no interaction between both white genes and there is no presence of dominant gene then the properties exhibited are that of the white gene. We will be implementing this mode of dominance in our model.

4.2 InComplete Dominance

A brief definition of incomplete dominance can be the dominance property in which one allele partially effects or masks the effect of other allele. This type of dominance is mainly visible in plants. This property exists in plants where different genes are grown to form a hybrid plant. In the example given below we can see that two different flowers exhibiting their properties simultaneously.

Incomplete dominance also known as partial dominance. After crossing, no single property of the plant is directly exhibited in the flowers that are generated as hybrids. Neither of the parent properties are completely exhibited by the offspring's. In the Figure 10 we can see that the color of the offspring is neither red not white but its pink which is an intermediate color between white and red this is a clear example of incomplete dominance as the properties of both the parents are partially visible.

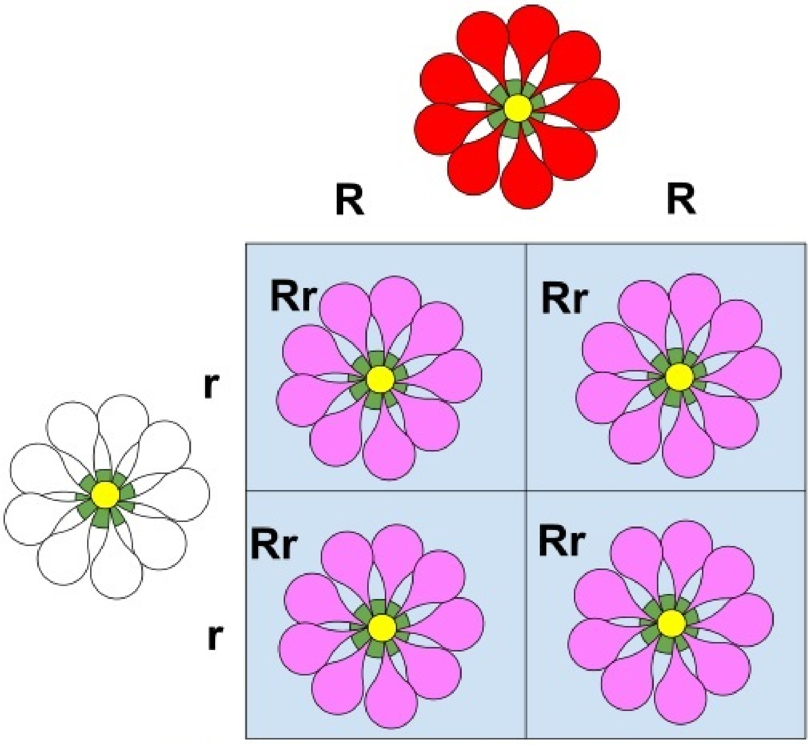


Figure 10: In-Complete Dominance Mendel et al [1996]

4.3 Co-Dominance

Co-dominance is a different type of dominance in which the characters of both the parent genes co-exhibit in the offspring's i.e. both types of allele properties are visible on the next generation. Another definition for co-dominance can be stated as the co-existence of two dominant alleles with a relatively uniform variance. It is a bond among two different types of genes. Co-dominance can be explained as a condition where both the alleles are dominant and none of them are recessive due to this the hybrid or the offspring generated exhibits the properties of both the alleles. Co-dominance is explained with the following example in Figure 11.

In the above example we can see that when a brown ox is mating with a white cow and the offspring generated exhibits the properties of both the parents. This is due to the dominant nature of both the parents. Co-dominance is completely different form incomplete dominance where partial dominance occurs. It is a state where the

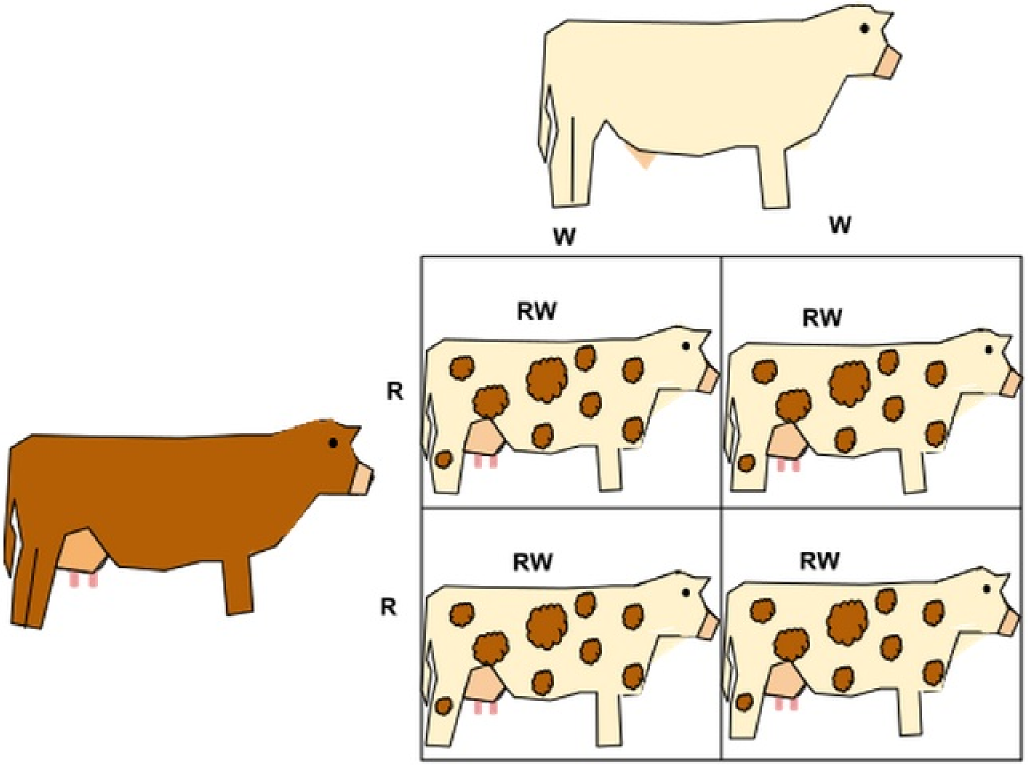


Figure 11: Co-Dominance Mendel et al [1996]

properties of both the parents are clearly visible in the next generation of offspring's.

4.4 Chapter Conclusion:

As part of our research work we will be implementing complete dominance feature in MPCA.

5 Proposed Approach

In this chapter we explain in detail about our framework. We describe how the MPCA network was designed and also explain about how we applied dominance on the new D-MPCA network (Dominance Multi-Population Cultural Algorithms).

5.1 D-MPCA Network

In our approach we developed a new multi-population cultural algorithm network based on Guo et al [2011] and assimilated many sub-populations and assigned each sub-population with a separate cultural algorithm. There were many optimization parameters for the MPCA network, each sub-population consists of individuals trying to optimize their own population as each population is assigned with a local cultural algorithm, and every sub-population tries to optimize their individuals to get the best solution i.e. the fitness values. In our network all the sub-populations share a common belief space known as global belief space, they can update the belief space with the best known value of each sub-population after every iteration. The global belief space stores normative knowledge of all the sub-populations i.e. the knowledge of the acceptance range of all the individuals in the sub-populations.

The individuals from one sub-population migrate to the other sub-population in order to improve the overall fitness value of the sub-populations. The individuals from the population with weak individual's, when compared to that of the other populations is selected for migration. After selecting the sub-population, the individuals are migrated into other sub-populations continuously and the fitness values of all the individuals in the new set of sub-populations are evaluated. The mean value of the sub-populations before migration and after migration are calculated and evaluated. Due to the migration of individuals the Dominant individuals effect the populations mean fitness value of the sub-populations. We tried to compare this effect of migration of the individuals on both the sub-populations and the whole population.

5.2 Migration in D-MPCA

Migration in D-MPCA is based on the fitness values of the sub-populations. After evaluating the individuals in the sub-populations, the sub-populations that have weak fitness values are migrated into the sub-populations with better fitness values.Figure 12 explains about the architecture of knowledge migration in D-MPCA.

Every individual in the population has its own fitness value and these values effect the population on the whole. Individuals from one sub-population are migrated to other sub-populations based on their fitness values. This process of migrating the individuals is known as Dominance migration. The dominance migration is similar to the dominance occurring in nature Peischl et al [2008].

5.3 Dominance in MPCA and Pseudo Code

5.3.1 Evaluation of Individuals

The individuals that are least dominant i.e. with least weights or fitness values are migrated to other populations and this process is repeated until the better fitness values are obtained.

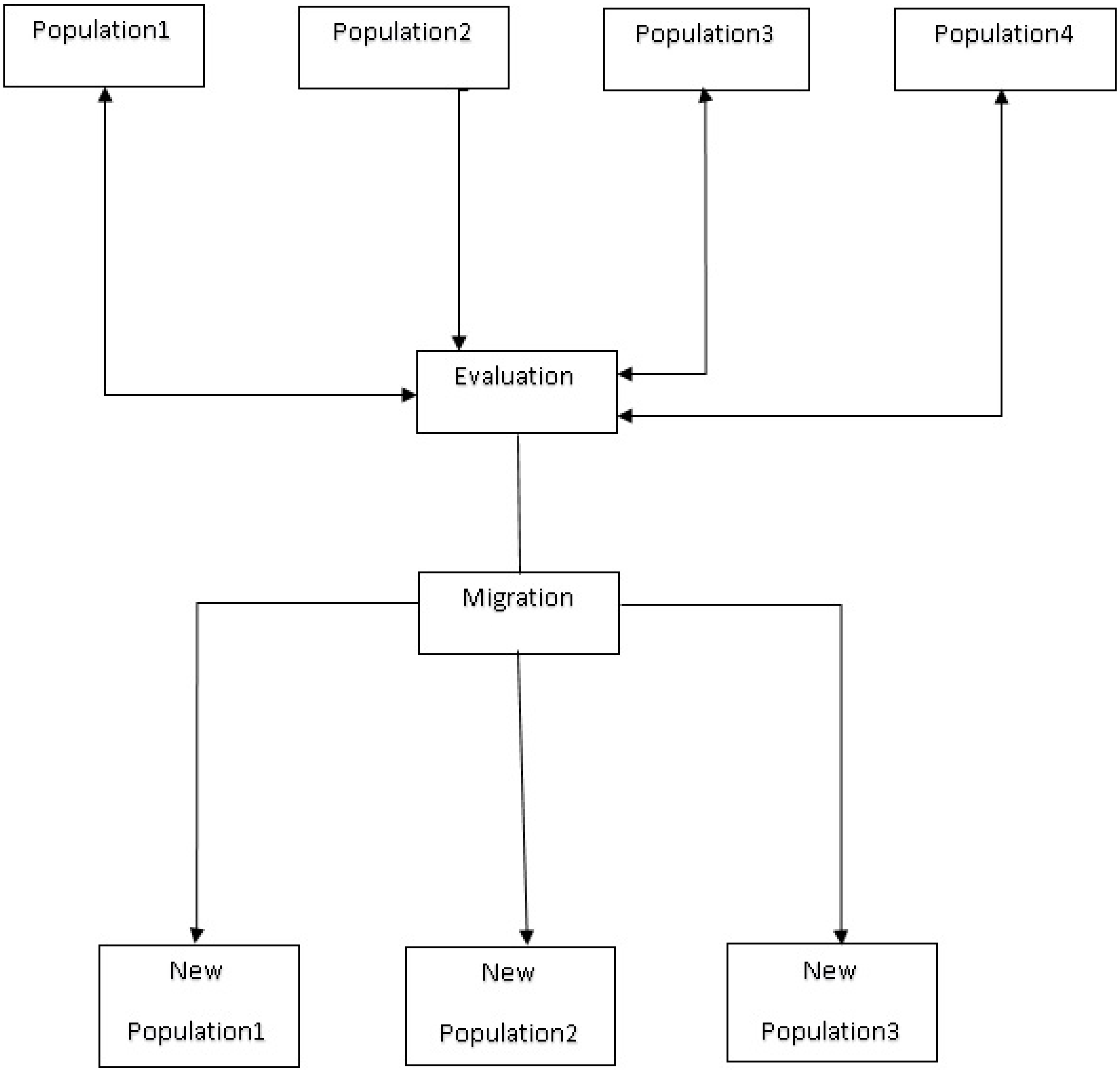


Figure 12: Migration in D-MPCA

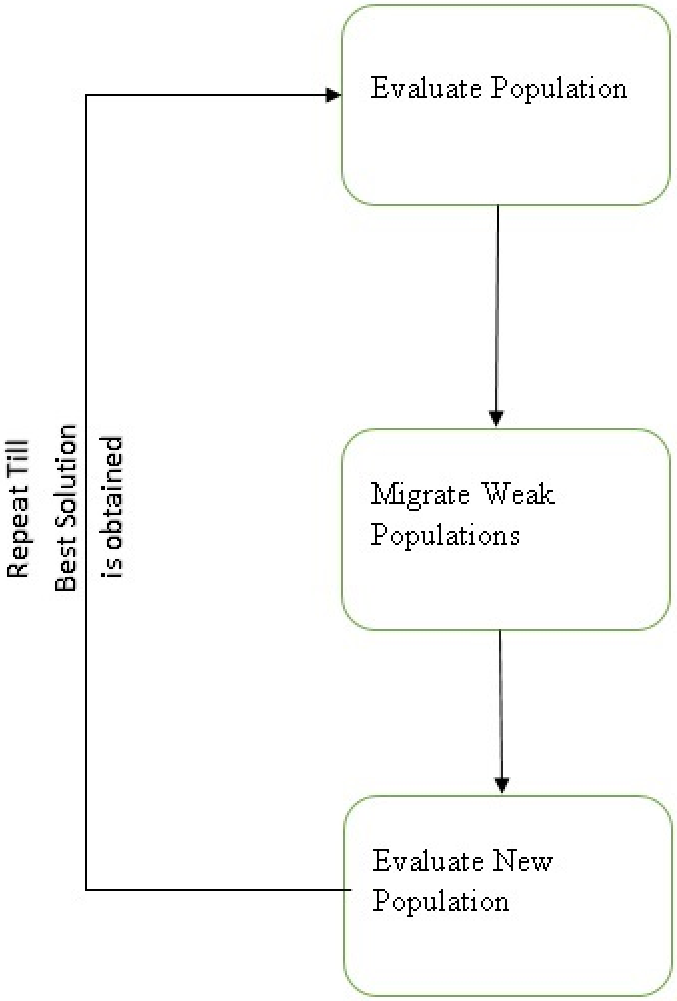


Figure 13: Evaluation of Each Individual in the population and migration

5.3.2 Pseudo Code

Algorithm 2 D-MPCA algorithm

```
PROCEDURE: MPCA NETWORK INPUT: Dominance Strategy and Algorithm Parameters
OUTPUT: A graph for different Dominance strategies and different populations
Randomly generate population f(P)
Assign Global belief space
Divide f(P)
Generate SubPopulations P1,P2,P3,P4,P5.
FOR(each subpopulation)
Evaluate all the individuals
Calculate the weights of all the individuals
FOR(Select Subpopulation)
Migrate individuals based on weak weights
Observe effect of Dominance
Update local belief space
Update Global belief space
END
IF(Iteration No = all subpopulations)
END
END
Output the graph of dominance implemented in the subpopulations
```

5.4 Belief Space of D-MPCA

Every sub-population has its own CA and its own belief space that is updated after each iteration by using the acceptance function in the local CA. The best individual from the sub-population is selected and this particular individual is used to update both the local belief space (CA) and global belief space (D-MPCA). The belief space extracts the normative knowledge of the population and uses this knowledge to evaluate the individuals in the sub-populations. This knowledge is shared among other sub-populations by the global belief space, by sharing the knowledge the populations help each other improve their overall knowledge. Normative knowledge records the fitness values or the desirable value range for the individuals in the sub-populations. This knowledge is very useful in selecting the individuals for migration among different sub-populations. In Figure 13 we explain the evaluation of the individuals in our network of populations.

5.5 Population Space in D-MPCA

The population space component in our D-MPCA is designed based on the genetic algorithms, a group of individuals with fitness values. We use crossover in our evolutionary programming and our individuals are selected randomly to interact with each other and improve their knowledge, this can be seen over time by the change in the fitness values of the individuals in both the sub-population and the whole population. The individuals in the sub-populations consist of only the dimensions of their corresponding sub-populations hence they are considered as partial solutions. To evaluate the partial solution the parameters coming from the belief space are utilized. The individuals are evaluated using a numerical optimization function (CEC 2010). Different benchmark functions were used like the Sphere function, Rastrigin's function and Rosenbrock's function. These functions are some of the standardized benchmarks in the field of evolutionary computations. The functions are explained in detail in the next chapters.

6 Experimental Setup

In this chapter we explain about the experimental setup, parameters and the benchmark functions in detail.

6.1 Description

In this section we explain in detail about our experimental setup and the cec benchmark functions. In our approach we first need to initialize the population space and generate the artificial populations. After generating the population a global belief space is to be assigned to the whole population. This global belief space store the normative knowledge of all the individuals in the population.

OBJECTIVE: Our main objective in this thesis work is to analyze if by implementing dominance in evolutionary algorithms like multi population cultural algorithms can we solve the knowledge migration issue.

Dominance was previously used in genetic algorithms to solve different problems like preserving the variability of the genes and optimization of dynamic environments, but it was never used to solve the knowledge migration issue in evolutionary algorithms. We tested dominance in MPCA by modeling our method so that we could analyze the effect of dominance in the knowledge migration. In our network, the migration of individuals is done in two ways i.e. the method in which there is no effect of dominance and the other where the dominance feature is used. We compare both the processes in our work.

There were different sets of parameters in our experimental setup. Population Size: Two different sets of populations were used a population of size 1000 and another of size 100. We limited our population size to these two sizes but varied the number of sub populations from 5 sub-populations to 20 sub-populations. The population is generated randomly using a random generation function in matlab after generating the individuals a global beliefspace is assigned to the population and later divided into sub-populations. For evaluating the individuals in the populations we used different benchmark functions. The experiments were conducted by varying the following parameters.

1. Population size: 1000 or 100
2. Number of Sub-populations: 5 and 20
3. Benchmark functions: Sphere Function, Rastrigin's Function and Rosenbrock's Function

The experiments were run only for 10 iterations for each set of parameters and the average values were compared for both MPCA and D-MPCA. Separate graph were plotted for each set of experiments.

6.2 Benchmark Functions

In the past few years many different novel optimization problems like niching algorithms, multi-objective optimization problems, constrained optimization problems and so on were addressed. Much of the research work on the single objective algorithms impact the development of these above mentioned optimization branches. In the last few years numerous kinds of novel algorithms were developed to solve the real-parameter optimization problems. The first set of benchmark functions were initiated in the year 2005 and in the past 10 years lots of research work was accomplished based on these standard benchmark functions.

The benchmark functions were divided into different sections i.e. uni modal functions, basic multi-modal functions and composition functions. Each set of functions consists of some functions. The sphere function belongs to uni modal function, rosenbrock's function and rastrigin's function belong to basic multi modal functions. There are many different functions but as part of our thesis we considered only these three functions. There are also other functions like the elliptic function, schwefel's problem, ackley's function and so on. In the next section we will be explaining the following benchmark functions:

1. Sphere Function
2. Rastrigin's Function
3. Rosenbrock's Function

6.3 Sphere Function

The sphere function is single objective optimization and a basic function used for optimizing large-scale benchmark problems.The general formula for sphere function is shown in figure 12.x = [-100,100]. The range of the function is [-100,100]. This formula was re framed to fit our D-MPCA algorithm. Here D is the size of the population i.e. 100 and 1000. X is the fitness value of each individual in D-MPCA. Based on this function the fitness values were calculated and the dominance strategy is tested.In the Figure 14 the graph plots a 3d map for 2d values of the function.

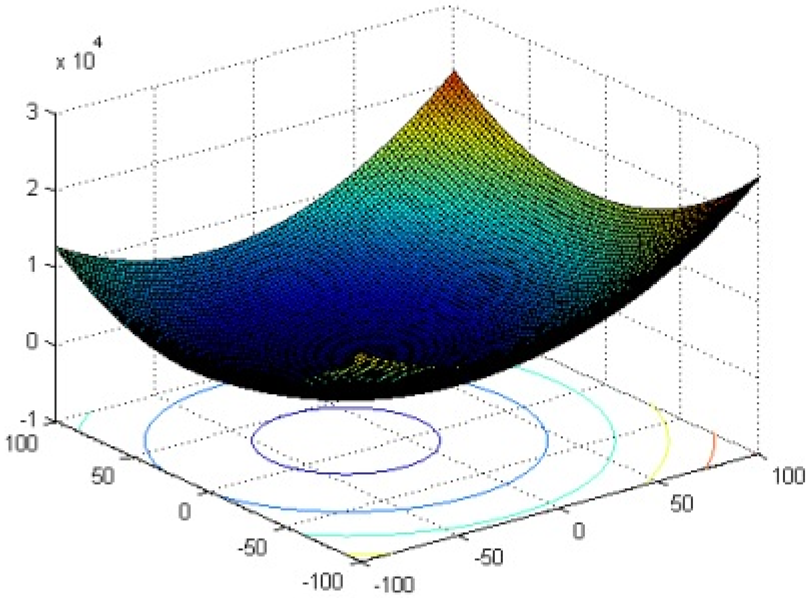


Figure 14: Sphere Function 3-D map for 2-D function

$$F_{sphere}(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$$

Figure 15: Sphere Function

6.4 Rastrigin's Function

This is a multi-modal function used to prove the performance of different optimization problems.

The formula for rastrigin's function is shown in Figure 16.

$$F_{rastrigin}(\mathbf{x}) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$$

Figure 16: Rastrigin's Function

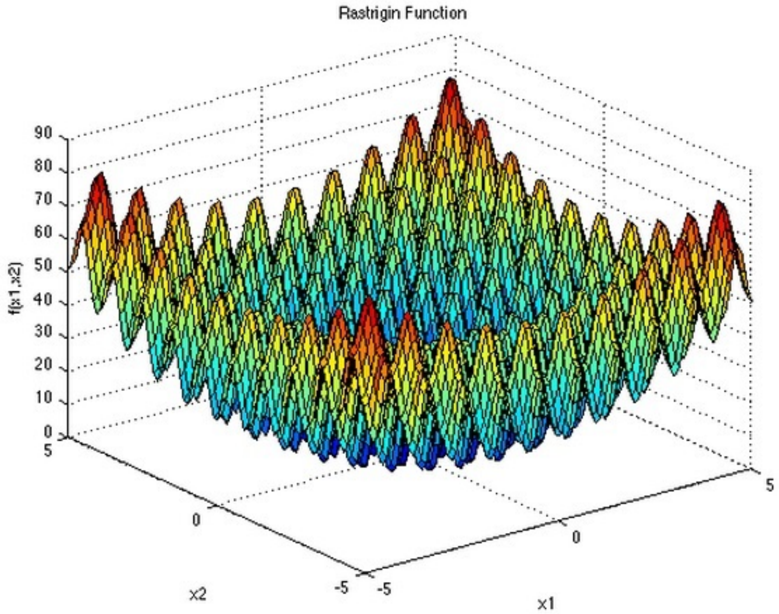


Figure 17: Rastrigin's function 3-D map for 2-D function

X = [-5.12, 5.12]

D is the population size which is 100 and 1000. The range of the function is [-5.12, 5.12]. The experiments were run for rastrigin's function and the values of our algorithm are compared with that of the generalized multi-population cultural algorithm. The Figure 17 shows a graph of the rastrigin's function. This graph consists of the generalized rastrigin's function and the values of the graph are based on the CEC 2013 benchmark functions.

$$F_{rosenbrock}(\mathbf{x}) = \sum_{i=1}^{D-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right]$$

Figure 18: Rosenbrock's Function

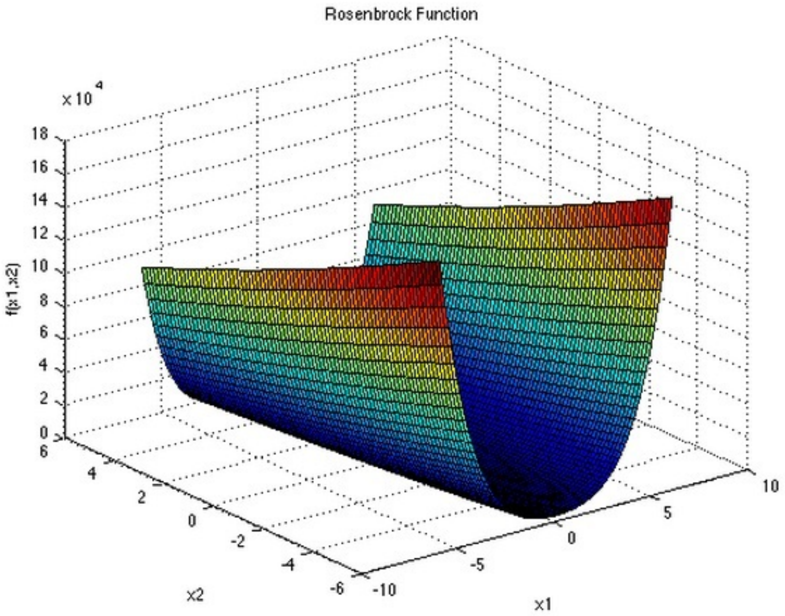


Figure 19: Rosenbrock's function 3-D map for 2-D function

6.5 Rosenbrock's Function

Rosenbrock's function is a multi-modal function and its generalized formula is specified in Figure 18.X= [-5, 5]. D is the population size which is 100 and 1000. The range of the function is [-5, 5]. This was another function that we used in our work. We evaluated the fitness values of our modal with the generalized MPCA and compared the results by using rosenbrock's function.

In the above Figure 19 generalized graph of the rosenbrock's function is plotted.

7 Results and Discussion

In this chapter we explain in detail about the performance of our D-MPCA model by comparing our results with that of the generalized MPCA. This chapter consists of a results and the detailed explanation of all the functions sphere function, generalized Rosenbrock's Function and Rastrigin's Function.

7.1 Comparison of MPCA and D-MPCA for Sphere Function

In all the 10 experiments for 5 sub-populations and each consisting of 250 individuals we observed that by migrating the weak individuals among the sub-populations helps improve the fitness values of all the sub-populations. In Table 1 we can see that the D-MPCA performs better that traditional MPCA by 23%. From the values in table 1 we can clearly state that artificial dominance feature helps in improving the efficiency of the algorithm. In experiment 9 we observed that the effect of dominance was more than the other set of experiments. This could be our best case scenario for this particular set of parameters.

In Figure 20 we plot the graph of the values of MPCA and D-MPCA for 5 sub-populations of size 200 each. This graph gives a graphical representation of performance of D-MPCA over traditional MPCA. The x-axis of the graph is the experiment number and the y-axis is the overall fitness value of the population.

Experiment number	Value for MPCA	Value for D-MPCA
1	$1.34437*10^5$	$1.68047*10^5$
2	$1.29017*10^5$	$1.61271*10^5$
3	$1.32817*10^5$	$1.66021*10^5$
4	$1.32870*10^5$	$1.66087*10^5$
5	$1.35177*10^5$	$1.68972*10^5$
6	$1.37251*10^5$	$1.71564*10^5$
7	$1.32951*10^5$	$1.66188*10^5$
8	$1.34836*10^5$	$1.68545*10^5$
9	$1.43430*10^5$	$1.79288*10^5$
10	$1.27672*10^5$	$1.59590*10^5$

Table 1: Population size 1000 and 5 sub-populations

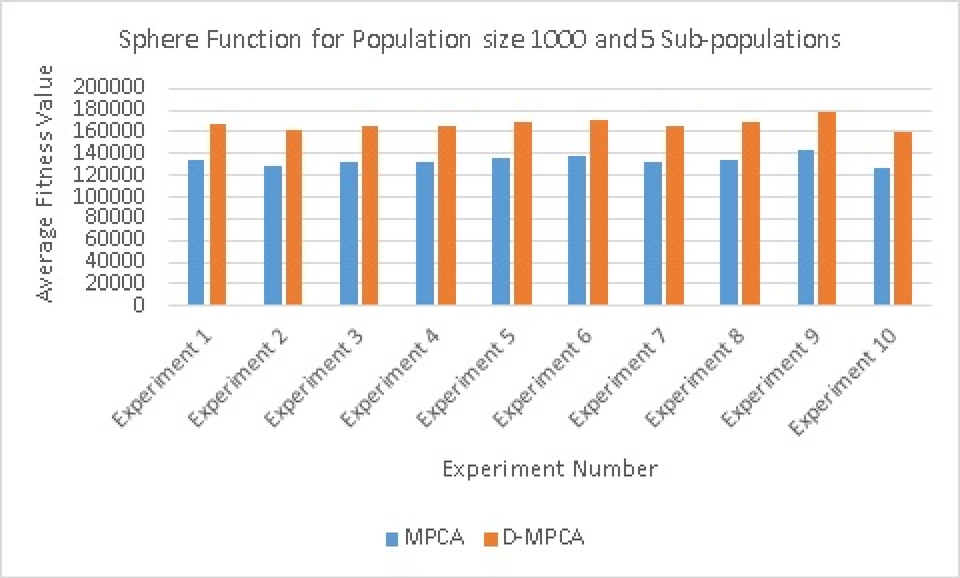


Figure 20: Graph of MPCA vs D-MPCA for a sub-population size of 200 and 5 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	$3.20194*10^5$	$6.40388*10^5$
2	$3.16173*10^5$	$6.32346*10^5$
3	$3.53200*10^5$	$7.06400*10^5$
4	$3.30253*10^5$	$6.60506*10^5$
5	$3.30286*10^5$	$6.60573*10^5$
6	$3.26535*10^5$	$6.53070*10^5$
7	$3.35515*10^5$	$6.71031*10^5$
8	$3.32974*10^5$	$6.65948*10^5$
9	$3.28452*10^5$	$6.56905*10^5$
10	$3.345046*10^5$	$6.90095*10^5$

Table 2: Population size 1000 and 20 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	$1.20884*10^5$	$1.51105*10^5$
2	$1.19500*10^5$	$1.49375*10^5$
3	$1.53207*10^5$	$1.91509*10^5$
4	$1.37434*10^5$	$1.71792*10^5$
5	$1.51291*10^5$	$1.89113*10^5$
6	$1.12892*10^5$	$1.41115*10^5$
7	$1.32369*10^5$	$1.65462*10^5$
8	$1.40732*10^5$	$1.75915*10^5$
9	$1.34662*10^5$	$1.68327*10^5$
10	$1.49032*10^5$	$1.86290*10^5$

Table 3: Population size 100 and 5 sub-populations

In Tables 2, 3 and 4 the values of the experiments run using Sphere Function are displayed. From our experiments we can see that when dominance feature is applied on 20 sub-populations the fitness values had greater effect over 5 sub-populations. In Figures 21,22, 23 we have plotted the graphs of the the values of MPCA and D-MPCA from Tables 2, 3, 4. All these experiments clearly show that dominance helps increase the knowledge migration for Sphere Function.

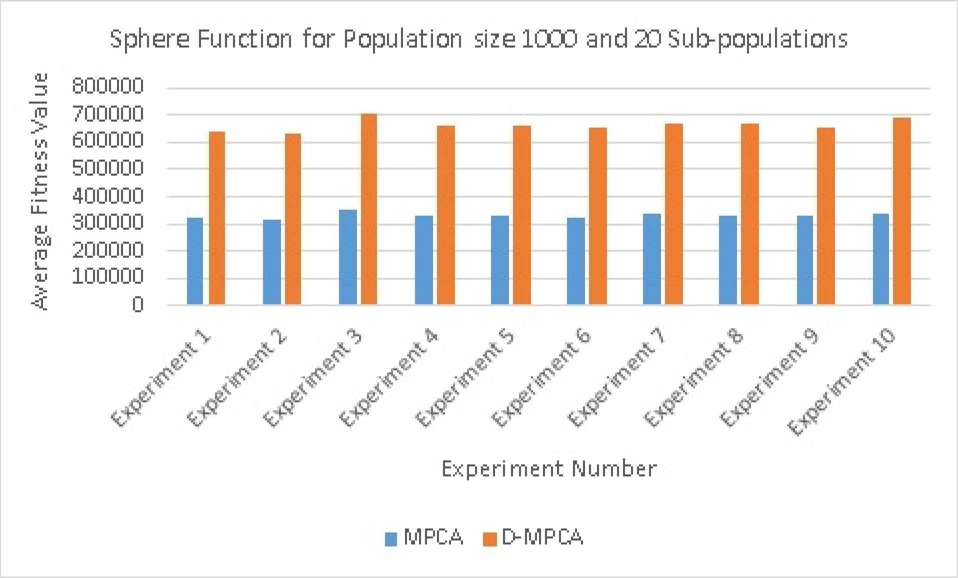


Figure 21: Graph of MPCA vs D-MPCA for a sub-population size of 50 and 20 sub-populations

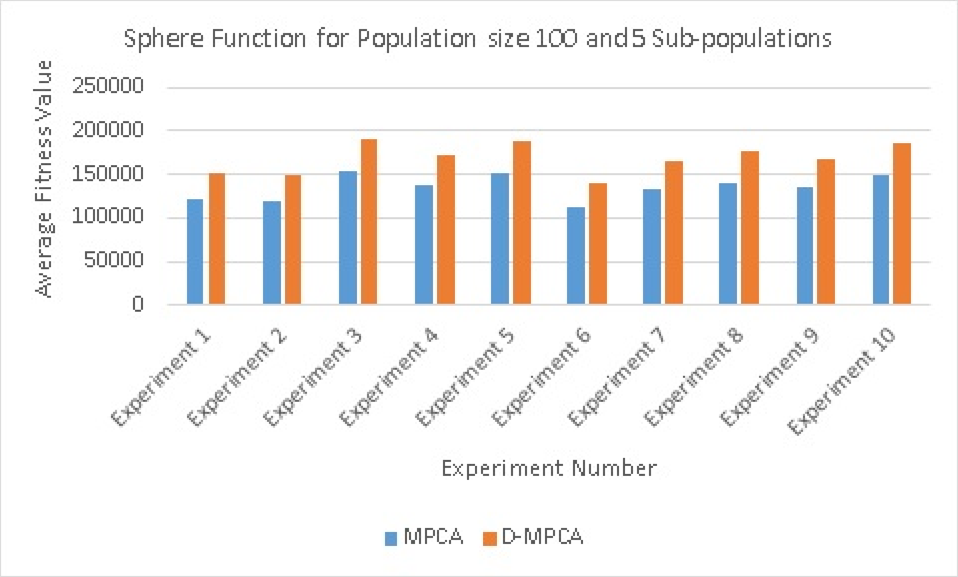


Figure 22: Graph of MPCA vs D-MPCA for a sub-population size of 20 and 5 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	$3.14386*10^5$	$6.28773*10^5$
2	$3.03191*10^5$	$6.06382*10^5$
3	$3.51411*10^5$	$7.02822*10^5$
4	$3.49565*10^5$	$6.99130*10^5$
5	$3.42419*10^5$	$6.84838*10^5$
6	$3.50244*10^5$	$7.00489*10^5$
7	$3.46342*10^5$	$6.92684*10^5$
8	$3.30089*10^5$	$6.60178*10^5$
9	$3.39673*10^5$	$6.79347*10^5$
10	$3.33521*10^5$	$6.67042*10^5$

Table 4: Population size 100 and 20 sub-populations

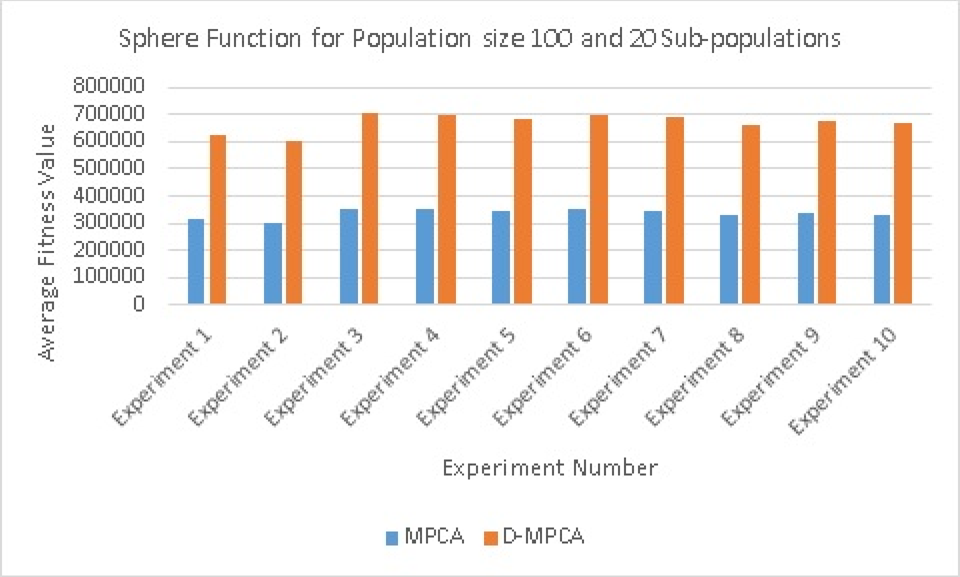


Figure 23: Graph of MPCA vs D-MPCA for a sub-population size of 5 and 20 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	3.84641*10^5	5.24318*10^5
2	3.75193*10^5	5.17803*10^5
3	3.99451*10^5	5.14853*10^5
4	3.73702*10^5	4.94880*10^5
5	4.12614*10^5	5.62231*10^5
6	3.90835*10^5	5.04030*10^5
7	4.02955*10^5	5.27658*10^5
8	4.22923*10^5	5.55076*10^5
9	4.03470*10^5	5.446422*10^5
10	4.00382*10^5	5.25509*10^5

Table 5: Population size 1000 and 5 sub-populations

7.2 Comparison of MPCA and D-MPCA for Rosenbrock's Function

The experiments were conducted for the individuals whose fitness values were calculated using Rosenbrock's Benchmark Function.The range of the fitness values was limited form [-5,5]. We ran the experiments for 4 different sets of parameter and these parameters are explained in the Tables 5, 6, 7, 8 and the graphical representation for these values are represented in Figures 24, 25, 26, 27.

From the above values we can observe that the artificial dominance feature helped improve the knowledge migration to a great extent. In Tables 5, 7 and Figures 25, 27 we can see that for 5 sub-populations the increase due to migration based on dominance was atleast 24% more than that of the traditional migration process i.e. D-MPCA performed better than MPCA. And in Tables 6, 8 and Figures 24, 26 for 20 sub-populations the effect of dominance was even greater the values have almost doubled.

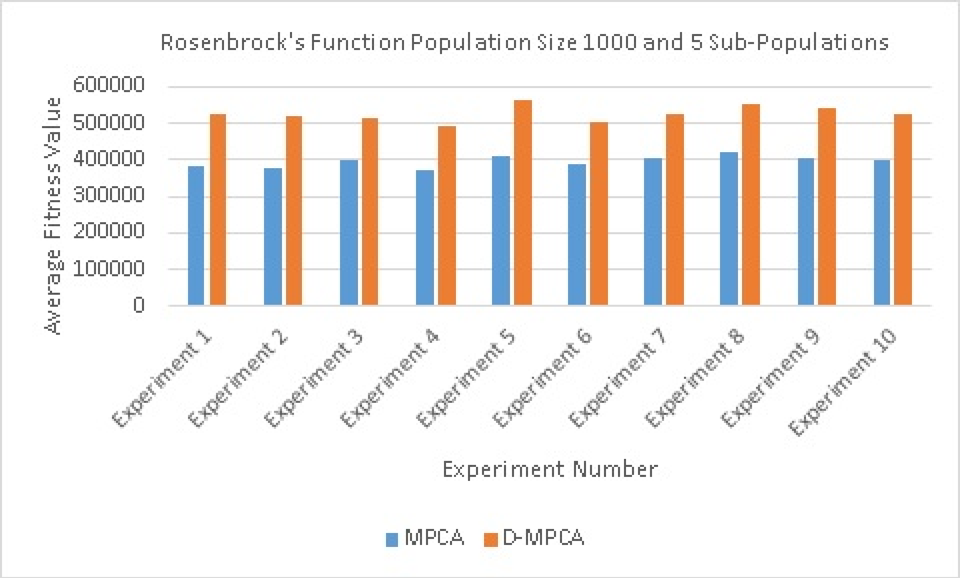


Figure 24: Graph of MPCA vs D-MPCA for a sub-population size of 200 and 5 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	1.216893*10^5	2.566294*10^5
2	1.190108*10^5	2.510954*10^5
3	1.193705*10^5	2.595539*10^5
4	1.249198*10^5	2.641603*10^5
5	1.216339*10^5	2.527103*10^5
6	1.181720*10^5	2.513441*10^5
7	1.300251*10^5	2.735600*10^5
8	1.146717*10^5	2.440188*10^5
9	1.256568*10^5	2.746870*10^5
10	1.223741*10^5	2.568054*10^5

Table 6: Population size 1000 and 20 sub-populations

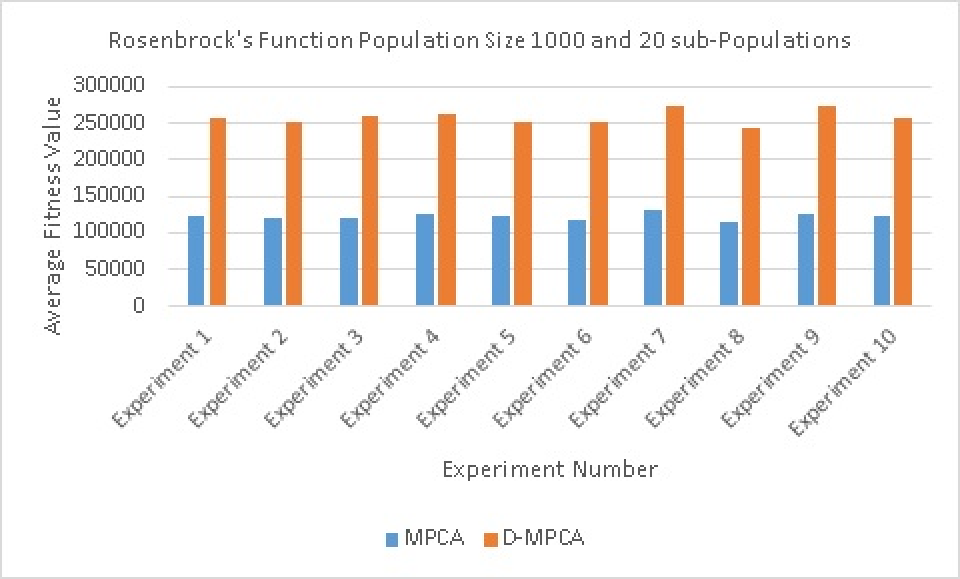


Figure 25: Graph of MPCA vs D-MPCA for a sub-population size of 50 and 20 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	$3.94014*10^5$	$5.10074*10^5$
2	$3.88197*10^5$	$5.40620*10^5$
3	$4.43964*10^5$	$5.77517*10^5$
4	$4.68133*10^5$	$6.12741*10^5$
5	$3.14353*10^5$	$4.49636*10^5$
6	$4.43103*10^5$	$5.70630*10^5$
7	$3.28071*10^5$	$4.04024*10^5$
8	$3.07025*10^5$	$4.05416*10^5$
9	$4.44588*10^5$	$5.99872*10^5$
10	$5.08151*10^5$	$6.22268*10^5$

Table 7: Population size 100 and 5 sub-populations

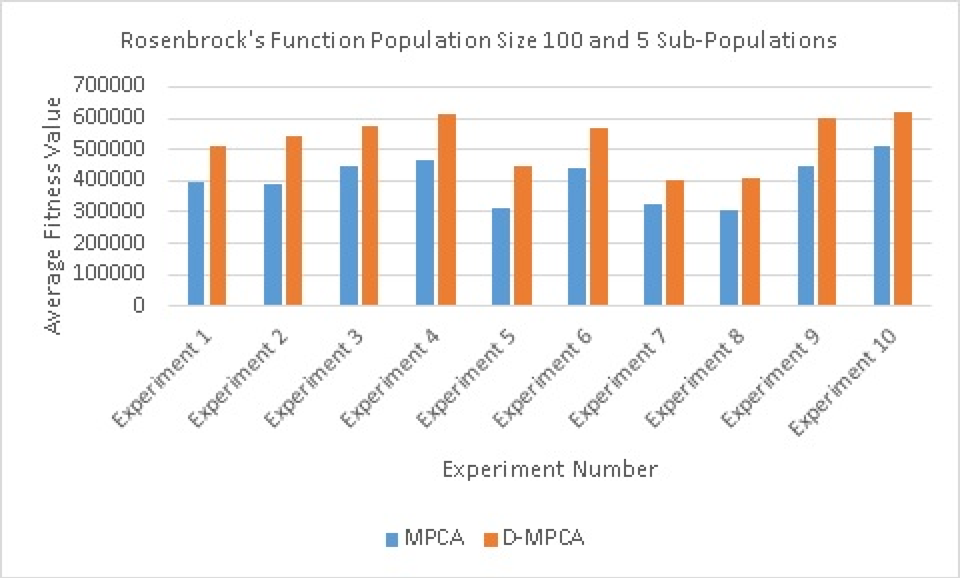


Figure 26: Graph of MPCA vs D-MPCA for a sub-population size of 20 and 5 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	$1.125019*10^5$	$2.452162*10^5$
2	$1.187037*10^5$	$2.496214**10^5$
3	$1.107640*10^5$	$2.347772*10^5$
4	$1.288899*10^5$	$2.716690*10^5$
5	$1.184502*10^5$	$2.431839*10^5$
6	$1.181689*10^5$	$2.435090*10^5$
7	$1.258864*10^5$	$1.66188*10^5$
8	$1.314831*10^5$	$2.803732*10^5$
9	$1.176508*10^5$	$2.482462*10^5$
10	$1.175093*10^5$	$2.458507*10^5$

Table 8: Population size 100 and 20 sub-populations

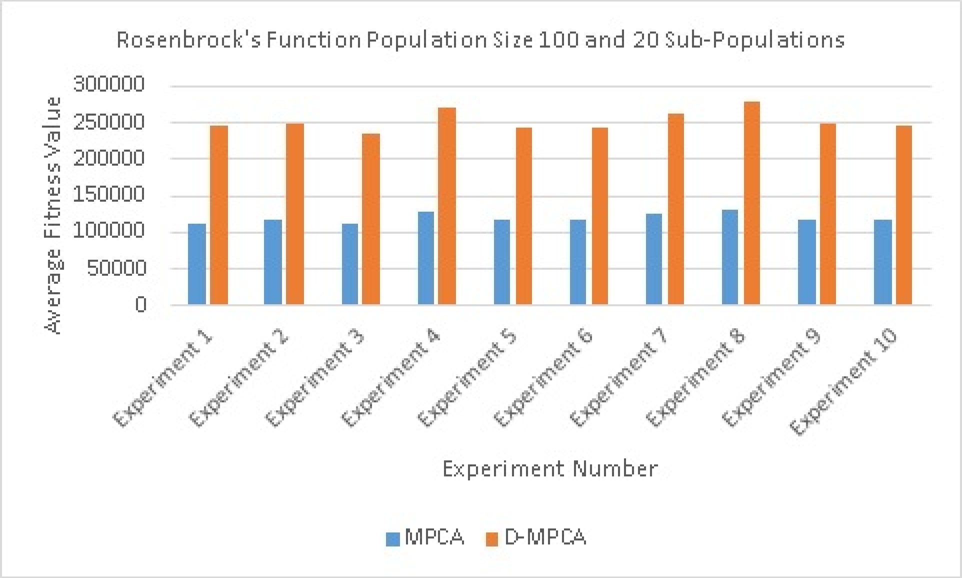


Figure 27: Graph of MPCA vs D-MPCA for a sub-population size of 5 and 20 sub-populations

Experiment number	Value for MPCA	Value for D-MPCA
1	73.492	91.865
2	73.835	92.294
3	74.007	92.508
4	73.734	92.168
5	75.073	93.842
6	74.169	92.712
7	74.0965	92.620
8	73.910	92.388
9	75.360	94.200
10	72.238	90.298

Table 9: Population size 1000 and 5 sub-populations

7.3 Comparison of MPCA and D-MPCA for Rastrigin's Function

In this section MPCA and D-MPCA were compared for Generalized Rastrigin's Function. In this function, the range of the individuals was limited to [-5.12, 5.12]. The weak individuals among all the populations were selected and were migrated to other populations and the effects were displayed in Tables 9, 10, 11, 12. In all these values we can see that by implementing dominance based migration of individuals the knowledge migration can be greately improved. In Figures 28, 29, 30, 31 we plot the graph for the tables and display a graphical representation of the values.

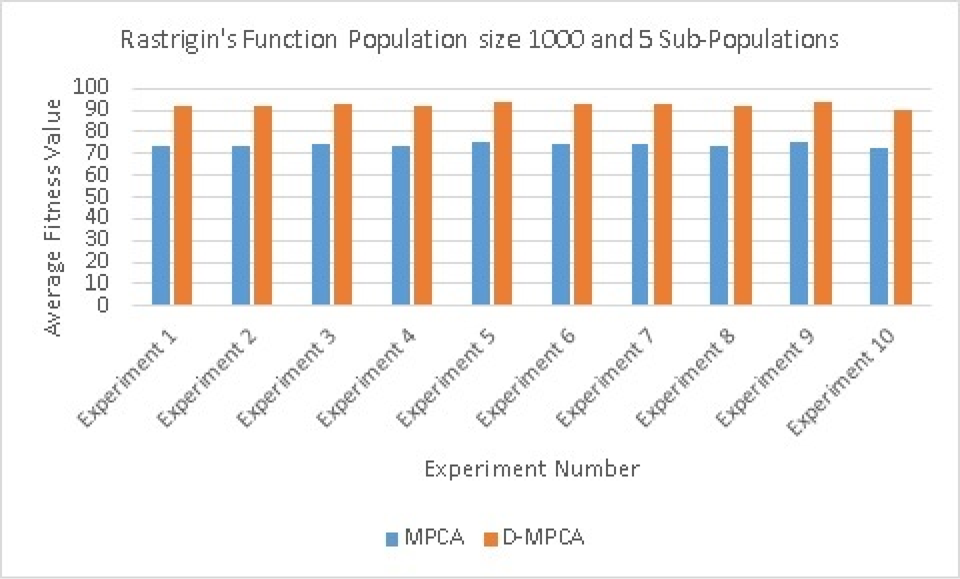


Figure 28: Graph of MPCA vs D-MPCA for a sub-population size of 200 and 5sub-populations

Experiment Number	Value for MPCA	Value for D-MPCA
1	186.5699727	373.1399453
2	189.6775972	379.3551944
3	188.3072278	376.6144556
4	184.2096173	368.4192345
5	192.8014047	385.6028094
6	185.2293175	370.458635
7	185.4651921	370.9303843
8	184.4504853	368.9009707
9	191.2645608	382.5291215
10	187.2957183	374.5914366

Table 10: Population size 1000 and 20 sub-populations

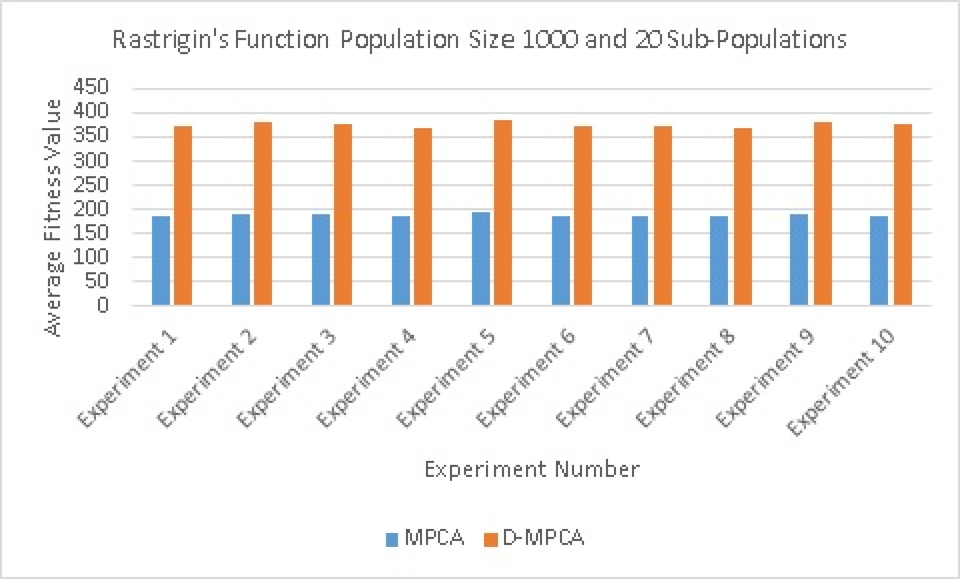


Figure 29: Graph of MPCA vs D-MPCA for a sub-population size of 50 and 20 sub-populations

Experiment Number	Value for MPCA	Value for D-MPCA
1	68.16036263	85.20045329
2	72.76330949	90.95413686
3	78.64111887	98.30139859
4	76.0158811	95.01985138
5	70.73699338	88.42124173
6	70.14220366	87.67775457
7	86.48801431	108.1100179
8	71.8429402	89.80367525
9	75.0360554	93.79506925
10	68.13621905	85.17027381

Table 11: Population size 100 and 5 sub-populations

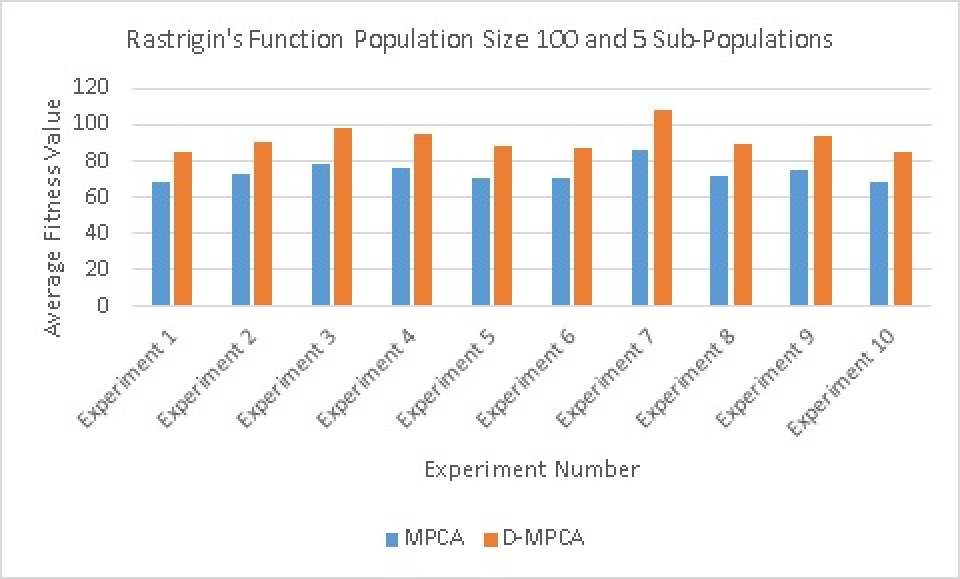


Figure 30: Graph of MPCA vs D-MPCA for Population Size 20 and 5 Sub-Populations

Experiment Number	Value for MPCA	Value for D-MPCA
1	197.5450166	395.0900332
2	183.3237656	366.6475311
3	173.4800728	346.9601456
4	174.1078409	348.2156817
5	201.3208117	402.6416235
6	192.5966796	385.1933593
7	184.9472147	369.8944295
8	192.7753687	385.5507374
9	179.684864	359.3697279
10	181.4368562	362.8737124

Table 12: Population size 100 and 5 sub-populations

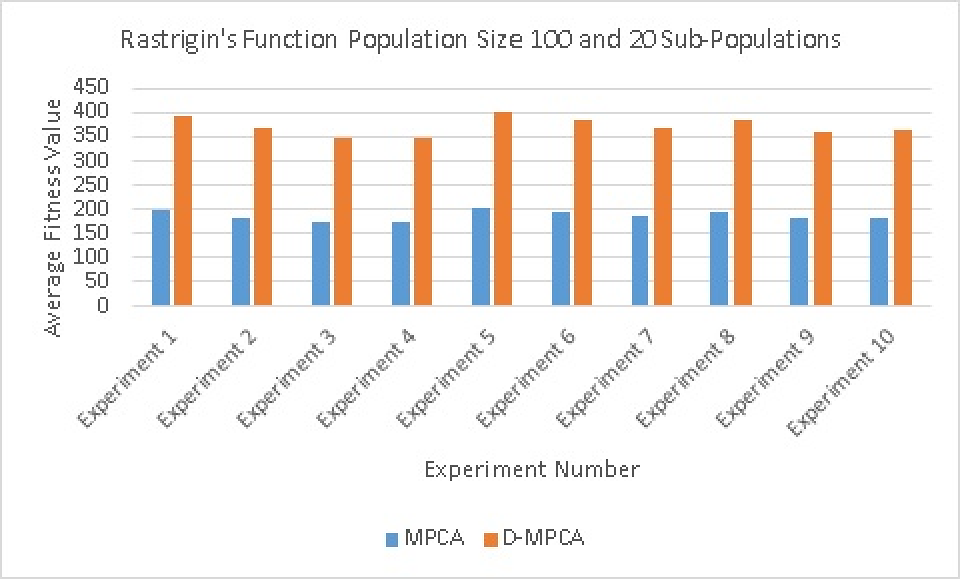


Figure 31: Graph of MPCA vs D-MPCA for Population Size 5 and 20 Sub-Populations

7.4 Summary of Results:

The results for Sphere Function clearly show that dominance helps in improving the knowledge migration for uni-modal functions. We can observe that in all the experiments for sphere function there was an increase of knowledge migration by atleast 23%. But when observed closely at tables 1 and 3 we can see that the perecentage of increase was 23% but when we observe tables 2 and 4 we clearly observe that the rate of increase was up be atleast 50% from this observation we can summarize that for Sphere Function, as the number of sub-populations increase the knowledge migration also increases.

When we observe the tables of Rastrigin's Function which is a multi-modal function the same scenario as of the Sphere function i.e. the knowledge migration was better for large number of sub-populations.

In Rosenbrock's Function though we observed the same amount of increase in the knowledge migration we also observed that the fitness values of the function for 5 sub-populations was way higher that the fitness values of 20 sub-populations. This was not expected in our case. But it occured due to the limitation of the range of the function to [-5,5]. From all our experiments we observed that after running each experiment for 5 iterations we obtained the best migration value for all the experiments and in the rest of the function the value obtained is pretty close to the best result.

8 Conclusion And Future Work

8.1 Conclusion

In this thesis, we proposed a new approach that can be used for solving the knowledge migration issue in multi-population cultural algorithms. We designed our artificial dominance mechanism based on the natural occurring dominance phenomena. We designed our algorithm such that it implements the dominance strategy over the population by migrating the weakest individuals among all the sub-populations. For this purpose all the sub-populations are evaluated. After this is done the individuals are migrated into other sub-populations and the effect of the dominant individuals in the sub-populations is evaluated.

The main goal of our research work was to improve the efficiency of knowledge migration on MPCA.To test the effect of dominance on MPCA we implemented three different benchmark functions on the fitness values of the individuals and compared the values. Our algorithm the D-MPCA (Dominance Multi Population Cultural Algorithm) was tested on two different sizes of populations(1000 and 100). And the number of sub-populations was also varied from 5 to 20 sub-populations.

Based on all our experiments we observed that the migration of individuals among sub-populations using the dominance schema improved the efficiency of the algorithm by at least 20% over traditional process of migration. We tested our algorithms and ran each experiment for 10 iterations and plotted graphs for each set of experiments. The experiments were run on Sphere Function for population sizes 1000 and 100, sub-population sizes 5 and 20; Rastrigin's Function for population size 1000 and 100, sub-population size 5 and 20;Rosenbrock's Function for population sizes 1000 and 100,sub-population sizes 5 and 20 respectively.

In each experiment we observed that dominance improved the efficiency of the algorithm and for large number of sub-populations we observed that the improvement was more than 50%. This was important as the previous works have not used dominance for migrating the individuals and by using dominance better results can be obtained. We compare our algorithm the D-MPCA model with the traditional MPCA approach of migration for all the three functions. We conclude our work by stating that dominance helps in improving the efficiency of knowledge migration in multi-population cultural algorithms.

8.2 Future Work

In this thesis we used only three different benchmark functions so, one of the future works would be to use different sets of benchmark functions. Also we only considered normative knowledge for our belief space, in future other knowledge's can be considered. Also as part of our work we made some conclusions like an individual at any time can belong to only one sub-population, this could be looked at in the future. Also we considered only complete dominance strategy for our thesis so, our work can be extended to other dominance features. All these future works look very promising.

References

[1] Thomas Back, David B Fogel, and Zbigniew Michalewicz. *Handbook of evolutionary computation*. IOP Publishing Ltd., 1997.

[2] Harris Bernstein and Kathleen M Fisher. Dominance in bacteriophage t4d. *Genetics*, 58(3):307, 1968.

[3] Robert Ian Bowers and Emre Sevinç. Preserving variability in sexual multi-agent systems with diploidy and dominance. In *Engineering Societies in the Agents World VI*, pages 184–202. Springer, 2006.

[4] Reinhard Bürger and H Bagheri. Dominance and its evolution. *Encyclopedia of Ecology*, pages 945–952, 2008.

[5] FM Clarke and CG Faulkes. Dominance and queen succession in captive colonies of the eusocial naked mole–rat, heterocephalus glaber. *Proceedings of the Royal Society of London B: Biological Sciences*, 264(1384):993–1000, 1997.

[6] Carlos A Coello Coello and Ricardo Landa Becerra. Evolutionary multiobjective optimization using a cultural algorithm. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pages 6–13. IEEE, 2003.

[7] Carlos Coello Coello, Gary B Lamont, and David A Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems*. Springer Science & Business Media, 2007.

[8] David A Coley. *An introduction to genetic algorithms for scientists and engineers*, volume 1655. World scientific Singapore, 1999.

[9] Jason G Digalakis and Konstantinos G Margaritis. A multipopulation cultural algorithm for the electrical generator scheduling problem. *Mathematics and Computers in Simulation*, 60(3):293–301, 2002.

[10] William H Durham. *Coevolution: Genes, culture, and human diversity.* Stanford University Press, 1992.

[11] Agoston E Eiben and James E Smith. *Introduction to evolutionary computing.* Springer Science & Business Media, 2003.

[12] John Galletly. Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms. *Kybernetes*, 27(8):979–980, 1998.

[13] David E Goldberg and Robert E Smith. Nonstationary function optimization using genetic algorithms with dominance and diploidy. In *ICGA*, pages 59–68, 1987.

[14] Yi Nan Guo, Yuan Yuan Cao, and Dan Dan Liu. Multi-population multi-objective cultural algorithm. In *Advanced Materials Research*, volume 156, pages 52–55. Trans Tech Publ, 2011.

[15] Yi-nan Guo, Jian Cheng, Yuan-yuan Cao, and Yong Lin. A novel multi-population cultural algorithm adopting knowledge migration. *Soft computing*, 15(5):897–905, 2011.

[16] Yi-nan Guo, Shuguo Zhang, Jian Cheng, and Yong Lin. Cooperative interactive cultural algorithms based on dynamic knowledge alliance. *Artificial Intelligence and Computational Intelligence*, pages 204–211, 2011.

[17] Brian Hall et al. *Strickberger's evolution.* Jones & Bartlett Learning, 2008.

[18] Daniel L Hartl, Andrew G Clark, and Andrew G Clark. *Principles of population genetics*, volume 116. Sinauer associates Sunderland, 1997.

[19] Andrew William Hlynka and Ziad Kobti. Knowledge sharing through agent migration with multi-population cultural algorithm. In *The Twenty-Sixth International FLAIRS Conference*, 2013.

[20] David J Hunter. The changing roles of health care personnel in health and health care management. *Social Science & Medicine*, 43(5):799–808, 1996.

[21] Gareth Jones. Genetic and evolutionary algorithms. *Encyclopedia of Computational Chemistry. John Wiley and Sons*, pages 323–330, 1998.

[22] Gareth Jones. Genetic and evolutionary algorithms. *Encyclopedia of Computational Chemistry. John Wiley and Sons*, pages 323–330, 1998.

[23] BS Jung, BW Karnev, and MF Lambert. Benchmark tests of evolutionary algorithms: mathematic evaluation and application to water distribution systems. *Journal of Environmental Informatics*, 7(1):24–35, 2006.

[24] Robert C King, Pamela Mulligan, and William Stansfield. *A dictionary of genetics*. Oxford University Press, 2013.

[25] Ziad Kobti, RG Reynodls, and Tim Kohler. A multi-agent simulation using cultural algorithms: The effect of culture on the resilience of social systems. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 1988–1995. IEEE, 2003.

[26] Ziad Kobti, Robert G Reynolds, and Tim Kohler. Agent-based modeling of cultural change in swarm using cultural algorithms. *Funded by NSF grant BCS-0119981*, 2004.

[27] JJ Liang, BY Qu, PN Suganthan, and Alfredo G Hernández-Díaz. Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University,*

Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, 201212, 2013.

[28] JJ Liang, BY Qu, PN Suganthan, and Alfredo G Hernández-Díaz. Problem definitions and evaluation criteria for the cec 2013 special session on real-parameter optimization. *Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report*, 201212, 2013.

[29] Rammohan Mallipeddi and Ponnuthurai Nagaratnam Suganthan. Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization. *Nanyang Technological University, Singapore*, 2010.

[30] Orel V Gregor Mendel. *the first Geneticist*. Oxford: Oxford University Press, 1996.

[31] Amineh Omidpour, Kamran Alagheband, Babak Nasiri, and Mohammad Reza Meybodi. A new real-valued diploid genetic algorithm for optimization in dynamic environments. In *Intelligent Systems (ICIS), 2014 Iranian Conference on*, pages 1–6. IEEE, 2014.

[32] Stephan Peischl and Reinhard Bürger. Evolution of dominance under frequency-dependent intraspecific competition. *Journal of theoretical Biology*, 251(2):210–226, 2008.

[33] Mohammad Raeesi and Ziad Kobti. Heterogeneous multi-population cultural algorithm. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 292–299. IEEE, 2013.

[34] Robert G Reynolds. An introduction to cultural algorithms. In *Proceedings of the third annual conference on evolutionary programming*, pages 131–139. Singapore, 1994.

[35] Kanta Vekaria and Chris Clack. Haploid genetic programming with dominance. *Departmental Research Note (RN/97/121)*, 1997.

[36] Kerry D Woods. Reciprocal replacement and the maintenance of codominance in a beech-maple forest. *Oikos*, pages 31–39, 1979.

[37] Shengxiang Yang. Learning the dominance in diploid genetic algorithms for changing optimization problems. 2007.

VITA AUCTORIS

NAME: Santosh Upadhyayula

PLACE OF BIRTH: Amalapuram, India

YEAR OF BIRTH: 1992

EDUCATION: Master of Science, Computer Science
University of Windsor, Windsor ON, Canada

PUBLICATIONS

Conference: IEEE ICMLA 2015.

Title: Population Migration Using Dominance in Multi-Population Cultural Algorithm

Status: Accepted for Publication.