Hybrid Algorithms of Microbial Genetic Algorithm and Particle Swarm Optimization for Automatic Learning Groups Composition

A thesis submitted for the degree of Doctor of Philosophy

By

Nicholas Simeon Dienagha

Department of Computer Science

Brunel University London

January, 2019

Supervisors

Dr. Fang Wang
Dr. Simon Taylor

ABSTRACT

Collaboration learning within diverse groups of varying knowledge levels and varying interest levels have been noted to improve learning outcomes. However, composing balanced learning groups with diversity in knowledge level and interest level within the groups while maintaining similarity among the groups is NP-hard and time consuming. The primary aim of this research is to develop an algorithm for automatic composition of balanced learning groups in (MOOCs) with minimal human intervention. The algorithm will assist facilitators in forming balanced learning groups with ease for learners in online classes to benefit from effective collaboration. The research design was experimental research. This design help established comparative experiments of the new algorithm with the particle swarm algorithm as the bench mark algorithm. The findings in the first experiment showed that, the hybrid MGAPSO (Microbial Genetic Algorithm and Particle Swarm Optimization) algorithm outperformed the PSO (Particle Swarm Optimization), an ANOVA (one-way test) showed high significant difference in the mean fitness of the two algorithms (hybrid Microbial Genetic Algorithm and Particle Swarm Optimization and Particle Swarm Optimization). A possible explanation for this might be that the microbial genetic algorithm component tends to re-introduce new particles at every iteration after every genetic operation, thus, introducing diversity in the swarm. In the second experiment, the new adaptive hybrid AMGAPSO (Adaptive hybrid Microbial Genetic Algorithm and Particle Swarm Optimization) outperformed both the PSO (Particle Swarm Optimization) and the hybrid MGAPSO (Microbial Genetic Algorithm and Particle Swarm Optimization) with high significant difference in the mean fitness of the new adaptive hybrid AMGAPSO (Adaptive hybrid Microbial Genetic Algorithm and Particle Swarm Optimization) and the mean fitness of the hybrid MGAPSO (Microbial Genetic Algorithm and Particle Swarm Optimization) and that of the Particle Swarm Optimization. A possible explanation of this finding is that particles stuck in the location in the PSO with their re-initialised new velocity may have searched the solution space in different directions and may have jumped out from their respective locations using the microbial genetic algorithm component, which suggest that the method of hybridisation could have resulted in the improved performance of the adaptive hybrid AMGAPSO algorithm relative to the hybrid MGAPSO. In the third

experiment 500 learners profile data was used in a comparative experiment of the adaptive AMGAPSO (Adaptive Microbial Genetic Algorithm and Particle Swarm Optimization), MGAPSO (Microbial Genetic Algorithm and Particle Swarm Optimization) and PSO (Particle Swarm Optimization), groups formed by the new adaptive AMGAPSO algorithm were analysed; The ANOVA (one way) test results showed no significant difference in the means of all groups for all six learners attributes among all groups formed by the algorithm. The understanding at this point is that, the adaptive hybridization method may have provided a means of avoiding the problem of parameter adjustment and the fitness function derived have contributed to the formation of groups with diversity within the groups while maintaining similarity among the groups formed by the algorithm. The findings answered the aim of the research as the new algorithm outperformed the existing algorithm used in the literature and the groups formed with the new algorithm were balanced in all profile features used in all the groups. The findings show that the new algorithm can be used to form balanced learning group with minimal human intervention. Limitations of the research are that only 500 learners' data was used for the validation of the experiment; this was because data for more students who belong to the same class could not be obtained; ANOVA (One Way Test) was the only statistical tool used in the analysis. In addition, the algorithm developed was evaluated only in terms of the groups form however the effectiveness of the groups and the overall learning improvement achieved by the groups formed by the algorithm was not evaluated.

Nicholas Simeon Dienagha

Acknowledgement

The last four years have been mountains and valleys, but I have come this far with the assistant and support of most persons around me whom I owe my profound gratitude. First and foremost, I would say thank you to my supervisor Dr Fang Wang for her patience and understanding, encouragement and her unwavering support when things were not going right; with sincere gratitude from my heart I say “thank you Fang”. My gratitude also goes to my second supervisor Dr Simon Taylor for all his kind words, encouragement and support all these years. I would also like to extend my gratitude to all staff of the department whom I have learnt so much from during my stay in Brunel University, particularly Professor Martin Shepherd, Dr Pamela Abbot, Dr Stephen Swift and Ela Heaney.
To my friends and colleagues at Brunel university, Tommaso Turchi, Boyce Sigweni, Billy Aminu, Daniel Ayoung, I say thank you all for your encouragement and concern. This section will not be complete without mentioning Dr Evans Osaisai who made it possible for me to come to the UK to study for a PhD degree, thank you Evans.
My profound gratitude also goes to my wife Baratuaipre whom through thick and thin have stayed back home to care for our kids all alone, working hard to give them a good life in my absence and all my children who have missed me all this year as I missed them. This will not be complete without my appreciation to my mother Inara Frank Ogbotobo who calls me every week to be sure am healthy, my Late Father Deputy-Chief Simeon N. Dienagha-Amain and my siblings (Ayebanengiyefa, Kumosuoyo, Ayebatonyo, Ayebaifiemiemi and Kumoayebagha) all of whom have contributed in no less measure to help me through this journey.
I wish to acknowledge the petroleum technology development fund (PTDF) for part funding of my programme

Nicholas Simeon Dienagha

Declaration

I declare that this work has not been submitted anywhere for any degree, it is the result of my own independent research except otherwise stated

Nicholas Simeon Dienagha

Publish List

The following papers have been published (or submitted for publication) as a direct result of the research discussed in this thesis:

Automatic Composition of Dynamic Online Learning Groups Using Particle Swarm Optimization

Nicholas S. Dienagha Extended Abstract

Presented at the Computer Science Doctoral Consortium 2016 Brunel University London, 22 April 2016

Nicholas Simeon Dienagha

Abbreviations

AMGAPSO	Adaptive Hybrid Microbial Genetic Algorithm+Particle Swarm Optimization Algorithm
FOAF	Friend Of A Friend
GA	Genetic Algorithm
GOMS	Goals Operators and Methods Selection - Rules
HGGA-RGCP	Hybrid Grouping, Genetic Algorithm for Reviewer Group Composition Problem
IAPSO	Inertia Adaptive Particle Swarm Optimization
KMPSO	K-Means Particle Swarm Optimization
MGAPSO	Hybrid Microbial Genetic Algorithm + Particle Swarm Optimization Algorithm
MKO	More Knowledgeable Objects
NP-PSO	Non-Parametric Particle Swarm Optimization
PSO-GA	Particle Swarm Optimization-Genetic Algorithm
PSO	Particle Swarm Optimization Algorithm
SSS-APSO	Survival Sub-Swarm Adaptive Particle Swarm Optimization
VPGA	Variable Population-Size Genetic Algorithm
ZPD	Zone of Proximal Development

TABLE OF CONTENTS

ABSTRACT II
ACKNOWLEDGEMENT IV
DECLARATION V
PUBLISH LIST VI
ABBREVIATIONS VII
TABLE OF CONTENTS VIII

CHAPTER 1: INTRODUCTION 1

1.1 INTRODUCTION ..... **ERROR! BOOKMARK NOT DEFINED.**
1.2 AIM AND OBJECTIVES 3
1.3 CONTRIBUTIONS OF THIS RESEARCH 4
1.4 THESIS STRUCTURE 4

CHAPTER 2: LITERATURE REVIEW 6

2.1 INTRODUCTION 6
2.2 SECTION 1 LEARNING 6
2.2.2 Learning Theories/Paradigms 7
2.2.3 Collaborative Learning 10
2.2.4 Summary 11
SECTION 2 GROUP FORMATION METHODOLOGIES 12
2.3.1 Introduction to Group Formation Methodologies 12
2.3.2 Agent Based 14
2.3.3 Case Base Reasoning methodology 14
2.3.4 Semantic Web and Ontology Based 15
2.3.5 Clustering Algorithm 16
2.3.6 Group Technology and Visualization Tools 18
2.3.7 Regression Analysis 19
2.3.8 Mathematical Programming 20
2.3.9 Artificial Intelligence Techniques 21
2.3.9.1 Particle swarm Optimization Algorithm 21
2.3.9.2 Genetic Algorithm 22

2.3.10 Summary of Group Formation Algorithms *30*
2.4 Section 3 Intelligent Algorithms 32
2.4.1 Introduction .... ***Error! Bookmark not defined.***
2.4.2 Particle Swarm Optimization(PSO) *32*
2.4.3 Genetic Algorithm *46*
2.4.4 Microbial Genetic Algorithm *48*
2.4.5 PSO + Genetic Algorithm Hybridization Methods *49*
2.5 Gaps in the Existing Literature *52*

CHAPTER 3: THE HYBRID ALGORITHM OF MICROBIAL GENETIC ALGORITHM AND PARTICLE SWARM OPTIMISATION MGAPSO **54**

3.1 Introduction 54
3.2 Group Formation for Large Collaborative Learning 55
3.3 The Hybrid MGAPSO Algorithm 56
3.4 Particle Representation 62
3.4.2 Velocity Representation *63*
3.4.3 Particle Initialization *63*
3.4.4 Velocity Initialization *64*
3.4.5 Grouping Representation *66*
3.5 Fitness Function Derivations 68
3.6 Mate Selection in MGA 73
3.6.2 Crossover *74*
3.6.3 Mutation *76*
3.7 Experimental Design 77
3.7.2 Purpose of Experiment *79*
3.7.3 Experimental Results *80*
3.8 Validation/ Statistical Analysis 96
3.9 Discussion of Experiment *99*
3.10 Summary and Conclusion of the Chapter 99

CHAPTER 4: ADAPTIVE MGAPSO **101**

4.1 Introduction 101
4.2 Types of Adaptations 102
4.3 AMGAPSO and MGAPSO 104
4.4 Adaptive AMGAPSO Algorithm 104
4.4.1 Overview of AMGAPSO 104

4.4.2 Hybridisation in Adaptive MGAPSO (AMGAPSO) *106*
4.4.3 Mate Selection, Crossover and Mutation *107*
4.4.4 Parameter Adjustment Rules *108*
4.4.6 Adaptive MGAPSO Algorithm with Adaptive hybridisation Type A *111*
4.4.7 Adaptive MGAPSO Algorithm with Adaptive Hybridisation Type B *112*
4.5 EXPERIMENTAL DESIGN 113
4.5.1 Overview of Experiment *113*
4.5.2 Purpose of Experiment *114*
4.5.3 Results of Experiment *115*
4.6 EVALUATION OF EXPERIMENT 125
4.7 DISCUSSION OF THE CHAPTER 127
4.8 CONCLUSION OF CHAPTER 128
CHAPTER 5: EVALUATION AND DISCUSSION 130
5.1 INTRODUCTION 130
5.2 EXPERIMENTAL DESIGN 131
5.2.1 Overview of Experiment *131*
5.2.2 Purpose of Experiment *131*
5.2.3 Data Source *131*
5.2.4 Data Pre-processing *133*
5.2.5 Results of Chapter Experiment *134*
5.3 QUALITY OF GROUPS 136
5.4 DISCUSSION OF THE RESEARCH 139
5.5 THE FINDINGS OF THE STUDY 142
5.6 BENEFITS OF THE RESEARCH 144
5.6.1 Educational Institutions *145*
5.6.2 Computer Science Research Community *145*
5.7 ADDITIONAL / ALTERNATIVE METHODS 146
5.7.1 Algorithm *146*
5.7.2 Data Collection *146*
5.8 CONCLUSION OF CHAPTER 147
CHAPTER 6: CONCLUSION 148
6.1 INTRODUCTION 148
6.2 RESEARCH SUMMARY 148
6.3 IMPORTANCE OF THE STUDY 149

6.4 LIMITATIONS OF THE STUDY..150
6.5 FUTURE WORK..152

REFERENCES...154

APPENDIX A ....ERROR! BOOKMARK NOT DEFINED.

APPENDIX APPROVAL LETTER CLXVI

APPENDIX C..CLXVIII

RESULTS OF THE ANDERSON-DARLING TEST CLXVIII

APPENDIX D CLXXII

LIST OF TABLES

TABLE 2. 1 SHOWING SUMMARY OF PUBLICATIONS ON GROUP COMPOSITION 13
TABLE 2.2 SHOWING SUMMARY OF DISTRIBUTION PAPERS WITH GENETIC ALGORITHM 27
TABLE 2.3 SUMMARY OF ALGORITHMS FOR GROUP FORMATION.. 30
TABLE 2.4 SHOWING METHODS USED IN GROUP FORMATION 31
TABLE 2.5 PSO ADAPTIVE FORMULAE TABLE 44
TABLE 3.1 GBEST FITNESS OBTAINED FOR UNIFORM DISTRIBUTION.. 80
TABLE 3.2 MEAN FITNESS OF ALGORITHM UNIFORM DATA 86
TABLE 3.3 STANDARD DEVIATION OF PSO AND MGAPSO 87
TABLE 3.4 MEAN FITNESS FOR PSO AND MGAPSO FOR DATA WITH GAUSSIAN DISTRIBUTION 93
TABLE 3.5 STANDARD DEVIATIONS FOR PROFILES WITH GAUSSIAN DISTRIBUTION 94
TABLE 3.6 PERCENTAGE INCREASE OF MGAPSO WITH RESPECT TO PSO... 96
TABLE 3.7A SUMMARY OF ANOVA ANALYSIS FOR ALGORITHMS WITH 200 PROFILE WITH GAUSSIAN DISTRIBUTION 97
TABLE 3.8 SUMMARY TABLE OF ANOVA RESULTS 98
TABLE 3.7B SUMMARY OF ANOVA ANALYSIS FOR ALGORITHMS WITH 500 PROFILE WITH GAUSSIAN DISTRIBUTION 98
TABLE 4.1 SHOWING MEAN GBEST FITNESS FOR PSO, MGAPSO AND AMGAPSO 118
TABLE 4.2 SHOWING SD OF PSO, MGAPSO AND AMGAPSO ALGORITHMS 119
TABLE 4.3 SHOWING MEAN FITNESS FOR GAUSSIAN DATA... 124
TABLE 4.4 STANDARD DEVIATION OF GAUSSIAN DISTRIBUTION 125
TABLE 4.5 SUMMARY OF ANOVA RESULTS 126
TABLE 5.1 SHOWING SAMPLE LEARNERS DATA FROM UNIVERSITY 133
TABLE 5.2 RESULT OF THE TEST FOR THE DISTRIBUTION OF THE LEARNERS' PROFILE DATA 134
TABLE 5.3 SUMMARY OF ANOVA-ONE WAY TEST RESULT 137

LIST OF FIGURES

FIGURE 2.1 HARVEY, (2009) MICROBIAL GENETIC ALGORITHM 49
FIGURE 3.1 FLOWCHART OF MGAPSO 1 61
FIGURE 3.2 PARTICLE REPRESENTATION AT INITIALIZATION 65
FIGURE 3.3 GROUPING REPRESENTATION 67
FIGURE 3.4 SHOWING A SINGLE ROW OF A PARTICLE ARRAY 68
FIGURE 3.5 ROW IN GROUPING CORRESPONDING TO ROW OF FIGURE 3.3 IN PARTICLE ARRAY 68
FIGURE 3.6 EXAMPLES OF CROSSOVER OPERATION 75
FIGURE 3.7A GBEST FITNESS OBTAINED FOR 200 STUDENTS AND UNIFORMLY GENERATED PROFILE DATA 80
FIGURE 3.7B EVOLUTION OF PSO AND MGAPSO WITH 200 UNIFORMLY GENERATED DATA 81
FIGURE 3.8 GBEST FITNESS OBTAINED FOR 500 STUDENTS AND UNIFORMLY GENERATED PROFILE DATA 82
FIGURE 3.9 EVOLUTION OF GLOBAL BEST PARTICLE OF PSO AND MGAPSO WITH 500 STUDENTS AND UNIFORMLY GENERATED PROFILES 83
FIGURE 3.10 FITNESS OF PSO AND MGAPSO WITH 1000 UNIFORMLY GENERATE PROFILE 84
FIGURE 3.11 EVOLUTION OF GLOBAL BEST PARTICLE OF PSO AND MGAPSO WITH 1000 UNIFORMLY GENERATED PROFILE 85
FIGURE 3.12 FITNESS WITH VARIOUS SIZES OF PSO AND MGAPSO 86
FIGURE 3.13 FITNESS OF PSO AND MGAPSO FOR 200 PROFILES WITH GAUSSIAN DISTRIBUTION 88
FIGURE 3.14 EVOLUTION OF GLOBAL BEST PARTICLE FOR PSO AND MGAPSO WITH 200 PROFILES DATA WITH GAUSSIAN DISTRIBUTION 89
FIGURE 3.15 FITNESS OF PSO AND MGAPSO WITH 500 PROFILE DATA WITH GAUSSIAN DISTRIBUTION 90
FIGURE 3.16 EVOLUTION OF ALGORITHMS WITH 500 PROFILES WITH GAUSSIAN DISTRIBUTION 91
FIGURE 3.17 FITNESS OF PSO AND MGAPSO WITH 1000 PROFILE DATA WITH GAUSSIAN DISTRIBUTION 92
FIGURE 3.18 EVOLUTION OF GLOBAL BEST PARTICLE OF PSO AND MGAPSO WITH 1000 PROFILE WITH GAUSSIAN DISTRIBUTION 93
FIGURE 3.19 GRAPH OF FITNESS FOR DATA WITH GAUSSIAN DISTRIBUTION FOR PSO AND MGAPSO 94
FIGURE 4.1 SHOWING BOX PLOT FOR PSO, MGAPSO AND AMGAPSO FOR 200 UNIFORMLY GENERATE PROFILE 115
FIGURE 4.2 SHOWING GRAPH FOR PSO, MGAPSO AND AMGAPSO FOR 200 UNIFORMLY GENERATE PROFILE 115
FIGURE 4.3 SHOWING BOX PLOT FOR PSO, MGAPSO AND AMGAPSO FOR 500 UNIFORMLY GENERATE PROFILE 116
FIGURE 4.4 SHOWING GRAPH OF PSO, MGAPSO AND AMGAPSO FOR 500 UNIFORMLY GENERATE PROFILE ..117
FIGURE 4.5 SHOWING BOX PLOT FOR PSO, MGAPSO AND AMGAPSO FOR 1000 UNIFORMLY GENERATE PROFILE 117

FIGURE 4.6 SHOWING GRAPH OF PSO, MGAPSO AND AMGAPSO FOR 1000 UNIFORMLY GENERATE PROFILE 118
FIGURE 4.7 GRAPH SHOWING MEANS OF ALGORITHMS....119
FIGURE 4.8 SHOWING FITNESS OF ALGORITHM WITH 200 PROFILE DATA WITH GAUSSIAN DISTRIBUTION 120
FIGURE 4.9 SHOWING BOX PLOT FOR PSO, MGAPSO AND AMGAPSO FOR 200 GENERATED PROFILE WITH GAUSSIAN DISTRIBUTION 120
FIGURE 4.10 SHOWING FITNESS OF ALGORITHM WITH 500 PROFILE DATA WITH GAUSSIAN DISTRIBUTION....121
FIGURE 4.11 EVOLUTION OF GBEST FOR THREE ALGORITHMS WITH 500 PROFILE DATA WITH GAUSSIAN DISTRIBUTION 122
FIGURE 4.12 SHOWING FITNESS FOR ALGORITHMS WITH 1000 PROFILES WITH UNIFORM DISTRIBUTION 123
FIGURE 4.13 SHOWING EVOLUTION OF ALGORITHMS WITH 1000 PROFILE WITH GAUSSIAN DISTRIBUTION....123
FIGURE 4.14 SHOWING ALGORITHM FITNESS WITH GAUSSIAN DISTRIBUTION 124
FIGURE 5.1 SHOWING FITNESS OF ALGORITHMS WITH REAL LEARNERS PROFILE DATA....135
FIGURE 5.2 SHOWING GROUP OF 5 GENERATE BY ALGORITHM....135
FIGURE 5.3 SHOWING GROUP OF 4 GENERATES BY ALGORITHM 136

Chapter 1: Introduction

1.1 Introduction

The concept of identifying learning groups as a social unit for collaboration has been a consensus idea among educational psychologist and educational sociologist. Although various educational paradigms exist each of which have its definition and theories of learning, the concept of group learning in a social setting have been accepted as a better method of learning by all. Foremost of this claim is Vygotsky (1978) who claimed that knowledge is better constructed during social interaction among peers and as such, emphasised collaborative learning among peers. This study is set towards forming balanced collaborative learning groups from members of a given learning environment.

Motivation

The advent of the Internet technology with its advantages brought people from all over the world closer and it brought among others ecommerce and e-learning. Institutions and learners are taking advantage of the cost and effectiveness of distant learning (Maia, 2008). In addition, the Internet is used to deliver and receive online courses. While some are paid for, others are free, this has changed the direction of distance learning. Typical examples of free online courses are the MOOCs (massive open online courses), which comprises the two types of MOOCs, xMOOCs and cMOOCs, (Fidalgo-Blanco, et.al., 2016). MOOCs and xMOOCs are differentiated based on their pedagogical foundation and method of course delivery courses (Fidalgo-Blanco, et.al., 2016). While xMOOCs pedagogy is based on the use of videos and short texts, cMOOCs is based on the paradigm of constructivism in which case, learners construct knowledge by themselves through their interaction with one another and their interaction with course materials and other materials on the web (Siemens, 2008; Downes, 2012). In cMOOCs, lecturers are encouraged to provide the initial basics and create resources for the course while learners are encouraged to select the contents that suit them. In addition, learners create a network of learners with similar interest where they can exchange ideas, discuss their ideas and resources (Downes, 2012). An additional feature of cMOOCs is that there are diverse learners from different backgrounds who reside in different parts of the world with diverse cultures and ages, all interacting with similar purposes of achieving similar objectives. However, some of these learners lack required prerequisites knowledge (McAuley et.al.,2010). Although cMOOCs is based on the philosophy that knowledge is distributed and learners are encouraged to interact socially and construct knowledge (Vygotsky1978), the platform does not provide for the formation of small learning groups among the participants.

Vygotsky (1978) referred to the process of learners working together in groups to collectively solve problems, tackle a given task or create a product as they interact among themselves as collaborative learning. Collaborative learning not only facilitates co-creation of knowledge, but increases motivation, improves achievement among learners (Johnson & Johnson, 1989) but also encourages interaction and critical thinking which result in better cognitive strategies (Hsu, 2003; Johnson & Johnson, 1998). However, effective collaboration and improved learning outcomes are better achieved with diversity within the group (Prettenatti & Igognimi, 2007) which facilitators consider during learning groups composition particularly when learners have varying interest levels in different topics and varying knowledge levels in different topics.

The question that comes to mind is what is the composition of an effective collaborative learning group. An answer to this question is a group with diversity within the group where different knowledge levels and interest levels exist within the group (Prettenatti & Igognimi, 2007). This means facilitators need to form learning groups with diversity within each group, and for all groups to be effective, the groups have to be similar in their composition. The consideration of this two constraints (diversity within the groups and similarity among the groups) in the group composition makes the group formation hard to obtain especially when the number of increases. In addition, when the number of participants become large it becomes very difficult for facilitators to be able to recognise the strength, knowledge level, interest level of all participants. Manually forming the groups also becomes very time consuming.

Although few studies (Dascalu, et al., 2014; Ullmann et. al., 2015; Moreno et al., 2012; Abnar et al., 2012; Ho et al., 2009; Lin et al., 2010) have attempted answering this question using algorithms, these studies, have examined the group formation problem with limited number of participants. In addition, none have developed a fitness function for the formation of groups which are diverse within the groups while the groups formed are similar.

To tackle the problems as identified above, this study is set out to find out a better way of composing balanced learning groups.

1.2 Aim and Objectives

The primary Aim of this research is to develop efficient intelligent algorithms for automatically creating heterogeneous collaborative learning groups.

To achieve the above stated aim, the objectives below will be accomplished

- To identify the current research status and gap in the literature, including the investigation of the existing algorithms used in learning groups formation.
- To develop algorithms for grouping learners automatically and efficiently.

- To compare the proposed algorithms with existing swarm intelligence algorithm used in the literature using synthetic data and real learners' data
- To evaluate the quality of the groupings formed with the new algorithms

1.3 Contributions of this Research

The contributions of this research are

- An adaptive algorithm for forming balanced learning groups.
- New generic fitness design which is more suitable for collaborative learning and not constrained by number of attributes.
- A static hybrid algorithm of microbial genetic algorithm and particle swarm optimisation algorithm (MGAPSO).
- Proposal of two levels (individual and environmental) of adaptation method in a hybrid algorithm (AMGAPSO).
- Adaptive hybridisation type A for the hybridisation of particle swarm and genetic algorithm adaptively (in chapter 4).
- Adaptive hybridisation type B for the hybridisation of particle swarm and genetic algorithm adaptively (explained in chapter 4).
- A new method of adaptation using inertia weight, social and cognitive learning factors

1.4 Thesis Structure

The overall structure of the thesis is composed of six chapters,

Chapter two of the thesis is divided into three sections. Section one contains learning, learning theories/ paradigms and the advantages of collaborative learning, while section two of the chapter entails a review of existing literature on group formation methodologies which include agent based grouping method, case based reasoning, the use of semantic web and ontology based systems, the use of clustering algorithms, the use of visualization tools and regression analysis. Other methods discussed in chapter two include mathematical programming techniques and the use of artificial intelligence techniques/algorithms. Section three contains analysis of the algorithms used in the existing literature for group formation and the methods of hybridization of particle swarm and genetic algorithms. The chapter is concluded with a discussion of the gap in the literature and summary of the chapter.

In chapter three collaborative learning is discussed. A fitness function for the problem is been derived and the algorithm MGAPSO (Microbial Genetic Algorithm + Particle Swarm Optimization) is been proposed. Chapter three also contains a comparative experiment of the proposed algorithm (MGAPSO) and the PSO (Particle Swarm Optimization). The chapter ends with summary and conclusion of the chapter before which is the experimental design

Chapter four extends the algorithm in chapter three, an adaptive version of the MGAPSO algorithm is proposed and a comparative experiment conducted. Results of the experiments are discussed with hypothesis stated and tested for significance between the adaptive AMGAPSO (hybrid Adaptive Microbial Genetic Algorithm and Particle Swarm Optimization) algorithm and the earlier proposed (MGAPSO) algorithm in chapter three.

Chapter five is divided into two sections evaluation and discussion. Under evaluation, learners’ profile data are collected, grouping is performed with the new adaptive algorithm and groups formed with the adaptive algorithm are analysed. The second section of the chapter contains a discussion of the study.

The concluding chapter of the thesis is chapter six, which contains a summary of the research, importance of the study, limitations of the study and future work.

Chapter 2: Literature Review

2.1 Introduction

The motivation behind this study has been the need for online learners and distance learners to experience the benefit of collaborative learning as is obtained in conventional classrooms however, collaborative learning occurs when learners of varying knowledge level work and learn together to achieve individual and group objectives (Laal & Laal, 2012).

The chapter reviews the existing literature on the formation of learning groups, the various methods used in the last decade are identified; then the trend in the last decade of the literature is analysed; the commonly used algorithms for group formation are then analysed. The gap in the literature is then identified followed by an analysis of the most used algorithms in the literature.

The rest of the chapter is arranged as follows, section 1 contains learning under which collaborative learning and its advantages are discussed while in section 2 the existing methods for group formation are discussed. The chapter is closed with a summary and conclusion of the chapter before which is the section 3 in which an analysis of the commonly used algorithms in the literature, particle swarm optimization algorithm and the genetic algorithm are discussed.

2.2 Section 1 Learning

Researchers in educational psychology have viewed learning in different perspectives leading to the different learning paradigms, with each paradigm giving birth to corresponding learning theories and definition of learning. While some scholars have viewed learning in terms of a change in behaviour (Behaviourism) (Skinner,1972; Shaffer, 2000) others have viewed learning from the view point of cognitivism; constructivism, humanism, design based and 21st century skills. However, the commonly accepted definition of learning is that given by behaviourist paradigm (Skinner,1972; Shaffer, 2000). Learning is the process of the interaction of an individual with another individual or object, helping the individual acquire new knowledge, modifying or reinforcing existing knowledge, behaviour, skills, values or preferences which result in a change in behaviour (Skinner,1972; Shaffer, 2000). Learners could interact with other learners, content material, learning management system, teacher, video etc. directly or indirectly using some other medium.

2.2.2 Learning Theories/Paradigms

The behaviourist perceptive that learners are passive, they start on a clean slate and respond to external environmental stimuli which they are exposed to and their behaviour is shaped due to reinforcement hence learning is a change in behaviour towards the desired direction (Watson, 2013). The behaviourist theories include the classical conditioning theory(Pavlov), the GOMS model (Card, Moran and Newell), the operant conditioning (Skinner), the psychological behaviourism theory (Staats) and the social learning theory (Albert Bandura).

The Cognitivism paradigm uphold that people are rational thus are active participants who think and respond to external stimuli, their change in external behaviour is a result of their thinking and perception. Cognitivist view learners as information processors. Among cognitivist are the attribution theory (Weiner), the cognitive load theory (Sweller), the cognitive theory of multimedia learning (Mayer), elaboration theory (Reigeluth), the expertise theory (Ericsson, Gladwell), the functional context theory (Sticht), the gestalt theory (Von Ehrenfels), the information processing theory, the metacognition theory (Flavell), situated cognition theory(Brown, Collins and

Duguid), stage theory of cognitive development (Piaget) and the theory of empathy, mind-blindness (Premack, Woodruff, Perner and Wimmer).

Constructivist's view the learner as an active participant in the learning process who uses prior knowledge and experience with cultural factors to construct new knowledge. Thus, the learner as information constructor, construct his own subjective representation by linking new knowledge to prior knowledge to create the subjective reality. The constructivism paradigm has the anchored instruction theory (Bransford), the cognitive apprenticeship theory (Collins et. al.), the cognitive dissonance theory (Festinger) and the communities of practice theory (Lave and Wenger) in addition to which are the connectivism theory (Siemens, Downes), the discovery learning theory (Bruner), the ecological theory of development (Bronfenbrenner), the multi-literacies theory (New London group), the semiotics theory (De Saussure, Barthes, Bakhtin), the situated learning theory (Lave), the problem-based learning theory and the social development theory (Vygotsky, 1978).

The design based paradigm discusses how the learning environment should be designed, how, when and why educational theories, design artefacts work in practice. This paradigm has various design theories and model put forward, these are the multimodality theory(Kress), learner centred design (Soloway, Guzdian, Hay), the elaboration theory (Reigeluth), the ARCS model of motivational design (Keller) and the ADDIE model of instructional design.

A contrast of the behaviourism paradigm is the Humanism paradigm; have that, people act on their own freedom with their own intention and values. Learning in humanism is learner centred and should be personalised with the teacher acting as a facilitator. The goal of this paradigm is the creation of self-actualised people through the cooperation among people with the support of the environment; that self-actualisation which people strive for intentionally is based on their perceived values (Huitt, 2001). It is based on the premise that people have this ability and urge to grow and develop to achieve their potentials. Human learning theory educators tend to create learning content and instructions in different ways giving the learners the opportunity to choose their topic of interest from the content and learn the way they like in their own pace (Rogers and Freiberg, 1994). Humanist ideology is that learning is better achieved when the teacher helps the learners, for the learners to create personal meaning and perceive connections between ideas. Also, to encourage learners' inner exploration of the subject matter while they interact with the information and experience provided for self-transformation (Johnson,2012; DeCarvalho,1991.) The humanism paradigm has the ARCS model of motivational design (Keller), the theory of emotional intelligence (Goleman), the theory of experiential learning (Kolb), the theory of flow (Csikszentmihalyi), the Grit theory (Duckworth, Matthews, Kelly and Peterson), the intrinsically motivating instruction (Malone), the Maslows hierarchy of needs (Maslow), the positive psychology PERMA theory (Seligman) and the self-determination theory (Deci and Ryan).

Social development theory (Vygotsky, 1978) explains that social learning precedes internalisation and that learners learn better during social interaction; secondly, that anyone with a better understanding on a subject matter or higher ability to perform a task, who could be a teacher, older adult, a coach, younger person or peer could be referred to as a more knowledgeable other (MKO). MKO could also be a device. Learning occurs within the zone of proximal development (ZPD) which is the distance between the personal abilities of the learner in performing a task and the ability of the learner under guidance or peer collaboration or a MKO (Vygotsky). Social development theory hence postulates that learning be done in peer groups where some learners become MKO and that knowledge could be created during collaboration among peers in their learning groups when learners have diverse knowledge levels as other learners will learn from their MKO in the given topic. This means for effective collaboration, learners in a group should have varying knowledge levels in the different topic of interest. Vygotsky's social development theory was later supported by the Bandura social learning theory, in his theory, Badura explained that people learn from one another by imitating, observing and modelling others and their attitudes. Bandura's social learning theory hence explains human learning in terms of continuous interaction between the learner's cognitive behaviour and his environment which includes his peers.

In Lave's situated learning theory, Lave argued that learning is embedded within a situation and situated in activities, context and culture; that social interaction and collaboration are essential components of situated learning; as learners get involved in a community or group of learners with similar interest they become engaged in the culture of the group and due to their engagement and involvement they learn and later become experts. However, for effective collaboration people must work in groups and develop their small group skills (Johnson and Johnson,1994)

2.2.3 Collaborative Learning

Collaborative learning is an act of a group of peers at various performance levels working and learning together with which each respects the abilities and contributions of their peers, it is anchored on consensus building and creation of knowledge through cooperation among group members towards a desired common objective (Annet, 1997; Panitz, 1996; Laal and Laal, 2012). Collaboration has been extensively studied by various scholars its advantages have been identified.

2.2.3.2 Advantages of Collaborative Learning

In addition to creating a good atmosphere for a variety of assessments techniques, Collaborative learning promotes social, psychological and academic benefits (Laal and Ghodsi; 2012). Socially it helps develop a peer to peer support system among the learners while enhancing understanding among staff and learners and hence helps the development of learning communities. Psychologically collaboration reduces anxiety among learners hence increasing their self-esteem. Learners have the chance to defend their ideas and articulate them while also having the opportunity to question the ideas of their peers. This active engagement create room for better articulation of ideas and creation of conceptual framework (Srinivas, 2011). Collaborative learning encourages the exchange of ideas, increases interest among learners and promotes critical thinking among learners (Totten et. al., 1991; Goshala, 1995) resulting in an increase learners' participation and active involvement in learning task thus increase information retention among cooperating groups (Johnson & Johnson; 1986).

2.2.4 Summary

Although, the various paradigms mentioned have their respective learning theories with each defining learning based on their concept of learning, the social learning theory (Vygotsky, 1978) which states that learners learn better in the cause of their interaction is strongly supported not only by Lava's situated learning theory but also the Bandura social learning theory; the humanism paradigm and the design based paradigm both also highlighted that learning is better when humans interact with their environment. This agreement of the various learning paradigms suggests that collaboration amongst learners and their environment is of tremendous significance in the learning process. In addition, the social learning theory (Vygotsky, 1978) suggests that the composition of collaborative learning groups should contain learners with varying knowledge levels for some learners to be MKO (More Knowledgeable Objects) to others.

However, what is not yet clear is how groups selection can be done such that in each learning group and for each learning topic each group can have

MKO (More Knowledgeable Objects) and all the groups are similar so that all knowledge learners are not placed in a single group.

To examine learning group selection, the next section will review methodologies used in learning group formation in the existing literature.

Section 2 Group Formation Methodologies

2.3.1 Introduction to Group Formation Methodologies

In the previous section, the various paradigms in education were discussed in which most paradigms suggested collaborative learning as a better method of learning thus collaborative learning groups should be formed for learners to experience the benefits of collaboration. However, what is not clear is a mechanism for forming collaborative learning groups. Vygotsky (1978) social learning theory also suggests that MKO (More Knowledgeable Objects) of each learning content be members of each group for all topics while ensuring that the groups are balanced. Balanced groups in this context means the knowledge levels in all the groups for a given learning content are very close or equal in all the groups. This section reviews the methodologies used in group formation in the existing literature.

In the online distance learning scenarios, most instructors know very little about their students making it has become very difficult for instructors to group learners into balanced learning groups for collaboration. However, the last two decades have also witnessed a significant number of researchers proposing various methodologies; Agent based (Khandaker & Soh 2011), case based reasoning (Cocea & Magoulas, 2012); the use of ontology (Isotani & Mizoguchi 2008); clustering (Christodoulopoulos & Papanikolaou; 2007) Group technology approach (Srba and Bielikova, 2015); regression analysis (Mujkanovic et. al., 2012) and algorithms (Wang et. al., 2010; Ani et. al., 2010; Abnar et al, 2012; Moreno et al, 2012;

Pinningoff et. al., 2015; Liu et. al., 2016). In the rest of the section a review of these techniques and methods is reviewed. The remaining subsections of this section are arranged as follows; section 2.2.1 contains agent based grouping while subsection 2.2.2 case based reasoning. Other methods identified in this section are the use of ontology in section 2.2.3, the semantic technique in Section 2.2.4 and clustering algorithms in section 2.2.5. Section 2.2.6 discusses the use of group technology and visualization tools with regression analysis and mathematical programming in sections 2.2.7 and 2.2.8 respectively. The section is concluded with its summary in 2.2. 10 which follows section 2.2.9 where artificial intelligence techniques discussed.

Grouping learners into collaborative learning groups is fundamental in the creation of an environment for collaboration. However, evidence show a growing trend in the group formation problem which may be due to the recognition of the importance of collaboration in learning. Summary Statistics below shows the number of journal publications within years' interval on group composition.

Table 2. 1 Showing Summary of Publications on Group Composition

Summary of Publications on Group Composition

Interval	Number of papers	Percentage
2012 - 2016	23	56.1
2007 -2011	15	36.6
2000 - 2006	3	7.3
Total	41	

The last seven years (2012-2017) have witnessed over 50 percent of published papers in group composition since the turn of the millennium indicating an increase in the interest of researchers in the grouping problem;

Eight basic approaches (agent based, case based reasoning, semantic web and ontology based, clustering algorithms, Group technology and visualization tools, regression analysis, mathematical programming and artificial intelligence techniques) were identified in the existing literature thus the literature is discussed under such headings

2.3.2 Agent Based

VALCAM (Khandaker & Soh, 2011) is multi-agent based collaborative learning environment. VALCAM pairs experts with non-expert students where the members have high social relationship. However, the algorithm for group formation was not clearly stated.

2.3.3 Case Base Reasoning methodology

Further research conducted by (Cocea & Magoulas, 2010) proposed a synergetic approach using Case based reasoning for modelling users' behaviour and clustering for group formation. In the modelling of learners' behaviour, their past behaviour and how they tackle previous task makes up each learner's strategy.

Learners' models were built with their strategies for tasks defined in their models. The strategies-learner's matrix is derived from a resemblance coefficient which is defined based on the objective. The strategies learner's matrix was then used to cluster learners based on similarity of their strategies. This indicates that only learners with similar strategies in the task at hand can be clustered or grouped to form a group. However, what is not clear is how the approach can be used in grouping dissimilar learners since clustering enables the grouping of learners with similar characteristics only.

2.3.4 Semantic Web and Ontology Based

In the ontology based system (Isotani & Mizoguchi, 2008), an ontology which focuses on a framework based on learning theories that facilitate group formation and collaborative learning design is created and Learners' individual goals are determined and matched with collaborative learning sessions based on the ontology. Learners with common goals join a group, after which common goals are set based on their interaction patterns. However, this method can only be used when collaborative learning activities have been designed using same ontology meaning, for every grouping in any specific domain, and the collaborative activities should be specifically designed using the specific domain ontology before grouping can be done. What is required in practice is a technique which can simply be adopted and is not domain specific such that extra work like creating an ontology would not be required.

Similarly, Rubens et. al., (2009) proposed the collaboration formulation model (CAFÉ) for informal collaborative setting which used data about the learner obtained from different sources (their profile data, data about their social activities) which are then mashed-up. Mining of the mashed-up data is done to automatically form groups. Groups are then formed by linking members with learning similar learning materials. However, the authors have not clearly shown the implementation and testing of the model; secondly, the authors did not clearly show an algorithm for the group formation process stated in the research.

The use of semantic web technology and logic programming for grouping was proposed by Ounnas et al, (2009) The grouping was based on DLV solver, an implementation of disjunctive logic programming with an ontology which enabled the system to handle incomplete data. However, in their research only sixty-seven learners were used in the validation of the system. Although the number of learners seemed very good in a normal classroom grouping scenario, it would not be adequate for online learning where large numbers of learners could participate in a module.

A group formation tool composed of a user interface and group generator (Ounnas, et. al., ,2007) which uses FOAF friend of a friend ontology and semantics in the group formation based on satisfying the constraint set by the person forming the groups. The process is being supported by a set of algorithms which allows reasoning on the semantic data provided. However, the algorithms were not clearly stated in the paper and were not intended for learning groups formation. The algorithm cannot be used for learning groups formation because the friends of a friend network may not apply in a learning environment as friends may be having interest in different learning content. Furthermore, the implementation of the algorithm was not clearly demonstrated.

2.3.5 Clustering Algorithm

Clustering is the organisation or partitioning of objects as to finding the natural grouping among objects such that all similar objects are arranged into a group while each group is distinct from others; algorithms used to perform this type of portioning are referred to as clustering algorithms. In this subsection group formation using clustering algorithm is reviewed.

The Fuzzy C-mean clustering algorithm was used to form groups after which a negotiate and exchange process among groups was performed to obtain groups (Christodoulopoulos & Papanikolaou; 2007). The authors point out that the intervention of the teacher in the grouping is unavoidable. Secondly, the number of learners used in the experiment was limited (eighteen). In addition, diversity within the groups may not be easy to achieve because clusters are made of similar participants. Thus, it is not clear how the algorithm could be used for forming learning groups when diversity within the groups is required and the number of learners is increases. Gaudioso and Boticario (2003) used expectation maximization clustering algorithm for group formation.

The K Nearest Neighbour clustering(KNN)- algorithm was used to form groups (Jagadish; 2014) based on Interaction among learners, this algorithm needs records of previous interaction to perform grouping. However, records of these interactions may not be readily available. What is not clear is how the algorithm will perform grouping when records of interactions are not available, it is also not clear how introverts who do not readily interact would be assigned to groups, neither was it clearly stated how grouping could be done in situations where learners do not know one another. Zhou et al; (2016) applied an Improved density clustering algorithm, in this algorithm, initial groups were formed by clustering learners based on the distance matrix between learners Although, the cluster centres were adjusted to optimise the groups using a cluster adjustment parameter, the authors did not clearly state how groups with varying levels of the chosen characteristics could be formed with the algorithm.
A Leader-Follower algorithm used a polynomial(computational) to partition students into groups (Agrawal et. al., 2014). This algorithm could be viewed as a clustering algorithm. In the Count1G algorithm leaders of highest ability pull groups overall ability. However, it is not clear how the algorithm would respond when the number of learners increases.

In the Group technology approach (Srba and Bielikova, 2015) feedback from students' collaboration was used to improve group formation with groups created considering online presence, context, participants' interaction and collaboration. Learners vectors comprising their characteristics vectors were calculated with their comparison values, then the similarity and relevance coefficients were calculated and used to create their compatibility matrix.

The modified rank order clustering (MODROC) algorithm was then used to cluster the group compatibility matrix to form groups (Srba & Bielikova, 2015). The authors pointed out that the method can be applied when the rules for group creation are not known with no known information about the learners' characteristics which is indeed a strength of the method, however, the creation of the group compatibility matrix can be very difficult and computational expensive requiring high computational power when the number of learners increases.

Six different clustering algorithms were identified in this category however, clustering can only be used in cases where group members are similar.

2.3.6 Group Technology and Visualization Tools

Paredes et al (2010) used a supervised method with a visualization tool to form groups. The number of groups are known and for the first group a uniform member is selected and assigned to the first group as pivot of the group after which all other members are compared with the selected member by calculating their Euclidean distance from the selected member using K-means algorithm. Where the distance is greater than the group threshold and greater than the pair threshold the member is assigned to the group. This process is repeated until every member is assigned to a group. Some members may not be assigned to any group and to manage this the algorithm chooses a different member in each group as pivot of the group and tests if the unassigned member can meet the conditions to be assigned to groups. This is done with all groups and in some case assignments may not be done. At the end of this phase incomplete groups are undone and the algorithm tries to reassign members from the incomplete groups to other groups. Groups are then sorted based on their internal Euclidean distance. Exchange of members among groups is done and distances calculated where an improvement is experienced after an exchange, the exchange is maintained. Lastly, all unassigned members are assigned without considering any constraint. This algorithm may be computationally very expensive when the number of participants increase due the amount of computation involved. The authors also concluded that although the algorithm is heuristic it does not look through the entire solution space.

Similarly, Shakir et. al., (2014) proposed a method for building Heterogeneous Ability centred team. However, the algorithm was not clearly stated. The GroupAL algorithm was introduced by Konert et al 2014. In this algorithm, each group is filled by the addition of the best candidate that increase the GPI until the group is full, then the next group is started. However, it is not clearly stated how all the groups could be made similar since the first group could likely be better than the subsequent groups. In addition, testing all candidates before adding a member to group will be computational expensive when the number of learners is large making the algorithm not suitable for grouping large number of learners.

2.3.7 Regression Analysis

Mujkanovic et. al.,(2012) applied multiple linear regression analysis to adaptively update rules for groups formation. Each learner's characteristic was assigned a weight. Learners were shared into two equal groups then members of one group were assigned to members of the other group. The group size was limited to two members to a group. However, this method cannot be used when the number of learners is very large neither is it clearly stated how the groups could be formed if the number of learners in a group need to be more than two. In addition, the method used statistical techniques without any form of intelligence algorithm.

Recent studies conducted in 2014 modelled learning environment as a weighted undirected graph with each learner represented as a node and the relationship between learners as a weighed arc (Kardan & Sadeghi 2014). In their study the similarity between two learners was measured as a mean of their absolute interest levels. Although, a sample size of 32 learners was used in the experiment, what was not investigated was how the algorithm will behave when the sample size is increased. However, the authors in their

2015 paper (Kardan & Sadeghi 2015) critique this paper that it was immature and limited to composing only equal size groups with similar interest.

2.3.8 Mathematical Programming

Further research conducted by Tacadao and Toledo (2015) proposed the use of constraint logic programming in tackling the grouping problem, no algorithm was clearly stated.

The concept of justice was applied by Sadeghi and Kardan (2015) to introduced the justice model. In this model, they used the binary integer programming to solve the group formation problem. The compatibility matrix among learners is computed and sorted in descending order, after which the top m (where m is the group size) compatibilities are retrieved to form groups. However, the use of the model for large number of learners will be very difficult as computing the compatibility of all learners when the number of learners is large is computational expensive. Furthermore, the optimization of the groups was not clearly demonstrated. In addition, the technique does not clearly demonstrate how the groups formed can be of similar haven selected the top m based on their compatibility. The composition of groups from the sorted order means that as the number of groups increases the groups may not satisfy the constrains. Similarly, Alberola et. al., (2016) proposed the use Linear programming to generate groups and Bayesian learning to handle uncertainties in the grouping. Belbin’s role theory was used to identify the roles of members of the team which was used as the grouping criteria, the team adapts using the feedback from teammates to determine the roles of team members and optimize the grouping. Although the authors highlighted that the algorithm handled uncertainties, what is unclear is how the algorithm will optimize the groups without the feedback from teammates. This suggests that the tool could be used to optimise groups formed after other techniques have been used to form the groups.

2.3.9 Artificial Intelligence Techniques

In this subsection, artificial intelligence techniques used in group formation are reviewed, algorithms identified are the particle swarm optimization algorithm, genetic algorithms, tabu search algorithm, hill climbing algorithm and hybrid algorithms. The rest of the subsection contains the use of this algorithms in group formation.

2.3.9.1 Particle swarm Optimization Algorithm

Particle swarm optimization was used to tackle the group formation problem (Ho et al., 2009; Lin et al., 2010; Zheng and Pinkwart, 2014); Ullmann et al 2015) with each set of researchers optimizing their fitness function. Ho et al., (2009) experimented with sixty-one students, with the predetermined parameters (w= 0.8, c1 = 1.49445 and c2 =1.49445). The authors reported the algorithm performed well. However, the number of learners used in the experiment is limited. Thus, it is not clearly stated how the algorithm will perform when the number of learners becomes large as it is obtainable in some online classes like MOOCs. Similarly, Lin et al., (2010) proposed an enhanced particle swarm optimization in the grouping problem however, the enhancement made on the particle swarm algorithm is unclear. In addition, the particle and velocity representation were not clearly explained. The algorithm was evaluated using only simulated data and the distribution of the simulated data used in the experiment was also not stated.

The discrete particle swarm optimization technique was used to compose heterogeneous groups in a similar research by Zheng and Pinkwart (2014). In their implementation of the algorithm, the particle best and global best were updated before the velocity is updated which is followed by the update of the particle position. The sequence of update might result in the neglect of some good particles if these particle positions are arrived at during the last iteration. What is unclear in this algorithm is what will happen in a situation where the new update particle position in the last iteration becomes better than the existing global best position. This means that if any particle position during the last iteration is better than the existing global best it is of no value to the system since update of the global best position is not done after the last iteration. In addition, the authors have not clearly stated how the algorithm will be adaptive as to manage the problem of parameters setting (Shi and Eberhart, 1998). Similarly, Ullmann et. al., (2015) applied particle swarm optimization technique to form learning groups with fixed parameters for the PSO algorithm with an additional parameter being introduced in the velocity equation. The algorithm was evaluated against uniformized grouping. The authors reported that, the algorithm performed better than the uniform method when the number of learners was limited, but experienced stagnation after some iterations. In addition, there was a decline in performance as the number of learners increased resulting in the uniformized method performing better as the number of learners increased. Hence it was concluded that the particle swarm algorithm did perform better than the uniform when the number of learners to be grouped became large. This finding of Ullmann et. al., (2015) calls for an investigation into the use of the particle swarm algorithm.

The literature of particle swarm shows some controversy, while Ullmann et. al., (2015) reported that the uniform method outperformed the particle swarm when the number of learners became large, Zheng and Pinkwart, (2014) had a contrary report in favour of the particle swarm stating that the particle swarm outperformed the uniform method. While this controversy might be attributed to the variations in the particle swarm used by this authors, it is worthy of investigation

2.3.9.2 Genetic Algorithm

Genetic algorithms (Holland, 1975) are a family of population based algorithms inspired by evolution in which potential solutions are encoded as chromosomes which use selection, crossover and mutation operator to generate new candidate solutions (Whitley, 1994). The learning groups formation problem have been tackled using genetic algorithm (Wang et al., 2010; Ani et. al., 2010; Abnar et. al., 2012; Moreno et. al., 2012; Pinningoff et al., 2015; Liu et al 2016) however, modification was made by each set of authors on the algorithm to solve this problem. As is commonly obtained in the genetic algorithm all initial populations were uniformly generated and the fitness of each chromosome or particle was calculated

In the DIANA (Differences in and Non-Differences Among groups) (Wang et. al., 2010) the selection of parent was at uniform and fitness was checked to ascertain an increase in the fitness will follow a crossover operation, if this is found to be true then a crossover is performing else a mutation probability of 0.001 was used to perform a mutation operation. The newly created offspring are used to replace the parent chromosomes in the new population and to start the next iteration the algorithm goes back to the second step of calculating the fitness of each particle or chromosome. It is worthy to note that the aim of the authors in their experiment was targeted at evaluating the performance and behaviour of groups types (heterogeneous and uniformly assigned) in a learning environment. Although the genetic algorithm was used to form the groups the algorithm was not evaluated as the intent of the research was not on the performance of the algorithm but on the behaviour and performance of different type of learning groups. An additional weakness of Wang et al.,'s (2010) research that the number of learners was limited (66). Wang et al.,'s (2010) conclusion thus, was that, groups made up of heterogeneous learners performed better than uniformly assigned learners. A slight modification on the genetic algorithm was done by Ani et. al., (2010) while using the algorithm to group learners. The authors used a crossover range of 75percent to 95 percent with no mutation. They argued that the equality of learners' weight had no effect on the fitness of the chromosome. The likelihood of the swarm to be dominated by a particle is high leading to a homogeneous swarm which may result in the algorithm been trapped in a local optimum. Similarly, Pinningoff et al., (2015) proposed a genetic algorithm with crossover values of 70%, 80%, 90% and 100% for simulated population values ranging from 50 to 1000 forming groups and only forty-eight students' data was used as the largest class size in the experiment. These modifications of seventy-five to ninety-five percent crossover with no mutation by Wang et al., (2010) and 70% to 100% by Pinningoff et al., (2015) provides a very high likelihood of a homogeneous population which may likely not be the optimum.

An evolutionary algorithm based on genetic algorithm was introduced by Abnar et. al., (2012). Unlike other genetic algorithms, the initial population was not obtained using uniform method; rather a greedy search algorithm (modified model based balanced clustering technique) was used to form the initial population. The selection, crossover and mutation operations are then performed to generate the next generation and the best solution is selected which is repeated until termination conditions are met. Although not clearly stated by the authors, this algorithm is a hybrid of greedy search and genetic algorithm. What is unclear in this algorithm is how the initial population will affect the fitness and the performance of the algorithm. One will therefore argue that this evolutionary algorithm introduced by Abnar et. al., (2012) may have no significant difference in terms of performance with genetic algorithm. In addition, the algorithm should have been evaluated against existing algorithm; either the greedy search or the genetic algorithm or both algorithms.

Similarly, Pinningoff et. al., (2015); Liu et. al., (2016); and Wichmann et. al., (2016) proposed the use of genetic algorithm in grouping students. Pinningoff et. al., (2015) used rate crossover values of 70%, 80%, 90% and 100% for simulated population values ranging from 50 to 1000 forming groups while (Liu et al 2016) used 83 learners in their experiment. The crossover percentage rate used by Pinningoff et. al., (2015) may result in a homogeneous swarm where a few particles may dominate the swarm due to the high crossover rate. This may result in the inability of the individual to explore the search space. Although, (Wichmann et al., 2016) proposed the use of genetic algorithm in the group formation the use of the algorithm was not clearly demonstrated in their paper.

Silva et. al., (2011) adopted a generic approach to form groups in a decentralised environment, which was modelled as an undirected graph with each participant being a node on the graph. Groups were formed based on Information exchange between nodes, while constraints must be satisfied before a group can be initiated. However, the authors only demonstrated the grouping of active members, but nothing was said about grouping of inactive members of the community. In addition, the authors did not clearly state an algorithm for the group formation neither was an algorithm used for the optimization of the groups formed. Similarly, Moreno et. al., (2012) proposed a genetic algorithm-based approach in the group formation problem. In their study, they used the selection mechanism of genetic algorithms in the reproduction of the next generation with a limited sample size of forty-five learners. Multiple characteristics of learners were considered in the approach. It is worthy to point out here that the aim of this review is to identify various algorithms used in group formation, excluding the different approaches. This is because the approaches only mimic the algorithms but not implement the algorithms.

The initial groups of Yannibelli and Amandi (2012) were formed by assigning the learners into groups satisfying given constraints based on the role each learner plays in the team. In addition to ensuring that in each team formed, all the different roles as in the Belbin's model (Belbin, 1981, 1983) were present, groups were formed to foster interaction among peers during which process students could create knowledge and acquire skills. The algorithm is a genetic algorithm in which Deterministic crowding evolutionary algorithm (Eiben & Smith, 2007; Goldberg, 2007) was used to select the new population from the old population after the selection, crossover and mutation operations of the genetic algorithm. A notable key issue highlighted by the authors is that the teacher requires a good knowledge of the students. However, there are limits to the teacher's knowledge of the students when the number of learners becomes large (several hundreds). In addition, a limited number of learners was used in the experiment and the evaluation of the algorithm was based on the knowledge of the students' role. It is quite unclear how better the method will perform when the number of learners is increased.

Similarly, Craig et. al., (2010) defined a mathematical model for the grouping problem, an evolutionary algorithm was used for the optimization of the grouping. However, details of the algorithm are not clearly stated in the paper. This makes the evaluation of the algorithm impossible.

Although, ten papers were found to have listed the use of genetic algorithms in tackling the grouping problem, an analysis of the papers showed that in Diana the authors evaluate the performance of the learners formed with the algorithm; there is no clear evidence of evaluation of these algorithms used neither were the algorithms stated. Similarly, two other papers Wichmann et. al., (2016) and Craig et. al., (2010) although, claimed they used genetic algorithm, they had no algorithm stated in their work. Two of the papers found during the search with genetic algorithm were found to have used the genetic algorithm approach, this means the genetic algorithm was not actually used in the forming the group automatically rather the approach was used to manually form this groups. One of the papers found used a hybrid algorithm which comprises a greedy search algorithm and a genetic algorithm in which the initial population was obtained using the greedy search algorithm (Abnar et. al., 2012). This suggests that only four out of the ten papers obtained used genetic algorithm. However, two of these papers used crossover rates greater than or equal to 70 percent. This high crossover percentage could lead to homogeneous individuals in the population which might affect the population after some generations. The outcome is that only two papers and the Diana used the genetic algorithm.

Table 2.2 Showing Summary of Distribution papers with Genetic Algorithm

Algorithm	Number
GA but No Stated Algorithm	2
Genetic Algorithm Approach	2
Genetic Algorithm	2
Genetic algorithm with crossover >= 70% with NO mutation	2
Initialisation with Greedy search +Genetic algorithm	1
Genetic Algorithm with No algorithm stated BUT evaluate performance of Group	1

The table 2.2 shows a summary of the distribution of the papers which had genetic algorithm in the group formation problem. Although, ten papers were cited with genetic algorithm, only five used genetic algorithms out of which two had very high crossover percentages while one evaluated the performance of the learners only and not much was stated about the algorithm in the paper; it is thus worth concluding that only two out of the ten papers used a genetic algorithm and stated the algorithm in their paper.

2.3.9.3 Tabu Search

Tabu search algorithm combined the general and context specific criteria in group formation (Hubscher; 2010). The authors however reported that contradictory preferences were found to occur in the grouping. Notable characteristic of the Tabu search is the use of memory to store all potential moves which the algorithm uses as to check if they Tabu or not, this makes the Tabu search algorithm computationally expensive when many moves will be involved however, only 18 students were used in their experiment.

2.3.9.4 Hybrid Algorithms

Hybrid algorithms refer to algorithms which are composed of two or more algorithms. An example is a hybrid PSO and GA, which means an algorithm created by combining PSO and a GA. In this subsection, hybrid algorithms used in group formation are discussed.

The hybrid grouping genetic algorithm for reviewer group (HGGA_RGCP) was proposed by Chen et. al., (2010), in this algorithm grouping was based on expertise with the algorithm considered varying group sizes. Subsequent generation was selected from the entire population using roulette wheel selection. The population size of 30 was used in the experiment and a member could belong to more than one group. However, this algorithm may not be an appropriate algorithm for grouping learners due to the differences in the requirements.

The particle swarm algorithm was applied on an ontology based e-learning system for building communities such that new members could transits from been trainees to trainers, thus building long term communities (Dascalu et al., 2013). The authors formed groups composed of participants with different background (multidisciplinary) and with similar interest. With a learning paradigm, the system uses the PSO to automatically recommend groups for a user to join based on the profile of the user. However, the authors have only demonstrated the use of the technique and algorithm for limited number of participants; only 25 students were used in the experiment.

Three algorithms were reported by Caetano et. al., (2015) for group formation among which they concluded that the hybrid algorithm performed better than the remaining two algorithms. This resulted in the interest to examine the hybrid algorithm to understand how it functions and to understand how well the algorithm formed grouping. Despite the claims the paper did not clearly explain what algorithm was used to form a hybrid with a genetic algorithm, neither did the paper clearly state the algorithm. This makes the evaluation of the algorithm seemingly impossible for this research. Zheng et. al., (2016) proposed hybrid PSO+GA algorithm for the group formation problem. Integer permutation encoding is used in the representation of the particle (solution) The algorithm contained the update of the particle best and global best particle position. However, the particles in the algorithm were not shown to possess any velocity neither does the algorithm contain the particle position update equation as contained in the particle swarm optimisation algorithm. The only feature of the PSO contained in this algorithm proposed by Zheng et. al., (2016) is the particle best and global best positions defined in the standard genetic algorithm. In addition, the number of learners used in the experiment was limited to160. Four hybrid algorithms were stated however, two of the four algorithms were not stated. Although, described as hybrid algorithm (Dascula et. al., 2013) algorithm, only the features of particle swarm are obtained in the algorithm with the ontology used to class the learners; thus, only one of the four algorithms could be considered a hybrid algorithm.

2.3.9.5 Hill Climbing Algorithms

The Uniform Mutation Hill Climbing algorithm (Russell & Norvig, 1999) was applied in grouping students with an exhaustive search algorithm to find the distance between groups (Lin & Sun, 2000). The performance of the algorithm was evaluated against the uniform method and was discovered to perform better than the uniform method. Similarly, Cavanaugh, et. al., (2004) used a hill climbing approach for initial group formation where swaps were made at the optimization stage and the overall score computed. However, the swap is undone if the score reduces. One would argue that such team created after the swap may likely increase the score after future swap with another team. However, Montana and Davis, (1989) argued that not only does hill climbing algorithm have a great risk of being trapped in local optimum but the algorithm has been known to perform poorly in finding global optimum. Hence, the algorithm may not be very effective for optimum group formation when considering large number of participants.

2.3.10 Summary of Group Formation Algorithms

In this subsection nine major techniques were identified in the published literature. However, two algorithms, particle swarm optimization and genetic algorithms were dominant in the literature.

Table 2.3 Summary of Algorithms for Group formation

Algorithm	**No of papers found**	**No of papers with algorithm used mentioned**	**Remark**
Particle swarm optimization	4	4	
Tabu	1		
Hill Climbing	2		
Genetic algorithm	10	5	Only 2 were evaluated
Hybrid Algorithm	4	2	Two the algorithms used were not clearly defined

The findings of Ullmann et al., (2015) support the need for this research, as the authors reported that the uniform method performance better than the PSO as the number of learners increase however, the PSO has been among the most widely used algorithm in the artificial intelligence community in the group formation. There is hence an urgent need for a better algorithm.

These algorithms (particle swarm optimization algorithm and genetic algorithm) will be analysed in the subsequent section.

Table 2.4 Showing Methods Used in Group Formation

Grouping Method Used	**Number of papers**
Agent based	1
Case based + Clustering	1
Clustering Algorithms (6 different clustering algorithms)	6
Visualization tool	2
Regression Analysis	2
Mathematical programming	3
Particle swarm Optimization Algorithm	4
Genetic algorithm approach+ GA without Algorithm	5
Genetic algorithm with crossover >= 70 %	2
Genetic algorithm	3
Tabu search algorithm	1
Hill climbing algorithm	2
Hybrid Algorithms	2
Hybrid Algorithms (GA and other Algorithm which are not stated in paper	2
Ontology	3
Semantic	1
TOTAL	40

The table 2.4 above shows the detailed distribution of methods used in group formation in the existing literature, the most used methods are genetic algorithm and particle swarm optimization techniques. The figures show genetic algorithm with crossover value greater than or equal to 70 percent.

This might result in the swarm becoming homogeneous, which might affect the search of the algorithm. While five papers used genetic algorithm approach without clearly stated genetic algorithm however, three more papers used genetic algorithm, one of which was initialised with a greedy search algorithm. Although, the authors have stated they used genetic algorithm most of them did not state the algorithms.

Particle swarm optimization algorithm had four papers and was found to be the most used algorithm in the literature.

2.4 Section 3 Intelligent Algorithms

2.4.1 Introduction

The most used algorithms in the group formation problem over the last two decades were the particle swarm optimization algorithm and genetic algorithm although, there are many variations of these algorithm the basics of these algorithms will be examined in the section. The controversy in the report by (Ullmann, et. al., 2015) and (Zheng & Pinkwart, 2014) calls for the investigation into the use of the basic particle swarm optimization algorithm. The intension is to analyse the algorithms to identify their strength and weaknesses and proffer a method of annexing the strength of both algorithms.

2.4.2 Particle Swarm Optimization(PSO)

2.4.2.1. Background of PSO

The particle swarm optimization technique was introduced by Kennedy and Eberhart in 1995. The algorithm mimics the social behaviour of flocks of birds and school of fish during their search for food to solve optimization problems. Particles fly through the search space with each particle keeping track of the best position it has ever had called the pbest of the particle while the overall best particle position ever found as the global best (gbest) position. The particles accelerate towards the particle best (pbest) and the global best (gbest). Acceleration is affected by a uniform number generated for each of the pbest and gbest so far found affecting the particles towards this position.

The change in the particle position is a function of (a.) the particles inertia, (b) the particle's most optimist position (pbest) and (c) the best position experience by the swarm(gbest). The new position of the particle is updated with the new velocity and the previous position (Shi and Eberhart; 1998). The velocity is thus defined as

$$V_{id}^{k+1} = wV_{id}^{k} + C1.rand(Pbest_{id}^{k} - P) + C2.rand(Gbest_{id}^{k} - P)$$

$$P_{id}^{k+1} = P_{id}^{k} + V_{id}^{k+1}$$

Where V_{id}^{k+1} is the velocity of the particle at iteration k+1; P_{id}^{k+1} is the position of particle id at iteration k+1; P_{id}^{k} is the previous position of particle id; $Pbest_{id}^{k}$ is the best position ever achieved by particle P_{id} as at iteration k, $Gbest_{id}^{k}$ is the best position ever achieved by the swarm at iteration k, w is the inertia weight assigned to the previous velocity which is referred to as the learning rate. The social and cognitive constant are C2 and C1 respectively which move the search towards the personal and global best positions respectively while rand is a uniform number uniformly distributed between 0 and 1 which is generated to uniformize the search of the algorithm.

The algorithm has been applied in different domains for solving optimization problems which include among others, engineering, information processing (Kennedy & Eberhart, 1999) voltage control systems (Fukuama and Yoshida, 2001) and lot others. However, this research is not primarily concern with the various application of PSO hence will not review that.

2.4.2.2. The Strength and Weaknesses of PSO

The particle swarm optimization technique has been an importance technique in artificial intelligence community due to its applicability in different domains. This is because it is simple to implement and converges faster (Shayeghi et al 2008). Additional strengths of the algorithm include limited number of parameters, robustness of the algorithm in solving varying kinds of problems, low memory use and the algorithm is easy to combine with other algorithms to form hybrid algorithm for better optimization (Bai, 2010) however, the algorithm is not without its weakness. The algorithm converges prematurely due to the acceleration of all particles towards the gbest of the swarm. With no known mathematical proof for the algorithm, it is not clear how to correctly tune the parameters of the algorithm. Thus, it has been difficult using the algorithm to properly manage scattering and optimization problems (Bai, 2010).

In addition, the performance of the algorithm is problem dependent due to its parameter settings required for each problem set (Premalatha and Natarajan, 2009) resulting in the inability of the algorithm to maintain a balance between local and global search (Montalvo, et. al., 2007) and getting trapped in local optimum (Aghababa, et. al., 2010). The problem dependency of the parameters of the particle swarm algorithm have raised great concern among the research community. Hence, various adaptive mechanisms have emerged. While some authors have focused on adaptive inertia weight others have adopted feedback mechanisms for adaptation. The existing adaptive techniques are further discussed in the next subsection.

2.4.2.3 Self-Adaptive Particle Swarm Optimization

Self-adaptation is a strategy in which particle reconfigure themselves accordingly to suit a problem without user interaction (Van den Bergh & Engelbrecht, 2002) particles could adapt based on changes in their environment, population, individual or component which are often referred to as environmental, individual, population or component adaptation (Wang, et. al., 2013) however, in this context adaptation refers to individual adaptation where each individual particle adapts based on its own properties. When a particle can adapt it is referred at as an adaptive particle.

A major advantage of the particle swarm optimization technique is the limited number of parameters to alter in the algorithm. Adaptation in the algorithm takes the alteration of these parameters, the inertia weight and the acceleration coefficients which are the social and cognitive coefficients of the algorithm. In the particle swarm algorithm, the cognitive coefficient is responsible for the local search of the particle and thus uses the data of the particle while the social considers the properties of the entire swarm. Many authors have thus viewed adaptation in the particle swarm with respect to the adaptation of these parameters in the velocity function of the algorithm, while others have considered the adaptation without altering the parameters which are known as non-parametric adaptation (Beheshiti, et. al., 2015).

In the parametric adaptation the parameters of the algorithm are modified by the algorithm based on the changes to the environment or some change in the individual particle known as environmental and individual adaptation respectively (Wang et. al., 2013). In environmental adaptation, the changes considered are in respect of the entire swarm or the collective change of the swarm behaviour, while in individual adaptation changes are considered with respect to the behaviour of individual particle. Parametric adaptation also referred to as adaptive parameter control, (Hu et al, 2011) is one in which the particle swam algorithm takes two broad categories of adaptation. These are adaptation using inertia weight (Saber, et. al., 2006; Suresh, et. al., 2008; Jiao, et. al., 2008; Arumugan & Rao, 2007; Feng et. al., 2007; Yang et. al., 2007; Panigrahi et. al., 2008; Ghosh et. al., 2010; Rezazzadeh et. al., 2011; Nickabadi, 2011; Hu et. al., 2013; Chatterjee & Scarry, 2014; Zhang et al., 2014; Hu et al., 2015; Tang et. al., 2015; Kiani & Pourtakdoust,

2015) and adaptation using velocity (Carlisle & Dozier, 2000; Yang et. al., 2007; Wang et al., 2015; Pornsing et. al., 2015;). However, some authors have proposed adaptation of a mix of more than a single parameter. Velocity and inertia weight were used for adaptation by Liang et. al., (2015) while Ardizzon et al., (2015) proposed the use of the velocity of the particle and the current position of the particle.

2.4.2.4. Velocity

Researchers have defined the velocity function in different ways and used it for particle adaptation. In this subsection the various methods are discussed.

The velocity update function of the particle swarm algorithm was modified by Yang et al., (2007). The authors introduced the evolution speed factor and the aggregation degree factor as variables of the inertia weight. In this strategy, each particle has its inertia weight which dynamically changes at run time. The function is given as $\omega_i^t = F(h_i^t, s)$ where h_i^t, is the evolutionary speed factor and s is the aggregation degree. In addition, the inertia weight of each particle was individualised with the velocity of the particle decreasing as the iteration increased. However, Carlisle and Dozier (2000) proposed a method by which the particle swarm algorithm could adapt to dynamic environments. In this method the velocity of each particle which is triggered by change in the environment is periodically reset at given iterations. The resetting period for the algorithm is problem and user dependent. Hence, it is unclear how the particles will adapt based on their individual fitness rather than based on the entire swarm, neither is it clear how a general resetting mechanism could be adapted. In contrast to the use of velocity in the particle swarm adaptation other authors have used the fitness function of the particles.

2.4.2.5. Fitness

The cognitive and social components of the particle swarm algorithm were redefined and used for adaptation, in this method the fitness of particle best minus the fitness of the particle was introduced as a local adaptive coefficient while the fitness of global best minus fitness of the particle as global adaptive coefficients (Aghababa, et. al; 2010). However, it is unclear if these coefficients improved the algorithm as the modified algorithm was evaluated against a uniform method and not evaluated against the original PSO algorithm. Similarly, Xie et. al., (2012) proposed adaptation at the individual level of the particles in which the difference in the fitness of the ith particle and the global-best was used to identify inactive particle which were replaced to maintain diversity. However, identification of the inactive particles is complex as it involved the introduction of an error function which will be predetermined and problem dependent. It is also not clearly stated how the error function can be determined based on the data distribution and the problem domain. Another feature for adaptation used by other researchers is how good a particle performed; this is referred to as the success rate of the particle.

2.4.2.6. Success Rate

The success of the particle has been used to determine the adaptation of the swarm. In this vein, Wang et. al., (2015), proposed the Self adaptive strategy Particle Swarm Optimization with various strategies (mCL updating, Henon mutation, gbest updating, DbV updating and global neighbourhood search) in which each strategy with its update function changed its probability based on its previous success recorded without additional control parameters. The probabilities of the strategies are reinitialized in accordance with a model to introduce diversity. However, the authors pointed out that the algorithm is problem dependent. Similarly, Nickabadi et. al., (2011) used the percentage success rate of the swarm to determine if the swarm was slowly moving towards the optimum or oscillating within the search space which was used to alter the inertia weight of the particles in the swarm. The worst particles in the swarm are replaced with a mutated version of the best particles. The drawback of this mechanism of adaptation is that there is no individualized adaptation for the particles. Rather, the particles all tend to move in a single direction. In addition, it is not clear how this method will enable the particle to search in varying directions to prevent all particles from been trapped in a local optimum. The altering of the inertia weight as a form of adaptation have been considered by other authors.

2.4.2.7. Inertia weight

The fuzzy adaptive version of the algorithm was introduced by Shi and Eberthart (2001) in which the algorithm adapted dynamically to the environment, population, component and individual level of the particle (Shi 2000). Similarly, a fuzzy adaptive particle swarm optimization (FAPSO) technique was introduced by Saber, et. al., (2006), in this technique, the inertia weight of the swarm is adjusted using some fuzzy rules based on the diversity of the fitness of the swarm. The fitness of the particle at the current location and the current inertia weight were used to determine the inertia weight for the next iteration using IF/THEN rules. In contrast Arumugam and Rao, (2007) adaptively adjusted the inertia weight of the swarm and the acceleration coefficient in the velocity update function using the global best and average particle best which they introduced into the algorithm. Although the inertia weight of the swarm changed based on the performance of the particles in the swarm, it is unclear how the algorithm will allow for individualized adaptation, rather, the algorithm allows for entire swarm adaptation. A strategy of chaotic descending inertia weight, resulting in a linear descending inertia weight and uniform inertia weight concept was introduced in the PSO (Feng et al, 2007). Similarly, Jiao et. al., (2008) introduced a dynamic weight which decreased with increase in iteration while (Panigrahi et al., 2008) introduced an adaptive particle swarm technique, a method in which the inertia weight of a particle is a function of its fitness and rank in the population. In the study of (Panigrahi et al., 2008) non-performing particles are re-initialised and put back into the population after some generations to generate a new population. The maximum and minimum values for the velocity and inertia weight are predetermined. However, it is unclear how to determine the maximum and minimum values of the velocity based on the problem. In addition, the authors did not clearly state how the number of generations before re-initialisation can be determined based on the problem.

A new coefficient was introduced into the position update equation to adaptively alter the inertia weight, which decreased as the particle moved away from the currently found global best (Suresh et. al., 2008). However, it is not clearly stated how diversity will be introduced into the swarm since all particles are attracted to the current global best position. Rezazadeh et. al., (2011) proposed a multi swarm particle swarm optimization algorithm for a dynamic environment. Although the algorithm was designed for clustering as such could not be used for grouping of dissimilar participants, the algorithm possessed an adaptive weight which was adjusted based on the improvement of the convergence rate and rank of the swarm. Wang et. al., (2013) added k-means local search with particle swarm optimization (PSO) self-adaptive inertia weight to solve the problem of slow convergence and the fall into local optimum of the algorithm. They proposed the adaptive particle swarm optimization with mutation operator based on k-means (KMPSO). In this algorithm, the particles are dependent on one another, mutation is done between particles. Mutation was done by copying chromosomes from one particle to another However, no attempt was made to explain what will occur if mutation between particles results in replication of chromosomes or the loss of some chromosome in a particle occur during the mutation process.

In another vein, Liang et. al., (2014) created subpopulations and adopted a ring topology to share information within the subpopulations. The inertia weight of a particle was adjusted with respect to the behaviour of the subpopulation the particle belonged to giving particles power to search within their local space. However, it is unclear if this can be used in grouping where the group size is desirable.

The Bayesian search technique was adopted to maintain a balance between exploration and exploitation in the particle swarm algorithm (Zhang et. al., 2014). The inertia weight of the particle was adjusted based on the past position of the particle using Gaussian probability density function. However, the authors pointed out that the technique suffers a local optimum problem which resulted in the introduction of Cauchy mutation of particles to enable long jumps. A weighted variance based adaptive particle swarm optimization algorithm was put forward by Kiani and Pourtakdoust (2015). The weighted variance of the fitness of the particles in the swarm to determine the number of particles to remain in the swarm.

In the survival, sub-swarm adaptive particle swarm optimization (SSS-APSO) and the survival sub-swarm adaptive particle swarm with velocity line bouncing (SSS-APSO-vb) approaches (Pornsing et al, 2015), these algorithms combine time varying and adaptive topology with a fuzzy feedback mechanism to determine the inertia weight to control the parameters. Offsprings were generated from the best particles as the worst particles of the swarm die off from the swarm. The average velocity of the swarm was compared with the velocity of the particle to determine the inertia weight of a particle. This means the collective behaviour of the swarm affects the inertia weight of a particle. However, this does not allow individualized adaptation hence, limiting the particle exploration. In contrast, Chatterjee and Siarry (2006) proposed an adaptive PSO in which a nonlinear function was used to determine the inertia weight of the velocity which is a function of the iteration at each time step while In the Inertia-adaptive Particle Swarm Optimization (IAPSO) introduced by Ghosh, et. al., (2010), the authors introduced the mobility factor to the position update equation of the particle swarm optimization. The distance between each particle and the current global best particle position was used to modify the inertia weight of the particle to keep a balance between exploration and exploitation of the particle. However, there is no evidence that such found global best is the best in the search space which could also prevent other particles from searching into unknown space.

An agent based task oriented algorithm with Particles task (explore or exploit) changing based on the location of the particle relative to the current gbest position found was proposed by Ardizzon et. al., (2015). The algorithm with an adaptive acceleration coefficient in which adaptation was achieved by setting the velocity of the particle nearest to the gbest to zero and with the particle being uniformly moved around its current location to search within the local region. Particles far from the current global best have their inertia weight increased for them to explore the search space. However, it is unclear how the algorithm can be used for other optimization problems since the algorithm was designed for an agent based system. An adaptive parameter control mechanism was introduced which adaptively changes the parameters of the algorithm (Hu, et. al., 2015). The inertia weight, the social factor and the cognitive factor of the algorithm is altered based on the sub gradient of the objective function of the algorithm. Cauchy mutation was introduced to the particles to jump out of local optimum however, conditions for the introduction of the Cauchy mutation are unclear. In addition, the algorithm is complex and could be computational expensive when the number of participants becomes large.

Kiani and Pourtakdoust (2015) used the weighted variance of a particle to self-adapt other control parameters of the particle while Tang et. al., (2015) dynamically modified the inertia weight based on the diversity of the particles coupled with an elitism learning strategy to prevent the particles from been trapped in a local optimum, prompting the inertia weight to adapt to the changes in the environment.

In contrast to the use of parameters for adaptation, Beheshti and Shamsuddin (2015) introduced the non-parametric particle swarm optimization (NP-PSO) algorithm with no parameters other than the particle position being used while the particle best of local neighbours was used by particles to update their velocity. In addition, other additional operator was used in the particle swarm.

2.4.2.8. Additional Operators

Genetic operators were introduced into the particle swarm algorithm (Dong and Cooper; 2013) to increase diversity, these operators are implemented when the diversity criteria among the swarm is less than a predefined value. However, there is no clear standard procedure for determining the diversity criteria for specific problem.

2.4.2.9. Summary of Adaptation

The AI research community have used varying methods of adaptation, uniform weight, linear time varying weight, decreasing weight, increasing weight and feedback mechanism however, researchers who have used feedback mechanism used different variables for their feedback. These include best fitness, particle rank, average local best and global best, distance of particle to global best and local best, success rate, fitness of current and previous iteration. Each of these adaptive mechanisms have been to introduce diversity in the particles as to enable the particle to jump out of local optimum.

Adaptation in the particle swarm algorithm literature can be classified based on the method in which the adaptation is implemented in the algorithm. These adaptations are based on the particle position (Ardizzon, et. al., 2015;

Beheshti & Shamsuddin 2015), fitness of the particle (Aghababa, et. al., 2010), percentage success and error (Nickabadi, et al 2011) and success rate of the swarm (Wang et al., 2015). Similarly, Carlise and Dozie (2000) and Yang et al. (2007) used velocity in the adaptation of the particle swarm algorithm. However, about seventy percent of published literature on adaptive particle swarm algorithm have based their adaptation on the adaptation of the inertia weight in the velocity function of the algorithm with the adaptation of the inertia weight taking different forms.

One major technique of adaptation in the particle swarm algorithm that has+ dominated the research landscape in the past decades has been the use of adaptive weight. However, various researchers have defined (see table 2.5) their formulae by which the weight changes, which is either based on iteration count, fitness of the particle or velocity. Each of these researchers have defined the adaptation of the inertia weight based on the problem domain of the problem set for which their research was conducted showing that there is no single universally accepted technique for altering the inertia weight of the algorithm to balance exploitation and exploration in the particle swarm. This controversy mighty has resulted in the use of other techniques in which adaptation have been introduced into the algorithm however, there have been very few studies conducted in the use of particle fitness, success rate, particle position and the use of velocity of the swarm in the adaptation of the algorithm. Similarly, limited research has been conducted in the hybridization of particle swarm and genetic algorithm.

This section of this review examined the various techniques by which adaptation is entrenched in the particle swarm algorithm with a view to developing a better self-adaptive algorithm with the primary intent of annexing the power of particle swarm and genetic algorithms to form an adaptive hybrid algorithm in which diversity will be introduced into the particle swarm using genetic algorithm. However, only when particles are trapped in a local optimum will the genetic algorithm be invoked. The table below shows the adaptive formula by the various authors.

Table 2.5 PSO Adaptive Formulae Table

Ref	Inertia weight strategy	Adaptation mechanism and Feedback parameter
Suresh et al 2008	W= $w_0 \left[\frac{1-dist_i}{\max\_dist}\right]$	$dist_i = \left[\sum_{d=1}^{D}(gbest_d - X_{id})^2\right]$
Tang et al 2015	W$[E_t]$ = $\frac{1}{0.8+3.2e^{-2E_t ln4}} \in [0.25, 1]\ \forall\ E_t \in [0, 1]$	
Pornsing et al 2015	$w^{t+1} = max[w^t - \Delta w, w_{min}]$; $w^{t+1} = min[w^t + \Delta w, w_{max}]$;	$V_{ideal}^t = V_{ini}\ X \frac{1+\cos(\frac{t\pi}{T_{end}})}{2}$ $\begin{cases} V_{ave}^t \geq V_{ideal}^{t+1} \\ V_{ave}^t < V_{ideal}^{t+1} \end{cases}$
Arumugam and Rao 2007	$w_i = \left(1.1 - \frac{gbest}{pbest_{i_{average}}}\right)$	Acceleration coefficient $C_i = \left(1 + \frac{gbest}{pbest_i}\right)$; distance and global best position
Liang, Li and Zhang, 2015	$w_i^{t+1} = w_i^t - max\{ abs(w_{max} - w_i^t),\ abs(w_{min} - w_i^t)\}$;	Velocity New velocity update model
Hu, Wu and Weir 2013	$W_l^{i+1} = W_l^i - \alpha_l^i g_{wl}^i$	$\alpha_l^i\ g_{wl}^i$ where α_l^i *is step size* *and* g_{wl}^{i} *is gradient of objective function*

Ref	Inertia weight strategy	Adaptation mechanism and Feedback parameter
Ghosh, Das and Kundu, 2010	$W = W_0\left(1 - \frac{dist_i}{\max\_dist}\right)$	
Feng et al 2007	W $= \frac{((W_1 - W_2\ )x(\ max_{iter} - iter\ ))}{max_{iter} + ZW_{2\,.}}$	
Yang et al 2007	$W_i^t = W_{ini} - \propto (1 - h_i^t) + \beta S$	
Zhang et al 2014	$W = (\beta \emptyset \emptyset^T + \alpha)^{-1} \beta \emptyset^T H$	Cauchy mutation
Panigrahi, Pandi and Das 2008	$W_i = W_{min} + \frac{(W_{max} - W_{min}) * rank_i}{total\ population}$	
Yang, et al 2007	$W^i = f(h_i^t \quad S)$ $W_i^t = W_{ini} - \propto (1 - \ h_i^t) + \beta S$	
Jiao, Lian and Gu 2008	$W^i = W * U^{-k}(W \ \in [0,1], U \in [1.0001, 1.005]);$	
Chatterjee and Scarry, 2004	$W_{iter} = f(iter) = \left\{\frac{(iter_{max} - iter)^n}{iter_{max}{}^n}\right\} (W_{ini} - W_{final}) + W_{final}$	
Saber et al 2006	$W_{ijl} = W_{ijl} + \Delta W_{ijl}$	

Ref	Inertia weight strategy	Adaptation mechanism and Feedback parameter
Wang et al, 2015		$V_i^d = \left(\frac{W_{start} - W_{end}}{Max\_Fes}\,(Max_{Fes} - F_t) + W_{end}\right) V_i^d + C.rand_i^d\,(P_{besti}^d - X_i^d)$
Rezazzadeh et al. 2011	$W_i = W_{min} + (W_{max} - W_{min}) * \left(\frac{num\_improved}{swarm\_size}\right)$	
Hu et al. 2015	W = ($W_{max} - W_{min}$)*exp($\frac{g_{max}-g}{g_{max}}$) - W_{min}	
Ardizzon et al 2015	W(t) = ($W_{max} - W_{min}$)$P_s(t) + W_{min}$	Distance from gbest F(t-2) -F(t-1) Particle fitness and gbest Adapts based on closeness to gbest S(i,t) is fuzzy
Kiani and Pourtakdoust, 2015	$W^j = W_{max} - \frac{W_{max} - W_{min}}{iter_{max}} j$	$\sigma^2 w = trace\left\{\frac{\sum_{i=1}^{n} w_i\,(X_i - X_{min})(X_i - X_{min})(\;)}{\sum_{i=1}^{n} w_i}\right.$

2.4.3 Genetic Algorithm

Genetic algorithm (GA) was introduced by John Hollands in the 1960s is a heuristic search method which emulates/mimics the natural phenomenon of natural evolution amongst organisms to solve computational problems by mimicking the natural evolutionary processes of inheritance, mutation and natural selection in the reproduction processes. They consist of a uniformly generated initial population composed of individuals of the population each of which, represent a possible solution. Candidate solutions represent chromosomes which compete and cooperate among themselves for adaptation (Cezary & St Clair, 1995). The initial population evolves resulting in generations. Generations evolve out the process of selection, crossover and mutation often referred to as genetic operators. An objective function is derived based on the objective of the problem to be solved. The objective function is used to evaluate and assign the fitness score of any individual or solution. The fittest among the population are preserved to the next generation. This results in each subsequent generation becoming more suited for the environment. Individuals compete for food and mates for survival and the best produce offspring for the next generation while the less fit extinct. A given set of population at a given time is referred to as a generation, however movement from one generation to the next is characterised by the earlier mentioned operators on the current population. Selection process determines how a chromosome is selected from the population. Selections methods include roulette-wheel selection, tournament selection (Goldberg et al 1989), rank selection, elitist selection, steady-state selection, scaling selection, generational selection, hierarchical selection. Selection enables the system to assign more likelihood to individuals with better fitness and chose higher probability to be chosen for mating. Crossover operation mimics the mating process in natural organisms where the genes of the parents are passed to the offspring by recombining portions of the parent to create offspring. However, to maintain diversity mutation which is flipping of some bits of the chromosomes is done Goldberg, (1989).

Genetic algorithms exist in different forms; the difference in these algorithms exist in their mode of mate selection some of which are Tournament selection, proportionate selection, ranking selection, genitor (steady state) selection. Examples of genetic algorithms include messy genetic algorithm (Kargupta, & Buescher, 1996). In other to implement a GA decision must be taken on (a) chromosome representation (b) fitness evaluation function and selection scheme (c) method of reproduction (d) chromosome replacement method. However, in this review the basic and simplest genetic algorithm the microbial genetic (Harvey, 2009) algorithm is considered due to its simplicity in implementation.

2.4.4 Microbial Genetic Algorithm

A simple and basic genetic algorithm is the microbial genetic algorithm. This algorithm is chosen for this work because of its simplicity and basic nature which makes it easy to implement and easy to combine with other algorithms. Microbial genetic algorithm was proposed by Harvey, (2009) this algorithm mimics the reproduction process of micro-organisms. However, the basic concepts of genetic algorithm are represented by some micro-organism behaviours. In this algorithm, an offspring is generated with one uniform parent then the weaker parent is replaced by the offspring as the weaker parent disappear. Harvey proposed that for different effects, the recombination in genetic algorithms, which in this case is the microbial algorithm represent infection of the loser by the winner, should choose different values of infection (crossover) between 0% to 100%. The infection is then followed by mutation of the offspring generated after the infection. Due to the crossover and mutation between the loser and winner where the winner is maintained and the loser is infested, elitism is obtained for free. This will ensure the fittest member of the particles will remain. Below is the pseudo code of the algorithm by Harvey, (2009).

```
void microbial_tournament(void) {
int A,B,W,L,i;
A=P*rnd(); // Choose A uniformly
B=(A+1+D*rnd())%P; // B from Deme, %P..
if (eval(A)>eval(B)) {W=A; L=B;} // ..for wrap-around
else {W=B; L=A;} // W=Winner L=Loser
for (i=0;i<N;i++) { // walk down N genes
```

```
if (rnd()<REC) // RECombn rate
gene[L][i]=gene[W][i]; // Copy from Winner
if (rnd()<MUT) // MUTation rate
gene[L][i]^=1; // Flip a bit
}
}
```

Below is a diagrammatic representation of the algorithm as presented by (Harvey, 2009)

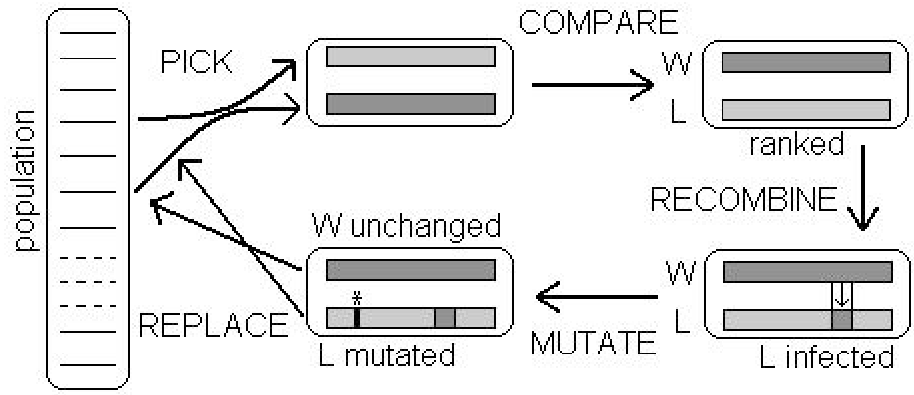


Figure 2.1 Harvey, (2009) Microbial genetic Algorithm

The microbial algorithm due to its basic and simple nature is usually combined with another algorithm like the particle swarm optimization algorithm to form a hybrid algorithm.

2.4.5 PSO + Genetic Algorithm Hybridization Methods

2.4.5.1 Introduction

Hybrid algorithms refer to algorithms which are composed of two or more algorithms, an example is a hybrid PSO and GA means an algorithm created by combining PSO and a GA while hybridization method refers to the way the two component algorithms interact. PSO and GA have some complementary properties because the weaknesses of the PSO seem to be the strengths of the GA while the weaknesses of the GA seem to be the strengths of PSO thus a hybrid algorithm of this two algorithm is believe will be stronger and perform better than both algorithms. Methods for the combination of genetic algorithm with particle swarm to form a hybrid algorithm have been put forward by (Shi et. al., 2005; Premalatha & Natarajan, 2009; Chang et. al., 2013). These methods define the different ways which the two component algorithms interact.

2.4.5.2 Parallel Hybridization Model

The parallel hybridization method was introduced by Shi et al., (2005), PSO-GA hybrid algorithm is based on the variable population size genetic algorithm (VPGA), the VPGA algorithm mimics the situation of war and disease which could cause death in the population. The dying probability for each particle is introduced and particles die when their probability is less than a given threshold value. In the PSO-GA hybrid algorithm, the two algorithms are executed simultaneously with each in a subsystem, when given condition is met, some particles from the two subsystems are selected at uniform and exchanged similarly, when termination conditions are met the algorithm terminates. Below is the algorithm,

- *Start*
- *initialize GA and PSO subsystems respectively*
- *execute PSO and GA simultaneously in their subsystems*
- *store best individual and stop if any individual from subsystem satisfy conditions for termination*
- *perform hybridization (exchange of selected particles by two subsystems) and go to step 3*

2.4.5.3 PSO-GA (Type 1)

In the type one method proposed by Premalatha and Natarajan, (2009) both genetic algorithm and particle swarm are executed simultaneously as is the case in Shi, et. al., (2005) however, the gbest position of the swarm remains unchanged over a given number of steps. In addition, all crossover operations are performed with the gbest. What is unclear in this model is the number of time steps for which the gbest should remain unchanged based on the problem and how will the algorithm respond if a new particle becomes better than gbest during the given steps in which gbest is meant to remain u changed.

2.4.5.4 PSO-GA (Type 2)

In this method the crossover operation of the genetic algorithm is avoided. Rather Particle best position which remains unchanged experience mutation using the mutation operator of the genetic algorithm. What is unclear in the model is how will the mutated particle best affect the algorithm if the mutated particle best has a fitness lower than the particle best from which it was generated. The basic operators of genetic algorithm which include crossover is not implemented hence the advantage of genetic algorithm might not be gotten form this algorithm. This method could be argued to be a modified PSO with mutation operator

2.4.5.5 PSO-GA (Type 3)

In this method, the number of iterations to be executed is divided into two with each of the GA and PSO perform one half of the iterations however, the particles are initialized at uniform and passed into the GA for the first half of the number of iterations. The solutions/particles generated after the number of iterations from the GA are passed, they are as initial particles into the particle swarm algorithm (PSO) to complete the number of iterations. This method could be termed a serial method in which the algorithms are in series whereby the GA completes its iterations and pass its output particle to the PSO. What is unclear about this method is how this could method prevent the PSO from been trapped in a local optimum and prevent the GA from producing similar particles. This method is also referred to by other authors as GA-PSO

2.4.5.6 PSO-GA (Type 4)

In this method for each iteration the particle passes through the PSO and the GA, the initialization of the GA followed by the initialization of the PSO; pbest, gbest, velocity and position are updated; the population is then ranked and passed to the GA where selection crossover and mutation occur. From the generated new population, the required number of individuals are selected to form the next generation of the population these individuals', fitness is evaluated if termination conditions are met the algorithm terminates else goes back to the PSO where update of gbest, pbest velocity and position is implemented.

2.5 Gaps in the Existing Literature

In the published literature, the particle swarm optimization algorithm and genetic algorithm formed most of the published papers in learning groups formation using artificial intelligence techniques. However, most of the experiments using these algorithms suffer from limited sample size and the use of real data. There has also been a division between researchers on the performance of particle swarm optimization algorithm in the learning group formation problem. While Lin et. al., (2010) reported the algorithm performed better than the uniform method Ullmann, et. al., (2015) reported that the uniform method outperformed the Particle swarm when the number of participant increase. In addition, Ullmann, et. al., (2015) reported that the particle swarm algorithm gets stuck after some iterations. Thus, there is the need for an adaptive algorithm which could perform better than the particle swarm. It is also worthy to note that, to date, there has been no experimental evidence to establish an algorithm which perform better than the particle swarm neither is there any experimental evidence of an algorithm better than genetic algorithm in group formation.

The two most common algorithms in the group formation particle swarm and genetic algorithm will be discussed in the next section however, due to the variants of the existing versions of these algorithms the standard algorithm for each will be analysed.

The concept of learning and learning paradigms were briefly introduced although, all the learning paradigms differ in their definition of learning there was a consensus on the need for collaboration to foster better learning and group learning thus how learning groups could be formed was consensus question that needed to be answered.

The existing literature on learning groups formation showed the use of nine different methods with artificial intelligence techniques having four algorithms. Among the four algorithms the particle swarm optimization and genetic algorithm were the most used algorithm. However, there are contradictory reports on the performance of particle swarm which was the most used algorithm. Secondly, sample sizes used with the particle swarm in the various experiments were limited. This indicates the need for an algorithm for grouping increased number of participants into learning groups.

Chapter 3: The Hybrid Algorithm of Microbial Genetic Algorithm and Particle Swarm Optimisation MGAPSO

3.1 Introduction

Analysis of the standard particle swarm and basic genetic algorithm conducted suggests that the particle swarm and the genetic algorithm could be complementary algorithms. Thus, a review of the hybridization methods was conducted as to assess the capabilities of the two algorithms to create a hybrid algorithm. The lack of suitable algorithms for the composition of learning groups with increased number of participants (Ullmann, et.al., 2015) was identified, even though some artificial intelligence algorithms have been used in the existing literature for the grouping problem on a small scale. Notable among these algorithms were the particle swarm algorithm and genetic algorithm. Majority of the researchers have used the particle swarm optimisation (PSO) in the grouping problem but contradictory results were obtained from different studies: some authors reported that the algorithm performed better than the uniform (Lin, et. al., 2010), whereas others proved that the PSO only performed better than the uniform method with limited number of participants and that the uniform method outperformed the PSO when the number of participants increased (Ullmann, et. al., 2015). Moreover, the PSO experienced stagnation after some number of iterations (Ullmann, et. al., 2015), which could be due to the inability of the particle swarm optimization algorithm to maintain a balance between local and global search (Montalvo, et. al., 2007). Another notable drop back of the particle swarm optimisation algorithm has been identified as the tendency of the algorithm getting stuck in a local optimum

(Aghababa, et. al., 2010) due to the speedy convergence of the algorithm, which makes it difficult to be used to properly manage scattering and optimization problems (Bai, 2010). However, the strengths of the PSO algorithm cannot be over looked which includes its simplicity in terms of implementation and the easy with which the algorithm could be combined with other algorithms to form a hybrid algorithm (Bai, 2010).

To combat the grouping problem while considering the drawbacks of the particle swarm algorithm, a hybrid algorithm composed of basic particle swarm optimisation algorithm and a simple microbial genetic algorithm is proposed. The proposed algorithm is tested in a comparative experiment with the particle swarm algorithm. An ANOVA one-way test was then conducted to ascertain whether there exists some significance in the means of the fitness obtained by the two algorithms.

Section 3.2 of the chapter contains a brief overview of collaborative learning while in section 3.3 the new hybrid algorithm of particle swarm optimisation algorithm and microbial genetic algorithm is proposed. Section 3.5 is the conclusion of the chapter which comes after section 3.4 where the experiment of comparing the new algorithm and PSO is described.

3.2 Group Formation for Large Collaborative Learning

Collaborative learning has been described as the process of learners learning together through their interactions while they collectively create knowledge through cooperation among themselves towards achieving their desired common objectives (Annet, 1997; Panitz, 1996; Laal & Laal, 2012). Collaborative learning encourages the exchange of ideas, increases interest among learners and promotes critical thinking learners (Totten, et. al., 1991; Goshala, 1995). In addition, collaborative learning increases learners' participation and active involvement in learning tasks and increases information retention among cooperating groups (Johnson & Johnson, 1986). However, for effective collaboration, learners should learn among small groups (Johnson & Johnson, 1994) which contain learners with similar interests so that they could participate (Vygotsky social learning theory). This means collaborative learning groups should be small groups in which members have the same learning objectives but diverse attributes and a more knowledgeable person should exist in the group as discussed in chapter two. Thus, the grouping requirement for collaborative learning groups are that the groups formed should be small (e.g., size of 5-6) (Johnson & Johnson. 1994) and in other to improve learning outcomes diversity within each group is required (Vygotsky). In chapter Two of this thesis, algorithms used for the composition of collaborative learning groups have been discussed, however, the contrary reports of research showed that there is lack of an algorithm for collaborative learning group composition with a large number of participants hence the proposition of a new algorithm in this chapter.

3.3 The Hybrid MGAPSO Algorithm

The proposed MGAPSO algorithm is an algorithm which is composed by the hybridization of the basic particle swarm algorithm (Kennedy & Eberhart, 1995) and the microbial genetic algorithm (Harvey, 2011) and both algorithms have been explained in the previous chapter. The intention of hybridization is to annex the capabilities and simplicity of both algorithms in a single algorithm. However, there are different hybridization methods.

Hybridisation methods describe the ways in which the particle swarm optimization algorithm is combined with the genetic algorithm (Premalatha & Natarajan, 2009). They discuss the interactions of the particle swarm algorithm with the chosen genetic algorithm. Hybridization methods are discussed in chapter Two of this thesis in detail. The hybridization method proposed by Chang, et. al., (2013) can be termed as a serial model, in which every particle experiences both the genetic algorithm and the particle swarm algorithm during each iteration. This means every particle in (Chang et al; 2013)'s model undergoes all the genetic operations of crossover and mutation and also experiences all the unique operations in the particle swarm algorithm. (see analysis of hybridization method in chapter Two of this thesis).

The proposed hybrid algorithm of MGAPSO follows a scheme similar to (Chang et al 2013) due to its completeness. However, in order to make the algorithm simple for large scale applications, the ranking of particles in (Chang et al., 2013) is not implemented in this algorithm. In the proposed MGAPSO algorithm, particles are initialized uniformly, particles select their mates at uniform, crossover and mutation of the genetic algorithm operations are performed, elitism is enforced by the selection of the fittest child to replace the parent, a new offspring from genetic algorithm replaces the parent and uses the velocity of the parent to achieve a new position through particle swarm update equations. Thus, in each iteration, a particle passes through the genetic algorithm and particle swarm algorithm but a simpler version of genetic algorithm, "microbial genetic algorithm" (MGA), is adopted to hybridise with particle swarm algorithm. The use of MGA makes the new hybrid algorithm easy to implement and customise.

The Microbial genetic algorithm (MGA) was introduced by Inman Harvey in his paper (Harvey, 2009). The algorithm is a simplified genetic algorithm which mimics the reproduction process in micro-organisms such as bacteria. Reproduction in Microbes is by asexual reproduction and hence referred to as binary fission in which the organism undergoes cell division to produce two identical DNA organisms (Angert, 2005). Although mimicking the reproduction in microbes, Harvey's microbial genetic algorithm is described in terms of bacterial infection. Bacterial infection is such that one bacterium infects the other and the processes involved are equated to basic processes in the standard genetic algorithm. In MGA two parent bacteria are chosen within the same geographical locality uniformly and their fitness values are evaluated. The bacterium with higher fitness is assigned as a winner and the one with the lower fitness is assigned as a loser. Recombination then occurs, termed as infection in microbial GA, while the loser is the genetic material infested by the winner. The infection rate is considered as equivalent to the recombination probability which is the percentage of the winner that will be copied to the loser as the winner remains unchanged and the loser is modified. After the infection of the loser by the winner, mutation is performed and a new offspring is produced which replaces the loser in the population. Harvey pointed out that one disadvantage of the selection process of the MGA algorithm is the tendency to form a homogeneous population due to the selection of mate within the locality of the member. To combat this drawback, mate selection in the proposed hybrid algorithm in this chapter is at uniform from the entire population instead of from the locality of the member. Thus, all mates will be selected from the swamp at uniform. Then the process of reproduction (e.g., crossover and mutation) is performed to produce offspring to replace weaker parents, which adds variety to the population. With this, the tendency of all particles in the swarm to converge prematurely can be mitigated.

The microbial genetic algorithm is thus used to add diversity to the particles in the swarm for the purposed preventing particles from getting trapped in a local optimum as experienced by the particle swarm algorithm. Similarly, the particle velocity of the algorithm will prevent the particles from becoming homogeneous as experienced by the chromosomes in microbial genetic algorithm. Thus, each particle at each iteration passes through the genetic algorithm and the particle swarm bit of the MGAPSO algorithm serially to accelerate convergence but avoid premature convergence.

Hence, in this new MGAPSO algorithm, particles will select a mate at uniform from the population. In the microbial genetic algorithm, the first parent is selected at uniform and the second is selected within a distance of five other parents closest to the first parent, this is not the case in this new algorithm. It can be argued that with the selection process in the microbial genetic algorithm, there is the tendency of some chromosomes not to be selected for some iterations. Harvey pointed out that the microbial algorithm due to its mate selection method could lead to a homogeneous population where a few dominate the population. To overcome this drawback, in this new algorithm all members select a mate at uniform from the entire population and the offspring is used to replace a single parent. In addition, to overcome the tendency of some chromosome not being selected for some iterations, in this algorithm every chromosome must be a parent and thus every parent chromosome is engaged in the reproduction process during every iteration. The concept of selecting a fitter male for reproduction is not the case in this algorithm; rather any member of the population could be selected from the entire population. This can be argued as closer to the natural selection among other living organisms in real life. An example is in humans where people do not most times choose a stronger or better mate, rather they chose mates like them which is referred to as non-uniform mating (Beauchamp & Yamazaki; 1997). However, non-uniform mating does not mean equality in fitness nor does it mean choosing a mate with higher fitness. What it does mean is that similar characteristic but at uniform within the population of members with similar characteristics. Thus, all mates will be selected from the swamp because all members of the population in this case are similar.

The process of reproduction involving crossover and mutation operations is performed to obtain offspring. This is to add variety to the population. The reason is to prevent the population from becoming homogeneous. The microbial genetic algorithm will thus add diversity to the particles in the swarm preventing the particles from getting trapped in a local optimum as experienced in the particle swarm algorithm.

In addition, the process of the winner infecting the loser as obtained in the microbial genetic algorithm will not be used, rather every member at each iteration chooses a mate at uniform with which crossover is performed to produce an offspring which replaces the member and the member dies. Similarly, the particle velocity of the algorithm will enable the offspring to explore the search space and prevent the particles from becoming homogeneous as experienced. This means, each particle at each iteration passes through the genetic algorithm and the particle swarm algorithm serially.

The hybrid MGAPSO algorithm is stated as follows

- *Initialization: Generate initial population*
- *Evaluation: calculate the fitness of each particle in the population, update global best position*
- *Evolution: generate offspring by the process of selection, crossover and mutation*
- *Evaluation: calculate the fitness of particle, update particle best and global best position*
- *Calculate velocity of particle, update particle position, particle_best position and gbest;*
- *Termination condition: if condition is not met then go back to third step*

The figure 3.1 below shows the flow chart of the MGAPSO algorithm.

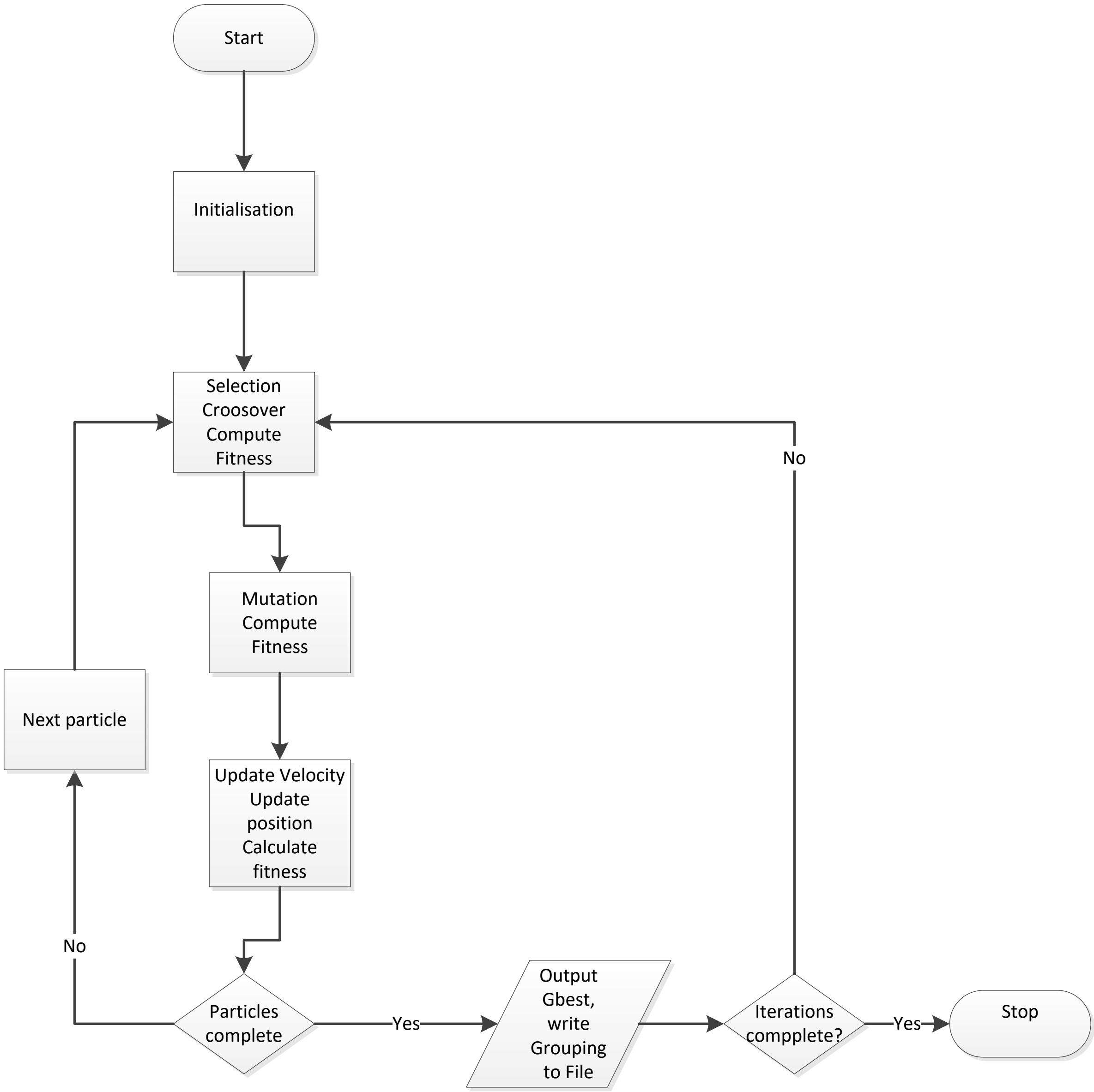


Figure 3.1 Flowchart of MGAPSO 1

The pseudocode of the MGAPSO algorithm is as follows.

ALGORITHM (Hybrid MGAPSO Algorithm)

Start

Initialisation of particles, particle velocity, pbest and gbest

While stopping condition (e.g., number of iterations or threshold) is not met

For each particle

Select particle_mate,

Perform crossover and mutation,

Compute fitness of offspring, update particle fitness, gbest, pbest

Compute velocity, update particle position,

Compute particle fitness

Update gbest, pbest, particle fitness

EndFor

Write out gbest particle to file

End

The hybrid MGAPSO algorithm design involves a series of essential operations, including particle representation, velocity representation grouping representation, fitness function derivations, mate selection in MGA, crossover and mutation, which are explained below.

3.4 Particle Representation

Particles in this algorithm are represented with a two-dimension array in which the rows represent learners involved and the columns represent the groups that can be formed. In this array each row thus represents a single learner and the learner can belong to any one of the groups, that is, a column of the array. The use of an array permits the movement of a learner from one group to another easily. At the initialisation stage of the particle, the number of the population is divided by the initial group size required which gives the maximum number of groups that will be found. The maximum number of groups to be found equals the number of columns in the particle. For example, if there are 20 learners and the initial group size is 4, a particle will be represented by a two-dimensional array with 20 rows and 20 divided by 4 columns, that is, the particle in this example is a 20 by 5 matrix.

3.4.2 Velocity Representation

Velocity is the rate of change in magnitude and direction of a particle in the solution space. In the particle swarm algorithm, it is used to move the particle from one location to another. The velocity has to be represented such that arithmetical operations can be performed between the particle and the velocity. To achieve this the velocity representation needs to be the same format as that of the particle. This will enable the corresponding item to item arithmetical operations when adding velocity to particle operation in the particle update equation. This means the velocity is also represented by a two-dimensional array which has the same dimensions of the particle. An example is: if the particle has 200 rows with 40 columns then the velocity will also have 200 rows and 40 columns.

3.4.3 Particle Initialization

The initialization of the particle is done by uniform assignment. A uniform number generator is used to do the uniform assignment. The uniform generator selects at uniform some numbers equal in number to the group size and those identity numbers of participants are used to form the first group. This selected numbers are not replaced, then a second set are selected from the number left and used to form the second group in the particle and assigned 1.0 till all numbers have been assigned to groups. An illustration is: if there are 20 participants then we have 1 to 20 as the identity numbers of the participants, assuming we want 4 people in a single group this means the particle will be a two-dimensional array with 20 rows and 5 columns. The uniform assignment will be done as follows. Four numbers between 1 and 20 inclusive will be chosen at uniform for the first group. Assuming the 4 numbers between 1 and 20 inclusive selected are 6, 9, 14, 17, this means for this row the first columns in the array will be assigned the number 1.0 meaning these participants with this identity numbers belong to the first group. Having selected those numbers from the universal set which is the set of numbers from 1 to 20 inclusive, this numbers are not replaced thus we are left with 16 numbers (1,2,3,4,5, 7,8,10,11,12,13,15,16,18,19,20) which are 20-4=16 numbers. The next selection is then done on the remaining 16 and selection is still done are uniform. If in the next step 7, 13, 11 and 19 are select, then the next selection 1, 3, 5 and 10 are selected and the fourth selection 2, 8 15 and 20 are selected then the remaining number will belong to the last set. At this juncture the first set will have 1.0 in their first columns the second set will have 1.0 in their second column while the third set will have 1.0 in their third column and so on after the completion of the assignment of 1.0 all other columns in the array will then be filled with the value 0.0. Although, instead of filling with 0.0 another alternative is to fill with a uniform number less between 1 and 0 but filling with 0 means the movement of the particle is solely dependent on the velocity and the initial particle position. The example explained above is thus represented as shown in figure 3.2 below.

3.4.4 Velocity Initialization

The velocity of a particle is represented as a two-dimensional array equal in dimension with the particle array thus in the example give above where the particle has 20 rows and 5 columns the velocity will also have 20 rows and 5 columns. However, at the initialization stage, uniform numbers between 0 and 1 are generated using a uniform number generator and assigned as values for each array element of the velocity array.

ID	Group1	Group2	Group3	Group4	Group5
1	0.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	1.0	0.0
3	0.0	0.0	1.0	0.0	0.0
4	0.0	0.0	0.0	0.0	1.0
5	0.0	0.0	1.0	0.0	0.0
6	1.0	0.0	0.0	0.0	0.0
7	0.0	1.0	0.0	0.0	0.0
8	0.0	0.0	0.0	1.0	0.0
9	1.0	0.0	0.0	0.0	0.0
10	0.0	0.0	1.0	0.0	0.0
11	0.0	1.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	1.0
13	0.0	1.0	0.0	0.0	0.0
14	1.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	1.0	0.0
16	0.0	0.0	0.0	0.0	1.0
17	1.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	1.0
19	0.0	1.0	0.0	0.0	0.0
20	0.0	0.0	0.0	1.0	0.0

Figure 3.2 Particle Representation at Initialization

3.4.5 Grouping Representation

The grouping is the representation of the actual groups in which participants are grouped into, by the grouping algorithm. The grouping is a two-dimension array equal in dimension to the particle and the velocity, it is a binary array which contains only 0 and 1. The constraint of a participant/learner belonging to one and only one group at a given instance is implemented in the grouping using binary numbers. This means that a row can have only a single 1 in the grouping array while all other elements of the row are 0. An example of the grouping is shown in Figure 3.3.

Figure 3.3 is an example of a particle representation with 20 learners grouped as follows

Group 1 has learners with ID 3, 5, 9 and 13

Group 2 has learners with ID 6, 7, 15 and 17

Group 3 has learners with ID 8, 11, 16 and 18

Group 4 has learners with ID 2, 12, 19 and 20

Group 5 has learners with ID 1, 4, 10 and 14

Although the grouping above shows 5 groups with equal number of learners, in each group the equality of the number of learners per group is not a requirement meaning the groups may vary in size.

After velocity and particle update, the grouping which is the groups formed is obtained from the particle by selecting the maximum value in each row of the particle array and converting it to 1 while the rest are converted to 0. This is because the chances of the learner belonging to that group is the highest. An example is assuming the following in values Figure 3.4 are obtained in a row of the particle matrix

ID	Group1	Group2	Group3	Group4	Group5
1	0	0	0	0	1
2	0	0	0	1	0
3	1	0	0	0	0
4	0	0	0	0	1
5	1	0	0	0	0
6	0	1	0	0	0
7	0	1	0	0	0
8	0	0	1	0	0
9	1	0	0	0	0
10	0	0	0	0	1
11	0	0	1	0	0
12	0	0	0	1	0
13	1	0	0	0	0
14	0	0	0	0	1
15	0	1	0	0	0
16	0	0	1	0	0
17	0	1	0	0	0
18	0	0	1	0	0
19	0	0	0	1	0
20	0	0	0	1	0

Figure 3.3 Grouping Representation

0.456	1.345	1.25	0.987	0.457

Figure 3.4 showing a single row of a particle array

During the conversion of Figure 3.4 shown above state, the corresponding row in the grouping is shown in Figure 3.5.

0	1	0	0	0

Figure 3.5 Row in Grouping corresponding to row of Figure 3.3 in particle array

The second column is assigned 1 because it has the highest value and the rest assigned 0; This implies that the participant/learner will belong to the second group and cannot belong to other groups at the time.

3.5 Fitness Function Derivations

There seems no consensus fitness function across the field of study with researchers defining their respective fitness function. Thus, each defining their fitness based on their intended objective. The fitness function is used to evaluate the fitness or how good a particle is with respect to the objective of the fitness design. It is a mathematical representation of the objective of the user. In this research the requirements are different from the requirements in Lin, et. al., (2010). In Lin, et. al., (2010) grouping is done so as to minimise the difference in understanding levels between groups such that the average understanding levels in the groups are equal or close to being equal and the second criteria in Lin, et. al., (2010) is that there is at least one member of the group who is interested in a topic in all the topics. However, this is not the case in this research. In this research consideration is given to diversity of learners attributes such as understanding levels and interest levels within the groups, although grouping is done to ensure equality in understanding levels between groups consideration is also given to diversity within the groups which is not considered in (Lin et al; 2010). Secondly, Lin, et. al., (2010) do not consider equality in interest levels between groups which is an additional consideration in this study. Lin et. al., (2010) thus result in groups in which the interest levels are equal or very close and in every group there is at least one member in the group who has interest in a topic, while in this research the outcome is groups which are equal in understanding levels and also equal in interest levels with there being diversity within the groups in their interest levels and understanding levels. This consideration makes this research more complex and realistic. The fitness function of this research is designed based on the conditions for collaboration supported by the social learning theory (Albert Bandura) and social development theory (Vygotsky).

An additional argument against Lin et al., 2010) is that a learner being interested in a topic may not necessarily mean the learner's understanding level is high enough to help his/her peers in the learning process. Also, if a single learner with a low level of interest is grouped into a group in which that learner is the only learner in the group who has interest in that topic, this may create an imbalance among the groups during collaboration and such learner may not be able to perform as a MKO in the group. These setbacks in the fitness design of Lin, et. al., (2010) necessitates the need for a more encompassing fitness design for the collaborative learning groups formation. Thus, in this research, grouping is done to achieve a balance between groups in the understanding levels and interest levels in all topics which will result to creating a more balance grouping for better and more effective collaboration.

The requirement is to group learners as follows:

(a) That each group should contain as much diversity as possible in terms of the attributes used in the groups;

(b) That a learner should belong to one and only one group at a given time;

(c) That the compositions of groups formed should be similar, meaning that the distribution of the various attributes is similar in all groups.

Harrison and Klein (2007) defined diversity with respect to the distribution of a common attribute among the population, how are the values of the attribute distributed in the population. Consequently, grouping learners into groups which are all diverse in their attributes will thus be achieved with respect to the distribution of the attributes in the population.

The authors also pointed out separation, variety and disparity as the different types of diversity with the Blau's index as a measure for variety, standard deviation for separation and coefficient of variation as an index for disparity. However, separation applies in a case where group members hold relative positions in the group and are from the same background, variety applies when group members are from different backgrounds while disparity applies when a member is superior to other members of the group (Harrison & Klein, 2007). In this research all group members are of the same background and members hold relatively important positions in the group with varying knowledge levels making them occupy varying positions in the group, meaning, diversity in this research is a separation type which makes standard deviation the choice of index for the measure of within group diversity.

To evaluate how diverse within the groups, while the groups are made similar, the difference between the measure of diversity within the groups and similarity among groups is maximised. This means if a measure of the diversity among the groups is subtracted from the measure of the diversity within groups, an increase in this fitness implies that the groups are becoming similar and the diversity within each group is getting larger. When the variation among the groups become smaller, this implies that the groups are becoming similar and as this similarity becomes higher and the diversity between groups become smaller, the fitness becomes larger. The means of the attributes in the groups are used to calculate the variation among the means of the groups, while the standard deviation within the groups is used to calculate the variation within the groups.

The standard deviation of a given attribute of the population is given by the following formula:

$$d_{(g,a)} = \frac{\sqrt{\frac{\sum_{k=1}^{|G|}(X_{(k,a)} - \bar{X}_{(g,a)})^2}{|G|_g}}}{R_a} \qquad eq\ 3.1$$

Equation 2.1 above gives the standard deviation of a given attribute in a group g. This value is normalised over the range Ra of the attribute a. Similarly, the diversity among the groups is given as:

$$D_a = \frac{\sqrt{\frac{\sum_{g=1}^{|G|}(\bar{X}_{(g,a)} - \bar{\bar{X}}_a)^2}{|G|}}}{R_a} \qquad eq\ 3.2$$

Equation 2.2 is the diversity among the groups formed. It is the standard deviation of the mean of the means of each attribute in the groups. To obtain this, the means of the groups are calculated first and then the mean of the means of the groups are used to find the diversity among the groups. This means when the groups become similar where the means of all the attributes of all the group become very close to one another, the standard deviation among the means of the attribute becomes very small and in turn this value increases the fitness. When the groups become less similar, this value will increase hence reduce the fitness value obtained. The range Ra of the attribute is used to divide the value as to standardize the value. In a case where the value of the attribute is between 1 and 5, the range is 4. Thus

$$\text{fitness} = \frac{\sum_{g=1}^{|G|}\sum_{a=1}^{|A|} d_{(g,a)}}{|G||A|} - \frac{\sum_{a=1}^{|A|} D_a}{|A|} \qquad eq\ 3.3$$

Thus

$$\text{fitness} = \frac{\sum_{g=1}^{|G|}\sum_{a=1}^{|A|} \frac{\sqrt{\frac{\sum_{k=1}^{|G|}(X_{(g,a)}-\bar{X}_{(g,a)})^2}{|G|g}}}{R_a}}{|G||A|} - \frac{\sum_{a=1}^{|A|} \frac{\sqrt{\frac{\sum_{g=1}^{|G|}\left(\bar{X}_{(g,a)}-\bar{\bar{X}}_a\right)^2}{|G|}}}{R_a}}{|A|} \qquad eq\ 3.4$$

where

|G|	Number of groups formed
|A|	Number of attributes
R_a	Range of attribute a
$\bar{X}_{(g,a)}$	The mean of attribute a for group g
$\bar{\bar{X}}_a$	The mean of the means of attribute a for all groups
D_a	Similarity between the groups for attribute a
$d_{(g,a)}$	Deviation of attribute a in group g

Equation 2.3 gives the fitness function while equation 2.4 gives the expanded expression of the fitness function. When the groups become similar meaning the means of the groups in all attribute get closer, the similarity value reduces towards zero and when the diversity within the groups increases the expression in equation 2.4 becomes large hence the algorithm is maximising the left end of the equation while minimising the right end of the equation. This fitness value is thus used to evaluate how good a grouping is.

3.6 Mate Selection in MGA

In biology mate selection is the process of choosing another organism of its type to reproduce an offspring with similarity. In genetic algorithms mate selection is the selection of chromosomes to couple with to produce offsprings, however, in genetic algorithms (Holland, 1995) selection of individuals to breed to produce the next generation is biased to selecting individuals with higher fitness. One would argue that with the bias selection some good traits in some chromosomes might be lost and also that although an individual might have a lesser fitness it can possibly produce an offspring with a high fitness. Naturally, organism most times do not always choose their mates based on strength. Instead, mate selection is relatively at uniform where an individual may choose a mate not minding how strong the mate is because the individual might not be able to determine the strength of the other individual she is selecting as a mate. Even when organisms choose their mates using any criteria they do not always search for the best or the strongest they go for the any member of their specie who seem better than them in any way. In Harvey's microbial genetic algorithm, the selection is at uniform but within the locality of the one parent. The first parent is selected at uniform and the second parent is selected within the locality of the first. An argument against this selection process is that there is the possibility of some individual not taking part in the production of the next generation since both parents are selected at uniform, however Harvey also pointed out that the disadvantage of this process is that this selection could result in a homogeneous population. Therefore, in this research there are some slight changes to the mate selection process of Harvey (1999) microbial genetic algorithm.

In this research every particle will be involved in the reproduction of the next generation, for every particle in every iteration a mate is selected at uniform from the population and the offspring will replace the first parent in the next generation. This tends to be naturally because as in the case of humans where it does not matter if the individual's child is better than the individual or not when a person dies her child is the next generation. The selected mate will then perform crossover and recombination with the first parent.

3.6.2 Crossover

Crossover techniques describe how the set of genes of the parents will be divided for recombination to produce the offsprings. This process enables some of the characteristics of the parents to be passed onto the offsprings. This concept inspired the crossover operations in genetic algorithm whereby properties of a solution can be passed from the parent generation to the next generation. Single point, two-point and uniform crossover are among crossovers techniques used in genetic algorithm (Vekaria & Clack, 1998). In the single point the genes of the parents split at a single point and recombination occurs while in the two-point technique the splitting occurs at two points for recombination.

There are various methods of the recombination of the divided genes, referred to as the crossover operator. Examples are partially matched crossover (PMX) (Goldberg & Lingle, 1985), cycle crossover (CX), order crossover operator (OX), position-based crossover operator (POS), voting recombination crossover operator (VR), sequential constructive crossover operator (SCX) (Zakir, 2010), edge recombination operator, alternate-position crossover operator (APX), brood recombination (Tackett, 1994) etc. This is not a detailed research into crossover techniques neither is it a detail research into genetic algorithms. Interested readers should see (Potts,et. al.,1994; Geethamani, 2016).

In partially matched crossover (PMX) (Goldberg & Lingle, 1985), crossover points are selected at uniform and parents are mapped to one another then there is a position by position exchange in the parent to generate the offspring.

Parents						Offsprings of single point crossover					
A1	A2	A3	A4	A5	A6	A1	A2	B3	B4	B5	B6
B1	B2	B3	B4	B5	B6	B1	B2	A3	A4	A5	A6

Parents						Offsprings of two point crossover					
A1	A2	A3	A4	A5	A6	A1	A2	B3	B4	A5	A6
B1	B2	B3	B4	B5	B6	B1	B2	A3	A4	B5	B6

Figure 3.6 Examples of Crossover Operation

In the single point of figure 3.6 above the parents split at a single point between 2 and 3 and B1 and B2 recombine with the rest of A while A1 and A2 recombine with the rest of B. However, in the two-point crossover demonstrated above the division is done at two points between 2 and 3 then, between 4 and 5. Thus A3 and A4 replace B3 and B4 in the offspring of B while B3 and B4 replace A3 and A4 in the offspring of A. The right hand side of figure 3.6 shows the offsprings while the left hand side of figure 3.6 shows the parents from which the offsprings were obtain.

In this research crossover rate is by a uniform, this means the number of genes to be copied from one parents is determined at uniform. Thus, a uniform number generated at a given time which varies from iteration to iteration and particle to particle determines the amount of genes to be used for crossover. The use of a uniform rate allows for variation of the crossover rate. This is in accordance with the Harvey algorithm where the crossover rate will vary between 0 and 100 percent, thus the crossover rate will be based on a uniformly generated value. The value of the uniformly generated crossover value will determine how much of the winner chromosome will be copied into the loser. Similarly, the starting point of copying will also be determined by a uniform number. This allows for variation among particles preventing the population from becoming homogeneous

3.6.3 Mutation

Mutation in biology is the alteration of the sequence of the DNA strands that make up the gene of the organism making it different from the one parent from which it was formed. However, mutation may not always result in very visible changes in the organism., Mutation can occur in any part of the gene and vary in size (Bernstein, et. al., 1985). This phenomenon is mimicked in genetic algorithms to generate offsprings that are slightly different from the parent individuals (Srinivas & Patnaik, 1994). Readers interested in types of mutation should see (Sheng & Gu, 2014)

In genetic algorithm, mutation achieved is by changing the order of the pattern in the solutions. In uniform mutation the point of mutation is obtained by generating a uniform position in the solution to mimic the uniform mutation position as obtained in the biology (Sheng & Gu, 2014). The size of the mutation which also varies in biology is also mimicked by the generation of a uniform value between 1 and 0, which is referred to as the mutation probability. In this research, the mutation probability is multiplied by the number of rows in the particle which gives the number of items to be swapped in the particle. Half of this number is obtained which give the number (k) of swaps that will be made on the particle. A loop is set from 1 to k and two uniform numbers are generated within the size of the particle which are not the same then these two rows of the particle are swapped. An example is: if we have a particle with 100 learners and we then had a mutation probability of 0.2, then we multiply 0.2 by 100 which gives 20. This means 20 rows will be involved in the exchange process, then 20 is divided by 2 which gives 10, which means 10 swaps will be implemented.

This section has introduced the MGAPSO algorithm and the essential operators involved. The next section will demonstrate the performance of MGAPSO on grouping through a series of experiments.

3.7 Experimental Design

Two types of computational experiments have been identified by (Barr, et. al., 1995) when using algorithms, which are experiments for comparing the performance of different algorithms for the same class of problem using same datasets and experiments in which algorithm performance are characterised in isolation. However, Mcgeoch (1996) classified experiments based on the aim of the researcher in the use of the algorithm into Dependency study or Experimental Average-case study. The aim of this type of experiment is to uncover the functional relationship between the factors of the algorithm and its performance measures. This experiment thus examines the average behaviour of an algorithm for which probabilistic analysis is difficult. For example, one could investigate the behaviour of the particle swarm algorithm when the parameters of inertia weight, social and constriction factors affect the performance of the algorithm.

Horse Race study or Competitive testing (Mcgeoch,1996): this is a study in which an attempt is made to determine the superior algorithm among a set of algorithms by running the set of algorithms in a given context. An example is running the particle swarm optimization algorithm and the microbial genetic algorithm in the grouping of student to determine which of the two algorithms will perform better in the context of grouping learners.

Probing Study (Mcgeoch,1996) is designed to examine the workings of an algorithm as to determine the strength and weaknesses of the algorithm while measuring the internal operations of the algorithm, for example, investigating the effect of initialization in the particle swarm optimization algorithm.

Robustness Study: it is a study which examines the distribution of a given property in the study, for example, the standard deviation of a measure of some sort, what is the mean measure of the selected property and what is the standard deviation of the measure and to find the standard deviation of the means of the fitness after several runs of the particle swarm optimization algorithm.

Comparative experiment: (Johnson, 2002) this type of experiment in computing is either attempting to determine the best solution for a specific problem or developing a new algorithm of solving a given problem. It involves comparing the newly developed algorithm with an existing algorithm using same data set and same parameter settings. This type of experiment allows the researcher to design a null hypothesis that there is no difference between two or more algorithms. Comparative analysis enables the test of a null hypothesis to determine if there is any significant difference in the performance of the algorithms so compared. Victor Basili described comparative experiment as the heart of quality improvement in computer science.

Application Study (Johnson, 2002): an application study deals with how a given set of algorithm or codes will work with a given application domain. An example is how well the particle swarm algorithm performs in the grouping of students. Application study is the specific use of the other studies in a specific domain. However, the choice of experiment depends on the problem domain and the life cycle of existing algorithms in the problem domain. This study is a comparative experiment as it is determined to find a best solution to a given problem. It also involves developing an algorithm to solve a given problem.

3.7.2 Purpose of Experiment

The purpose of this experiment is

- To ascertain if the new MGAPSO could be used in forming groups;
- To compare the new MGAPSO with the particle swarm optimisation algorithm in the grouping problem.

The proposed algorithm will be implemented using java. Both the proposed algorithm and the normal particle swarm optimisation algorithms (Kennedy & Eberhart, 1995) will be implemented and tested on some simulated datasets. Each experiment has been run 50 times on each set of data and the averaged global best fitness of the two algorithms over 50 runs are plotted on a single sheet to evaluate the proposed hybrid MGAPSO algorithm. A stability analysis is also conducted by finding the standard deviation of the fitness of the 50 runs of both algorithms for each data set and sample size.

3.7.2.1 Data Source

Data for this experiment was simulated and the attributes of learners will follow two distributions (Gaussian and uniform distribution) respectively.

Experimental Method

The inertia weight of the particle swarm, the social and cognitive coefficients of the particle swarm are variables used in the particle swarm optimization algorithm.

Parameters used in the particle swarm were w= 0.8, C1 = C2= 1.49445

3.7.3 Experimental Results

The proposed MGAPSO algorithm and the classic PSO are tested together for grouping 200, 500 and 1000 students whose attributes follow the uniform distribution. The best fitness obtained in each case are shown in the following tables and figures.

Table 3.1 Gbest fitness obtained for uniform distribution

Gbest fitness for Uniform Distribution		
sample size	PSO	MGAPSO
200	0.3829	0.6587
500	0.3514	0.6461
1000	0.3277	0.625

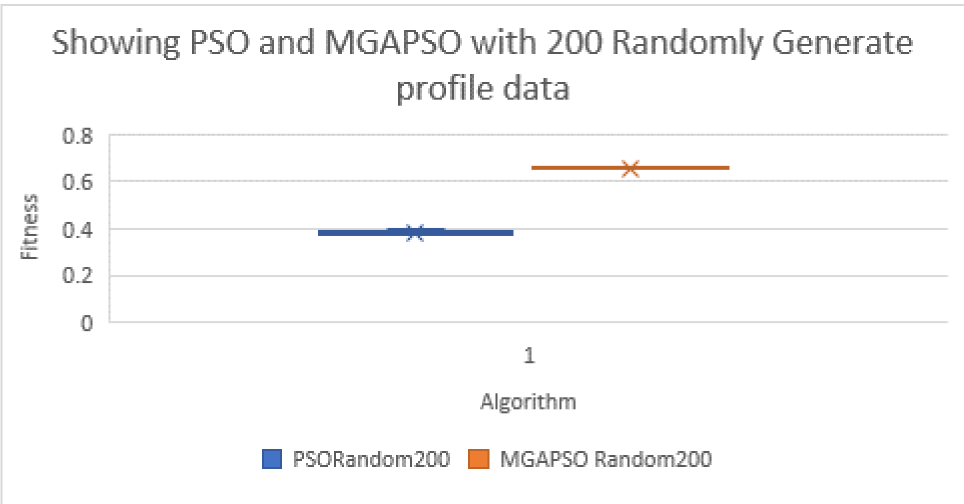


Figure 3.7a Gbest Fitness obtained for 200 students and Uniformly Generated Profile data

Figure 3.7a shows the range of the global fitness (Gbest) of the particle swarm algorithm and the MGAPSO. The figure shows a thin line for the MGAPSO which is an indication of high convergence meaning the range of the fitness obtained is very small. Similarly, the PSO has a thin line indicating convergence, however, its range is higher than the that of the MGAPSO. Moreover, the fitness of the MGAPSO is much higher than the fitness of the PSO, suggesting a much better performance on grouping. In this comparative experiment with this data type and sample size it could be stated that the MGAPSO has outperformed the PSO and converged better than the PSO.

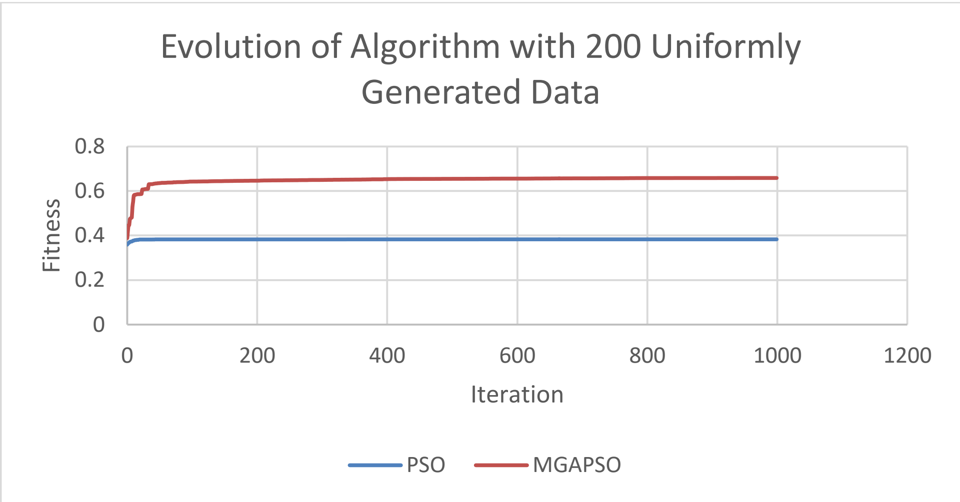


Figure 3.7b Evolution of PSO and MGAPSO with 200 Uniformly Generated data

Figure 3.7b shows the evolution of the global best particles for the particle swarm algorithm (PSO) and the MGAPSO algorithm. The graph shows that the global best fitness obtained by the MGAPSO is higher than the global best of the particle swarm, meaning the MGAPSO outperformed the PSO. However, the graph also shows that after a few iterations the particle swarm get stuck and could not find a better fitness after some iterations resulting in the curve of the PSO remaining nearly parallel to the horizontal axis of the graph and the global best fitness remained nearly constant all through the experiment.

The curve of the MGAPSO shows an increasing graph resulting in higher fitness values which means that the MGAPSO has a stronger exploratory. Although the slope of the curve tends to be very small, there are increments as the iterations increase. This is an indication that the global best particle of the MGAPSO evolves and the algorithm has a higher ability to jump out of local optimum than the particle swarm algorithm.

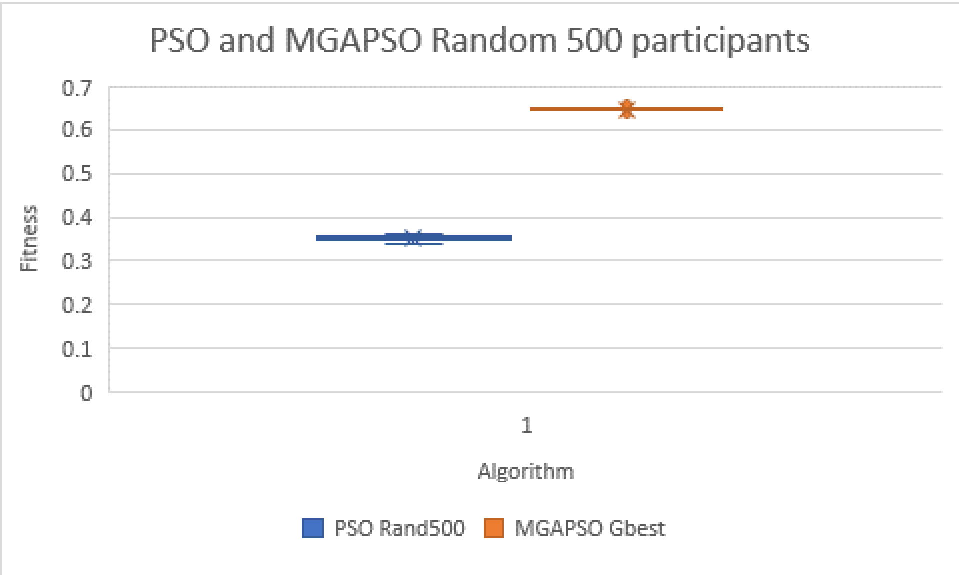


Figure 3.8 Gbest Fitness obtained for 500 students and Uniformly Generated Profile data

Figure 3.8 also shows a higher fitness by the MGAPSO than the PSO for 500 students with uniform attributes. The thinness of the MGAPSO meaning a higher convergence of the global best particles of the MGAPSO.

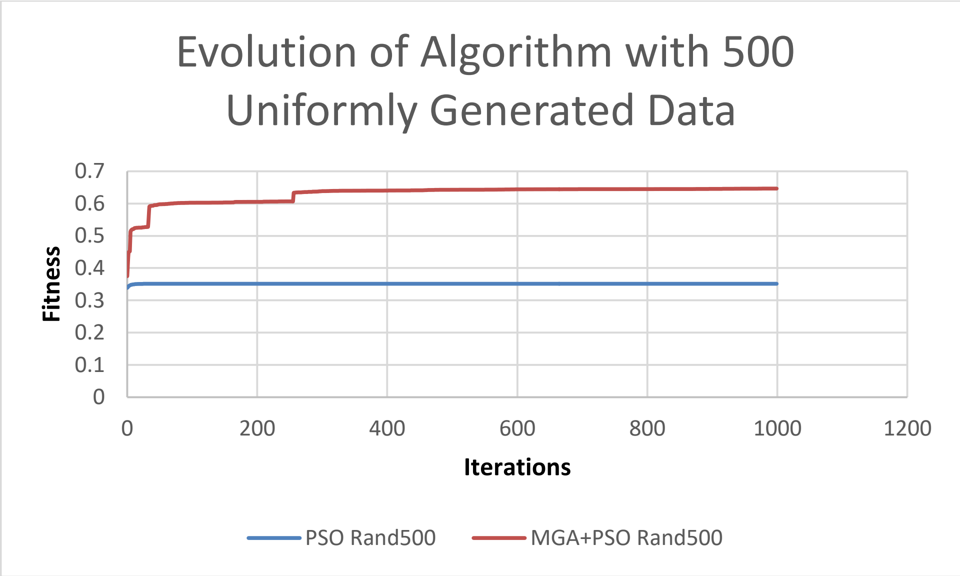


Figure 3.9 Evolution of Global best particle of PSO and MGAPSO with 500 students and Uniformly Generated profiles

Figure 3.9 shows the evolution of the global best particle for MGAPSO and the PSO with 500 uniformly generated profile data. The curve for the PSO starts moving and obtains a slight higher fitness from its initial global best, after which the curve becomes parallel to the horizontal axis. This is an indication that after the algorithm establishes a local optimum, the algorithm could not jump out of the optimum found and thus remains trapped in the local optimum.

The graph for the MGAPSO shows an interesting behaviour here. The graph shows the points at the initial stage is parallel to the vertical axis, which is an indication of a jump from a local optimum. The graph of the MGAPSO shows points where the gradient of the curve is very close to zero, that is where the graph is parallel to the horizontal axis and then the curve becomes almost parallel to the vertical axis. The points where the curve is parallel to the horizontal axis means the global best was in an established local optimum then the algorithm jumps out. This curve means that the algorithm could have the ability to jump out of a local optimum and thus has a higher search ability.

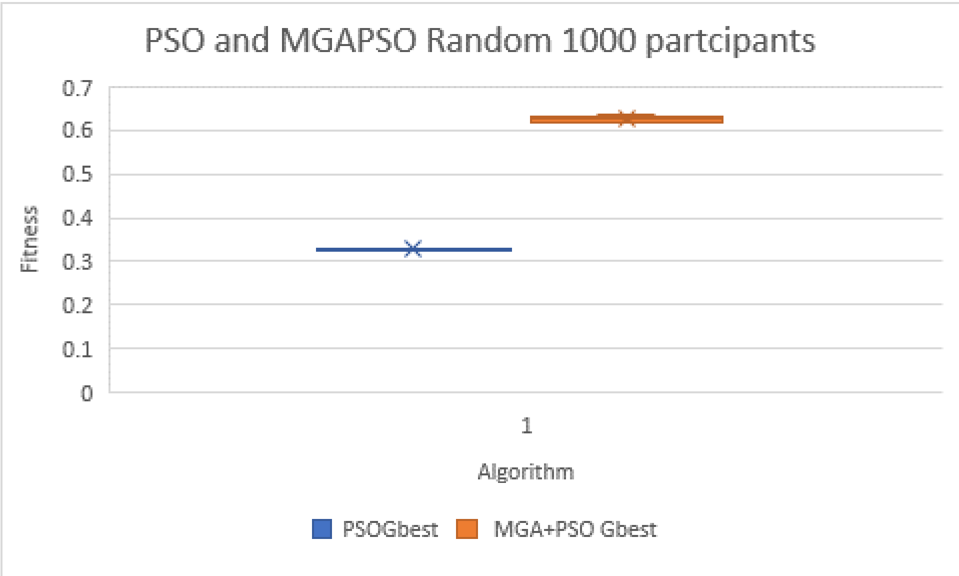


Figure 3.10 Fitness of PSO and MGAPSO with 1000 Uniformly Generate profile

Similar to Figure 3.8, Figure 3.10 shows a better global best fitness obtained by the MGAPSO relative to the PSO. However, this figure shows the PSO has a smaller range among the global best fitness. The PSO obtains a thinner box meaning a smaller range of the values of the global best obtained by the algorithm.

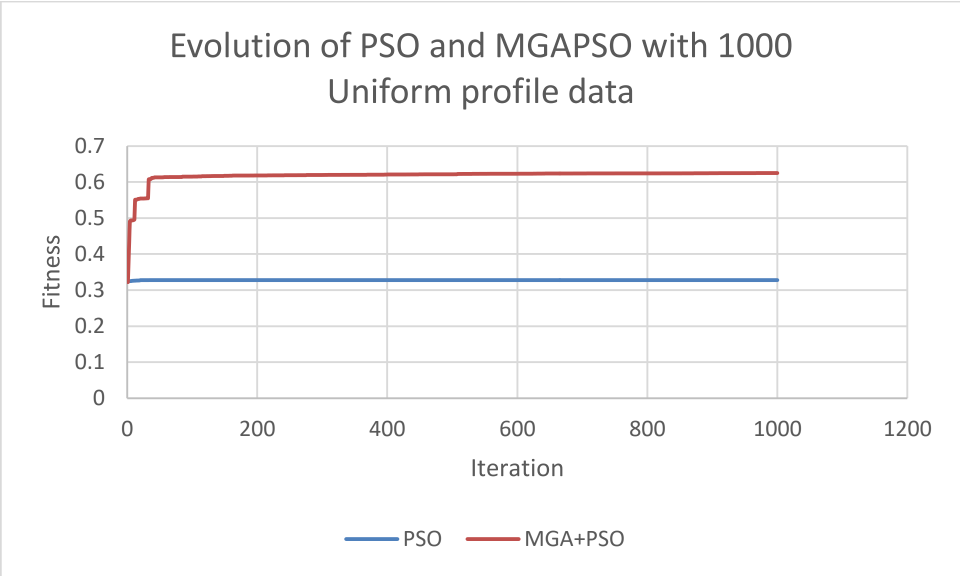


Figure 3.11 Evolution of Global best particle of PSO and MGAPSO with 1000 Uniformly Generated profile

Figure 3.11 shows the curve of the evolution of the global best particles for the MGAPSO and the PSO with 1000 participants with data which follow a uniform distribution. The curves of both algorithms are like the curves in figure 3.9 with the MGAPSO showing jumps as indicated in Figure 3.9, however, the duration of the stay in the local optimum is smaller in this curve than in Figure 3.9. this means the MGAPSO algorithm may be able to jump out of local optimum faster when the number of participants is large. However, this argument needs further investigation.

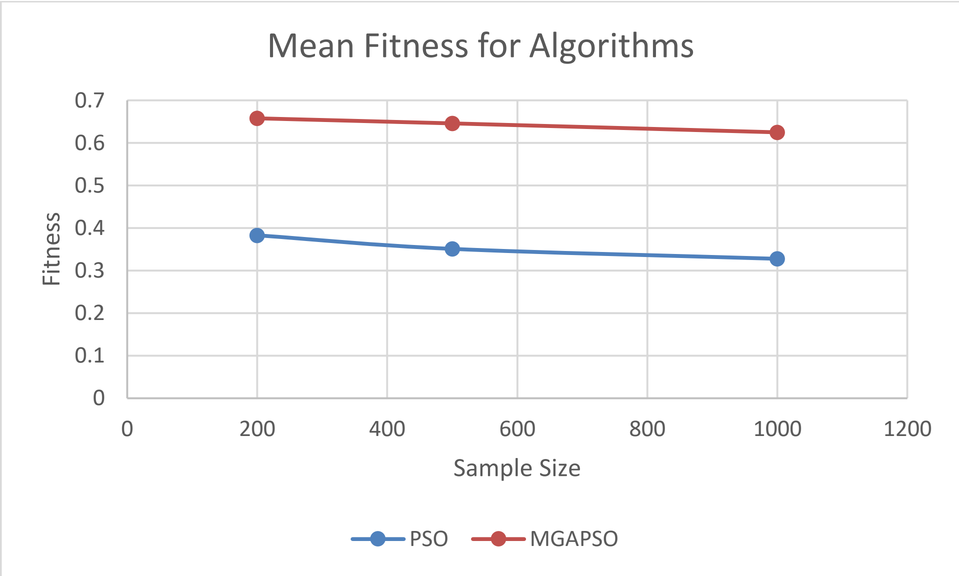


Figure 3.12 Fitness with various sizes of PSO and MGAPSO

Figure 3.12 shows the mean fitness obtain by the PSO and the MGAPSO with different profile sizes which were from a uniform distribution. The graph shows a similar trend of the two algorithms as the number of participant increase. The figure shows that the fitness gradually decreases as the number of participants increased.

Table 3.2 Mean Fitness of Algorithm Uniform data

Mean Fitness of Algorithm Uniform data		
	PSO	MGAPSO
200	0.3829	0.658
500	0.351	0.6461
1000	0.3277	0.6251

The graph above figure 3.8 shows the average fitness of the particle swarm algorithm and the average fitness of the hybrid particle swarm and microbial algorithm (MGAPSO) with uniformly generated learner's data for 200, 500 and 1000 participants. The graphs show that the hybrid algorithm had higher fitness however, as the number of participants increased the difference tends to reduce. Although the hybrid MGAPSO shows a higher mean fitness in the 50 runs, the PSO possesses some outliers which have fitness much greater than any possessed by the hybrid MGAPSO algorithm. This is an indication that the PSO possesses a higher ability to explore the solution space. The range of the fitness of the Hybrid MGAPSO shows that the algorithm tends to converge fast close to the global best found. The table below shows the standard deviation of the fitness of the two algorithms, in all three cases the standard deviation of the hybrid algorithm is much lower indicating a higher stability of the hybrid algorithm (MGAPSO) relative to the PSO.

Table 3.3 Standard Deviation of PSO and MGAPSO

	Standard Deviation for Uniform Data	
	PSO	MGAPSO
200	0.0088	0.0031
500	0.006	0.0068
1000	0.00302	0.00577

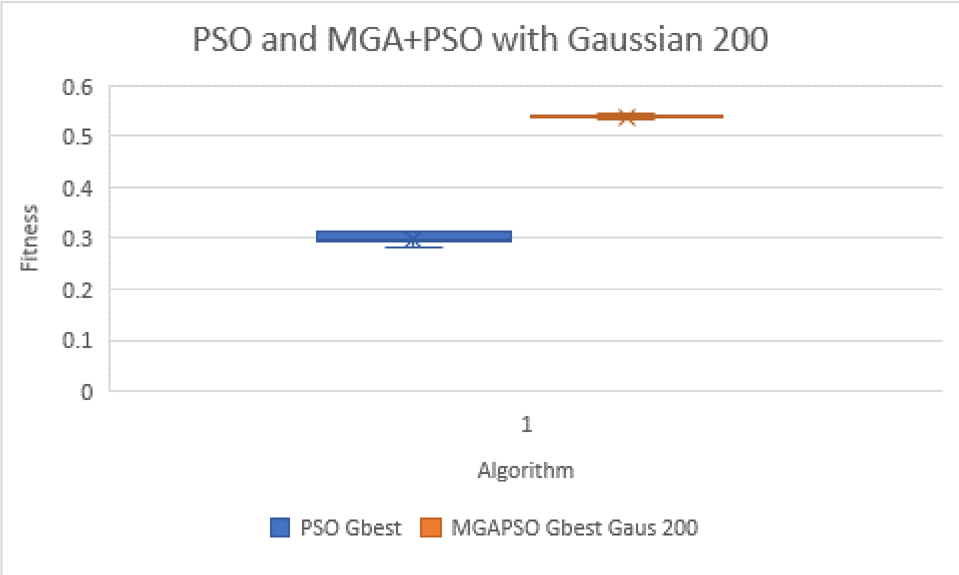


Figure 3.13 Fitness of PSO and MGAPSO for 200 profiles with Gaussian distribution

Figure 3.13 shows the box plot of the PSO and MGAPSO with 200 simulated profile data with Gaussian distribution. The figure shows the MGAPSO showing a higher global best fitness. The MGAPSO also has a smaller range which means a better convergence.

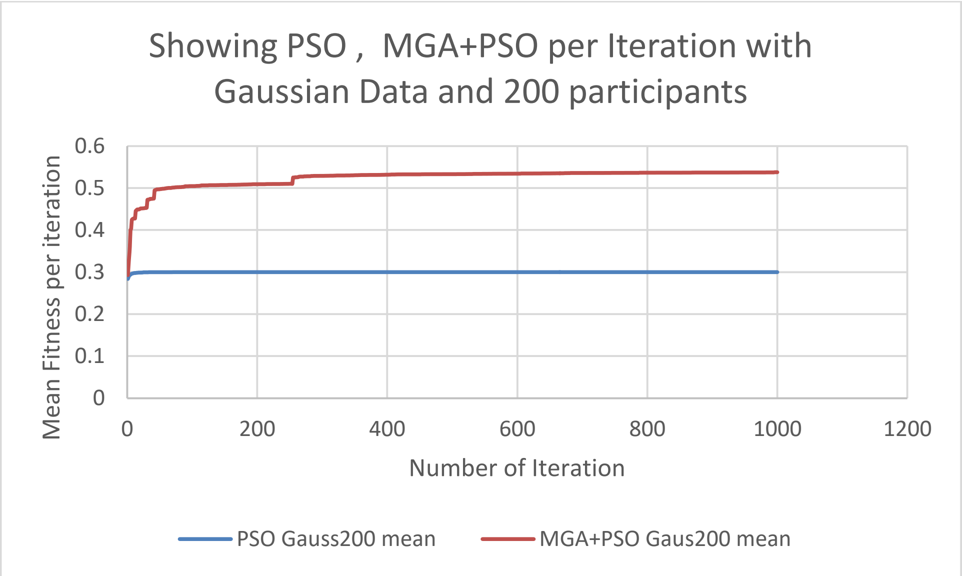


Figure 3.14 Evolution of Global best particle for PSO and MGAPSO with 200 profiles data with Gaussian distribution

Similar to other figures earlier discussed, in figure 3.14 the particle swarm algorithm gets trapped in a local optimum which the algorithm could not get out off however, the graph of the MGAPSO show several point where the MGAPSO jumped out of local optimum. The algorithm still shows a slight improvement with the gradient very close to zero.

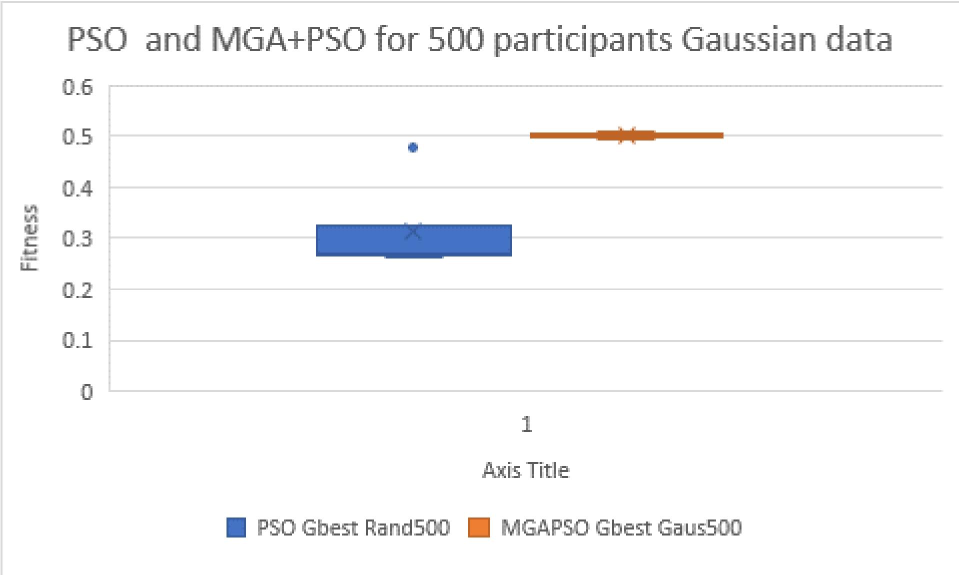


Figure 3.15 Fitness of PSO and MGAPSO with 500 profile data with Gaussian distribution

Figure 3.15 show box plot of the mean fitness of PSO and MGAPSO, the MGAPSO show a very small range among the global best of the algorithm and is an indication of the higher stability of the algorithm relative to the particle swarm algorithm.

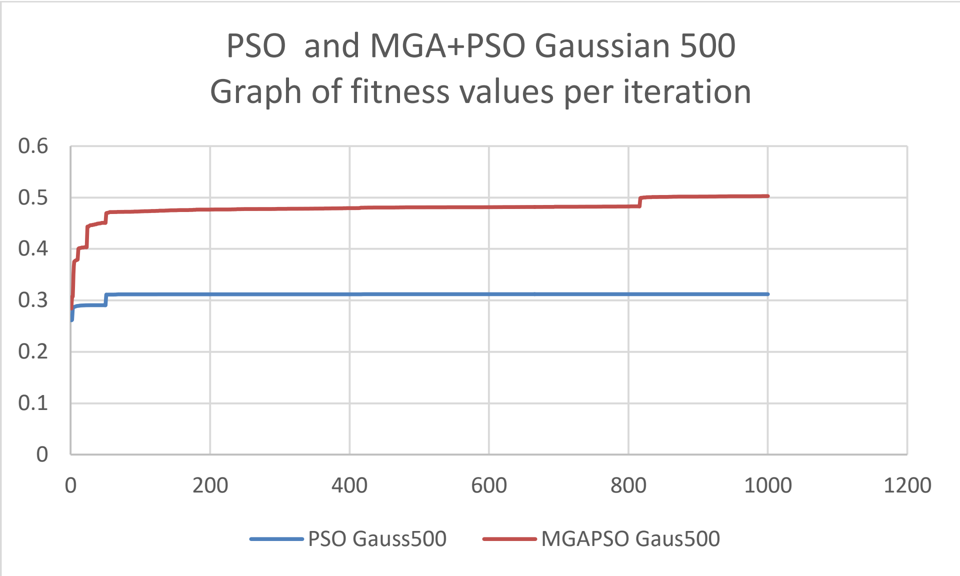


Figure 2.16 Evolution of Algorithms with 500 profiles with Gaussian distribution

Figure 3.16 shows the evolution of the global best particles of the PSO and the MGAPSO with a population size of 500 that follows a Gaussian distribution. The PSO shows a small jump at the 48 iterations, while the MGAPSO, similar to other graphs discussed earlier, shows several jumps out of some local optimum.

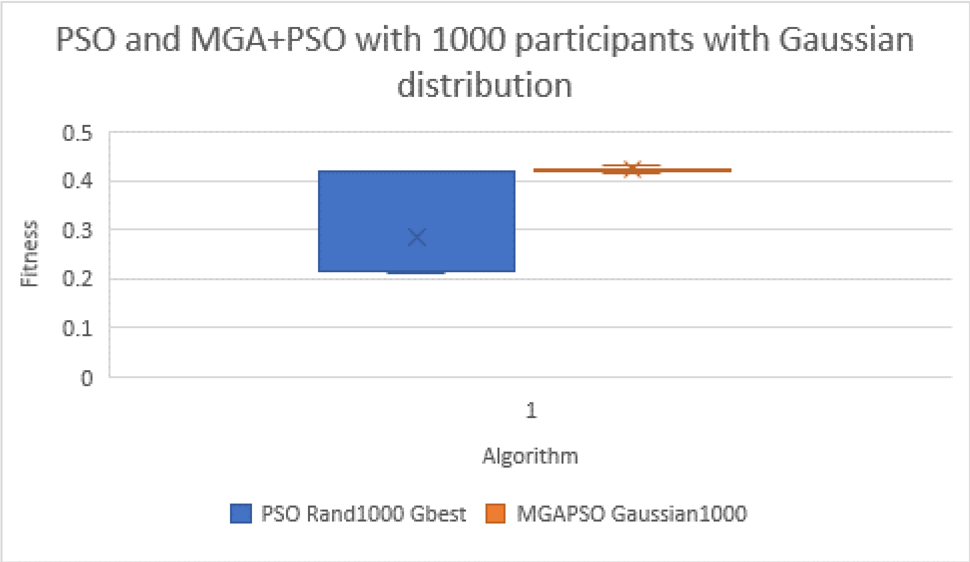


Figure 3.17 Fitness of PSO and MGAPSO with 1000 profile data with Gaussian distribution

Figure 3.17 show the PSO and MGAPSO with 1000 participants. The box plot if the MGAPSO is thin meaning the range of the global best is very small while the box for the PSO wider indicates a wider range of fitness.

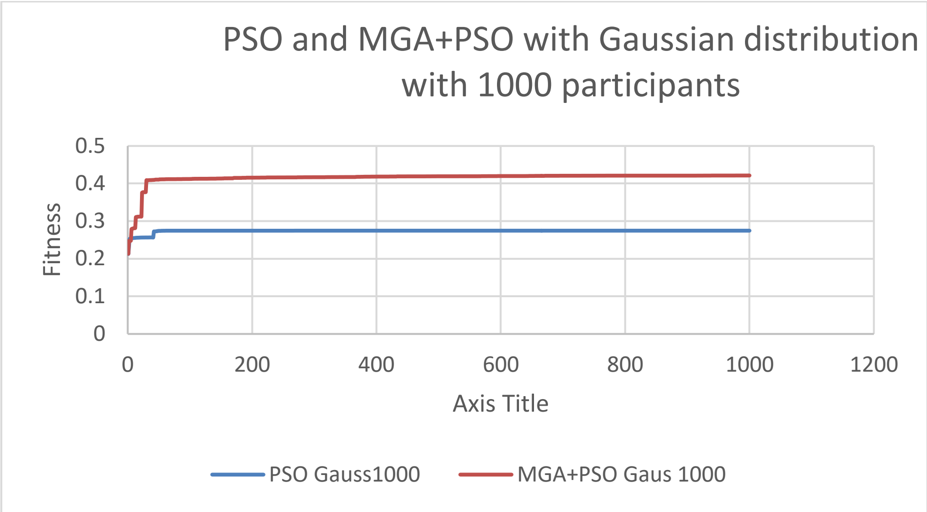


Figure 3.18 Evolution of Global best particle of PSO and MGAPSO with 1000 profile with Gaussian distribution

The evolution of the global best of the PSO and MGAPSO with 1000 participants with a Gaussian distribution is shown in figure 3.18. The graph is similar to the graph in figure 3.16. The PSO had a single jump after which it remained in the optimum found while the MGAPSO had several jumps.

Table 3.4 mean Fitness for PSO and MGAPSO for data with Gaussian distribution

Mean Fitness of Algorithm with Gaussian Distribution		
	PSO	MGAPSO
200	0.2998	0.5377
500	0.312	0.5026
1000	0.2744	0.4212

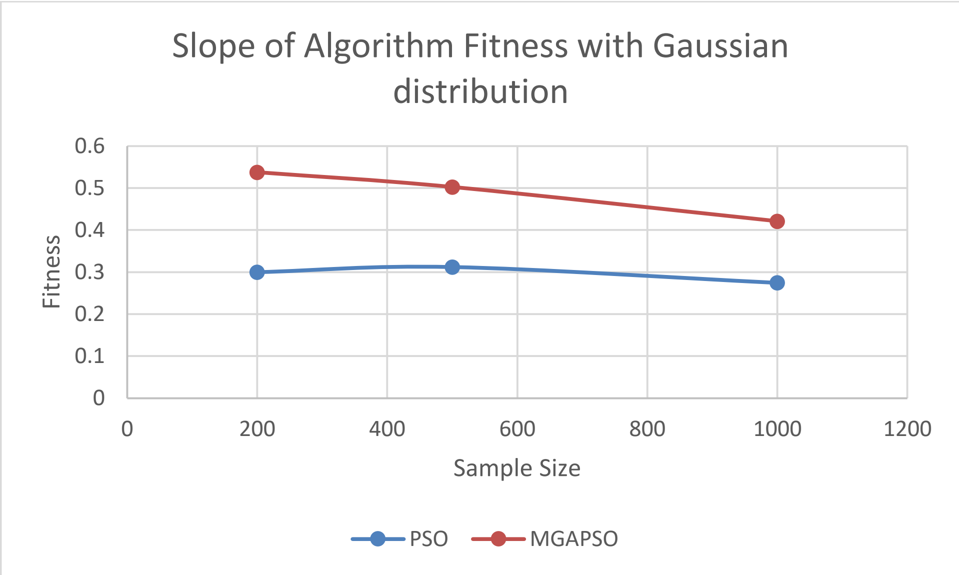


Figure 3.19 Graph of Fitness for data with Gaussian distribution for PSO and MGAPSO

Table 3.5 Standard Deviations for profiles with Gaussian distribution

Gaussian Data Standard Deviation		
	PSO	MGAPSO
200	0.010178	0.002728
500	0.08470	0.004586
1000	0.09609	0.004113

Figure 3.19 shows the graph of the mean global best of the particle swarm and the MGAPSO with the three sample sizes. The graph is a demonstration of the performance of the two algorithm. However, the results from figures 3.16 to 3.18 shows that the MGAPSO outperformed the PSO. In addition, the experiment demonstrated the ability of the MGAPSO to outperform the PSO and jump out of local optimum although the PSO also jumped out in figure 3.16 and figure 3.18.

Table 3.3 and table 3.5 show the standard deviation of the global best particle fitness of the two algorithms (PSO and MGAPSO) with uniformly distributed profile data and profile data with Gaussian distribution. With the uniformly distributed profile data, both MGAPSO algorithms showed very low standard deviation among the mean of the fitness of their global best particle fitness this is an indication of high stability however, it is worth noting that at 200 the MGAPSO had a lower standard deviation than the PSO but at 1000 the standard deviation had a higher standard deviation. This could mean that the MGAPSO has a higher exploratory ability thus although having a better fitness than the PSO the algorithm still had a higher standard deviation than the PSO at 1000. In contrast to table 3.3, table 3.5 shows a slight variation, in table 3.5 the standard deviations for MGAPSO in 500 and 1000 are lesser than the standard deviations for the PSO. This slight variation is an interesting thing to note from the experiment which is worthy of further investigation.

Although the results show the MGAPSO outperforming the PSO there is the need to examine the degree to which the MGAPSO outperformed the PSO. Table 3.6 shows the percentage increase of the MGAPSO relative to the PSO. The values range from 53.49 percent for the Gaussian with 1000 participants to 90 percent in the 1000 participants with Uniform distribution. This values shown in table 3.6 are the increase on the PSO which the MGAPSO achieved.

Table 3.6 shows the percentage increase in the fitness of the MGAPSO relative to the PSO however there is the need to ascertain the if there is statistical significance in the means of the two algorithms. To answer this question, statistical analysis is done using analysis variance to test for statistical significance in the mean fitness of the two algorithms. For each of the distributions and the simulated data sizes stated.

Table 3.6 Percentage increase of MGAPSO with respect to PSO

Data distribution and size	Mean fitness of PSO	Mean fitness of MGAPSO	Percentage improvement
Uniform 200	0.382907	0.658021	71%
Uniform 500	0.351063	0.646167	84%
Uniform 1000	0.327763	0.625065	90%
Gaussian 200	0.29988	0.537709	79.3%
Gaussian 500	0.312005	0.502675	61.1%
Gaussian 1000	0.2744	0.4212	53.49%

3.8 Validation/ Statistical Analysis

Test Hypothesis

Although the results of the comparative experiment above showed that in all sample sizes used with the profile data of uniform and Gaussian distribution the MGAPSO outperformed the PSO with varying percentage increase in the PSO by the MGAPSO, there is the need to verify this claim (Mcgeoch, 2012).

A null hypothesis and an alternative hypothesis was stated for each case and an ANOVA one-way test was conducted, the hypothesis were stated as

Null hypothesis (H0): mean(PSO) = mean (MGAPSO) (There is no significance in the means of the two algorithms)

Alternative Hypothesis (H1): mean(PSO) ≠ mean (MGAPSO) (There is significance in the means of the two algorithms)

An ANOVA one-way test was conducted for all experiments at 5 percent significance level and the results of the hypothesis are summarised in table

3.8. However, tables 3.7a and 3.7b show example of the hypothesis results and data for profile size of 200 and 500 with Gaussian distribution.

Table 3.7a Summary of ANOVA Analysis for Algorithms with 200 profile with Gaussian distribution

ANOVA: Single Factor

SUMMARY Gaussian 200

Groups	*Experiments*	*Sum*	*Average*	*Variance*
PSO	10	2.998802	0.29988	0.000115
MGAPSO	10	5.37709	0.537709	8.27E-06

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.282813	1	0.282813	4584.355	3.98E-23	4.413873
Within Groups	0.00111	18	6.17E-05			
Total	0.283923	19				

Table 3.8 shows a summary of the ANOVA statistics for all hypothesis tested, the p-value for all the hypothesis were found to be less than 0.05, the hypothesis of the means for 1000 participants with Gaussian distribution had the highest p-value of 0.009 which is less than 0.05 while other p-values ranged from 2.55 E-06 to 3.98E-23. Thus, the null hypothesis in all cases was rejected and the alternative hypothesis accepted in all cases, meaning the mean fitness of the MGAPSO in all the experiment was higher than the mean fitness for the particle swarm algorithm which is an indication that the MGAPSO could be a better algorithm in this instance.

Table 3.7b Summary of ANOVA Analysis for Algorithms with 500 profile with Gaussian distribution

	Mean Fitness of PSO	Mean Fitness of MGAPSO	P values	F values	F critical values	Remarks Null Hypothesis
Uniform 200	0.382907	0.658021	1.64E -14	4310.302	4.9646	Rejected
Uniform 500	0.351063	0.646167	1.84 E-20	7371.559	4.60011	Rejected
Uniform distribution 1000	0.327763	0.625065	2.34 E-13	8310.379	5.317655	Rejected
Gaussian 200	0.29988	0.537709	3.98E-23	4584.355	4.41387	Rejected
Gaussian distribution 500	0.312005	0.502675	2.55 E-06	45.4661	4.413873	Rejected
Gaussian distribution 1000	0.2744	0.4212	0.009151	10.37825	4.964603	Rejected

Table 3.8 Summary Table of ANOVA Results

ANOVA: Single Factor

SUMMARY Gaussian 500

Groups	*Experiments*	*Sum*	*Average*	*Variance*
PSO	10	3.120047	0.312005	0.007973
MGA+PSO	10	5.026752	0.502675	2.34E-05

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	0.181776	1	0.181776	45.4661	2.55E-06	4.413873
Within Groups	0.071965	18	0.003998			
Total	0.253741	19				

3.9 Discussion of Experiment

The purpose of the experiment in this chapter was to compare the new MGAPSO algorithm with existing particle swarm optimization algorithm in the learning group formation, particularly with increased numbers of participants. Learners profile were simulated with simulated data following a uniform distribution and Gaussian distribution. Two hundred, five hundred and one thousand number of profile data were used in the instances of the uniform and Gaussian data distribution. The results showed the MGAPSO showing higher fitness value than the particle swarm algorithm in all data sets with all data sizes. In addition, the graphs showing the evolution of the global best of both algorithms show the ability of the MGAPSO to jump out of local optimum more than the PSO.

ANOVA one-way test analysis was conducted to ascertain if there is any significant difference in the means of the two algorithm, the p-values for all ANOVA test indicate a high significance in the means of the two algorithm which suggest that the MGAPSO outperformed the PSO.

3.10 Summary and Conclusion of the Chapter

The chapter set out to propose an algorithm for forming collaborative learning groups with large number of participants and to compare the new algorithm with existing algorithm used in the current literature. A brief review of the concept of collaboration was introduced and a new algorithm MGAPSO, a hybrid algorithm of particle swarm and microbial genetic algorithm was proposed. The aims of the grouping were mathematically expressed as the fitness function for the algorithm to evaluate the groupings and a comparative experiment was conducted with simulated data. The results of the experiment showed the new MGAPSO algorithm when compared with PSO outperformed the PSO with the simulated learners profile data used. However, there is still room for improvement of the algorithm, because the parameters of the algorithm were kept constant. An adaptive MGAPSO is thus suggested.

Chapter 4: Adaptive MGAPSO

4.1 Introduction

In the previous chapter of this thesis the hybrid MGAPSO algorithm was proposed, the algorithm was compared with the particle swarm optimization algorithm in a comparative experiment in which the MGAPSO outperformed the PSO. However, the parameters of the algorithm were static thus need to be modified by the user when the algorithm is to be used in different environment. Similarly, parameters need to be determined in each domain in which the algorithm will be used thus the need for an adaptive algorithm.

According to the Cambridge English dictionary adaptation is a process in which an object modifies itself as the conditions around it changes and when an object has this ability it is adaptive. An adaptive algorithm thus is an algorithm which can modify itself as the environmental conditions change. Self-adaptation is a strategy in which particle reconfigure themselves accordingly to suit a problem without user interaction (Van den Bergh & Engelbrecht, 2002) particles could adapt based on changes in their environment, population, individual or component referred to as environmental, population and individual or component adaptation (Wang et. al., 2013) respectively. During individual adaptation the individuals in the swarm adapt, while in population adaptation when the features or parameters of the population changes the population changes. An example of individual adaptation is a case where an individual particle in the swarm could change its own parameters without interference with other individuals in the swarm while an example of population adaptation is where swarm variables (number of individuals in the population, global best particle) are changing. In the parametric adaptation the parameters of the algorithm are modified by the algorithm for the use of all individuals of the swarm (Wang et. al., 2013). An example of parametric adaptation in the particle swarm is a scenario where the particle swarm on its own modify the inertia weight, cognitive learning factor or social learning due to changes in the number of iteration or some outcome of the behaviour of the particles during run time of the experiment.

4.2 Types of Adaptations

Adaptation in the particle swarm algorithm literature can be classified based on the method in which the adaptation is implemented in the algorithm. These adaptations are based on the particle position (Ardizzon et. al., 2015; Beheshti & Shamsuddin, 2015), fitness of the particle (Aghababa et. al., 2010), percentage success and error (Nickabadi, et. Al., 2011) and success rate of the swarm (Wang et al., 2015). Similarly, Carlise and Dozie, (2000) and Yang et. al., (2007) used velocity in the adaptation of the particle swarm algorithm. However, the technique which has dominated the research community is the use of inertia (Suresh et al., 2008; Tang et al., 2015; Pornsing et. al., 2015; Liang et al., 2015; Arumugam & Rao, 2007; Hu et. al., 2013; Ghosh et. al., 2010; Feng et. al., 2007; Yang et. al., 2007; Zhang et. al., 2014; Panigrahi, et. al., 2008; Jiao et. al., 2008) other studies which have used inertia weight are (Chatterjee & Scarry, 2004; Saber et al 2006; Wang et al., 2015; Rezazzadeh et. al., 2011; Hu et al. 2015; Kiani and Pourtakdoust, 2015), each of these studies have used the inertia weight only for adaptation and have changed the inertia weight using different formulae (see chapter two of this thesis).

Although a review of adaptation in the particle swarm algorithm has been conducted in chapter two of this thesis however, a revisit of some adaptive algorithms is required due to their similarity with the adaptive mechanism proposed in this chapter. The algorithms of Saber, et al., (2006) and Aghababa, et al., (2010) are of interest due to their similarity to the method used in this research.

A fuzzy adaptive particle swarm optimization technique was proposed by Saber, et. al., (2006) in which the inertia weight of the swarm is adjusted using some fuzzy rules based on the diversity of the fitness of the swarm, the algorithm used the current location and the inertia weight to determine the inertia weight of the particle in the next iteration using IF/THEN rules. The algorithm does not calculate a variable, rather it used to predetermine values among which the fuzzy rule is used to determine the next inertia weigh of the particle. In this research although, IF/THEN rules are used there are no predetermined parameters from which the parameters are determined rather the parameters of the particle for the next iteration are calculated from the particles the particle learns from which are determine by the fitness of the particle.

The cognitive and social components of the particle swarm algorithm were redefined in Aghababa, et. al., (2010) and used for adaptation. In this method, the fitness of particle best minus the fitness of the particle was introduced as a local adaptive coefficient while the fitness of global best minus fitness of the particle as global adaptive coefficients while in this research the average of the parameters of the particles from which the particle learn from are used to determine the inertia weight, cognitive and social coefficients.

The adaptive mechanism proposed by Carlisle and Dozier, (2000) was also of interest due to its resemblance to the method used in this research. The authors periodically reset the iteration count and triggered a reset each time there is some amount of change in the environment. The method is applied in this algorithm where each time there is a change in the fitness of a particle its fitness count is reset to zero, meaning the particle position has changed and each time the new global best particle is found a reset of the count of the global best is reset to 1.

4.3 AMGAPSO and MGAPSO

The new adaptive hybrid Microbial genetic algorithm and particle swarm optimization algorithm (AMGAPSO) differs from the MGAPSO, the parameters (inertia weight, cognitive and social coefficients) are constant in the MGAPSO while in the adaptive hybrid Microbial genetic algorithm and particle swarm optimization algorithm (AMGAPSO) the parameters change to suit the particle and the environment. In MGAPSO the number of iterations is predetermined which is used as a condition for the termination of the algorithm. On the other hand, the adaptive AMGAPSO the terminal condition is determined by the behaviour of the particles in the swarm.

The major difference between the adaptive AMGAPSO and the MGAPSO is the ability of the particles in the adaptive algorithm to adaptively change their individual parameters. This is to enable the particle search for better solution. In the MGAPSO fixed number of iterations are set. The adaptive MGAPSO (AMGAPSO) will prevent the setting of parameters in the algorithm.

4.4 Adaptive AMGAPSO Algorithm

4.4.1 Overview of AMGAPSO

The particles of the adaptive AMGAPSO are initialised uniformly as the PSO and MGAPSO described in chapter three with the pbest of all particles and gbest initialised, then at the beginning of the first iteration mate selection, crossover and mutation operations are performed and new position, pbest and gbest are updated. The algorithm has some differences to the static version, while in the static MGAPSO the parameters of the all particles in the PSO component of the algorithm were the same, in the adaptive AMGAPSO each particle had its own unique parameter thus the parameters of each particle were stored with the particle. The velocity method was modified such that the parameters of the particle are accepted as input into the method to calculate the velocity. The position update equation as in the particle swarm and the MGAPSO are used to update the position of the particle and the new fitness of the particle is calculated and compared with the previous fitness. The outcome is then used to determine using IF/THEN rules the formula to calculate the parameters of the particle for the next iteration. The possible outcomes are either the new position is not as good as the previous position, better than the previous position but not as good as its pbest, better than its pbest but less than the known gbest or better than the known gbest of the swarm.

A variable (PFitcount) which counts how many iterations a particle has remained unchanged is introduced. This variable is re-initialised to zero each time the fitness of the particle improves by 1% or more, else the variable is incremented, when this variable is equal to 1% of the number of iterations meaning if the particle does not improve over 1% of iterations the particle is assumed to have been stucked and the particle is passed over to the microbial genetic algorithm component of the algorithm where mate selection is done, crossover and mutation are performed to introduce a new particle in its place, the velocity of the particle is re-initialised, a new generation of the particle now replaces the old parent.

After initialisation, all particles are passed into the microbial genetic component of the algorithm and particles are passed into the PSO component; each particle calculates its velocity and updates it pbest, parameter and Pfitcount. At the next iteration if the PFitcount is equal to 1% of the number of iteration, which means the particle has not change over a given amount of iterations thus a mate will be selected at uniform, crossover and mutation are performed and the parent dies and the offspring then replaces the parent. An argument could be made at this point that the algorithm at the initial stage should behave like the PSO, but this is not the case because the crossover and mutation performed immediately after initialisation enable the particles find better positions and the gbest is updated at this stage, which enables the algorithm to find a good solution and start the PSO component with relatively better gbest and pbest than the ordinary PSO.

In addition, a population variable (GbFitcount) which counts how many iterations the global best remained unchanged, was introduced. Each time a new gbest is found this variable is re-initialised this variable was used to determine the terminal condition of the algorithm. The terminal condition was that the global best gbest does not change over a given number.

4.4.2 Hybridisation in Adaptive MGAPSO (AMGAPSO)

The hybridisation of the microbial genetic algorithm and the particle swarm to form the MGAPSO have been discussed in chapter three of this thesis. In the AMGAPSO the hybridisation method changes while in the MGAPSO hybridisation method was serial (Chang et. al., 2013). In this method all particles are passed into the microbial genetic algorithm and the particle swarm algorithm at each iteration such that a particle does not survive more than a single iteration. At each iteration an offspring of the parent is produced to replace the parent which might be better than the parent or not.

The disadvantage of this method (Chang et. al., 2013) is that some particles might be having the potential of exploring better solutions in subsequent iterations if allowed to continue, meaning particles die pre-maturely. Secondly, due to the pre-mature death in the MGAPSO, the impact of the velocity on the particle is not properly felt.

The new adaptive MGAPSO (AMGAPSO) is accordingly designed to enable the particle to live their live span before death. Particles can explore the search space with their velocity until they can no longer find better location before they die. The adaptivity in this method is that particles die individually without affecting other particles in the swarm. Two alternatives of adaptive hybridisation are proposed here labelled adaptive hybridisation type A and adaptive hybridisation type B.

In adaptive hybridisation type A all particles are passed into the microbial genetic algorithm and the particle swarm however, only particles whose fitness have remained unchanged over a given period experience the microbial genetic algorithm. In this method adaptation is individualised such that each individual particle which does not improve dies and an offspring of the particle is produced using the microbial genetic algorithm to replace the parent thus the replacement of a parent by its offspring is dependent on the individual particle and its behaviour.

With Adaptive hybridisation type B method, the passing of the particles into the microbial genetic algorithm is based on the collective behaviour of the entire swarm, a new generation is formed at the same time. When the global best gbest remains unchanged over a given number of iterations the entire generation is replaced by their offspring thus the entire generation is replaced at a time. In this method each time the swarm gets stuck is counted as a single generation thus the number of generations is used to stop the algorithm. The advantage is that there might be a better search with this method than with type A however, the type A might take shorter time to execute.

4.4.3 Mate Selection, Crossover and Mutation

Mate selection in the adaptive AMGAPSO is by uniform selection as obtained in the static MGAPSO explain in chapter three of this thesis however, mate selection in the adaptive AMGAPSO is only done when the particle has been stuck for some specified number of iterations.

4.4.4 Parameter Adjustment Rules

The concept of crossover and mutation will be used by each particle to adaptively generate its individual parameters for the next iteration based on its current position. Mutation in this case will be the multiplication with a uniform number within 0 and 1, while crossover is by finding the average of the parameters of the particles the particle is learning from. Four conditions are identified and used in the parameter adaptation. However, the evaluation of parameters for adaptation is done after the new position of the particle has been evaluated.

The conditions and the particles they learn from are stated as;

(a.)The new fitness could be less than or equal to the fitness of the particle in the previous iteration which means the particle did not improve or the particle new position is worse than its previous position. $F_p^k =< F_p^{k-1}$ this is an indication that the new position is not as good as its immediate past position, hence the particle learns from it particle best and the global best; the parameter for the next iteration becomes the average of the parameters of the particles it is learning from multiplied by a uniform number. The multiplication by the uniform is to cause variation (mutation) so that the particle will not have the same parameters.

$$W_P^{k+1} = (W_{Pb}^k + W_G^k)\frac{r}{2}$$

$$C_p^{k+1} = (C_{Pb}^k + C_G^k)\frac{r}{2}$$

$$C2_P^{k+1} = (C2_{Pb}^k + C2_G^k)\frac{r}{2}$$

(b.) The new fitness of particle is better than the fitness of the particle in the previous iteration and is less than or equal to the fitness of its particle best at the previous iteration $F_p^{k-1} < F_p^k =< F_{pb}^{k-1}$ in this case the particle will learn from the its parameter, its particle best and global best particle thus the parameters for the next iteration is calculated by finding the average of the parameters of its own parameters, its particle best parameters and the parameters of the global best which is multiplied by a uniform number.

$$W_P^{k+1} = (W_P^k + W_{Pb}^k + W_G^k)\frac{r}{3}$$

$$C_p^{k+1} = (C_P^k + C_{Pb}^k + C_G^k)\frac{r}{3}$$

$$C2_P^{k+1} = (C2_P^k + C2_{Pb}^k + C2_G^k)\frac{r}{3}$$

(c.) The new fitness of the particle is greater than the fitness of the particle best and less than the fitness of the global best. $F_{pb}^{k-1} =< F_p^k < F_G$ this is where the current particle fitness is higher than the particle best before current movement, this means there is an improvement, it also implies that a new particle best position has been found thus the particle will learn from its current parameters and the parameters of the global best. The parameters of the particle for the next iteration will then be the average of its current parameters and the parameters of the global best multiplied by a uniform number.

$$W_P^{k+1} = (W_P^k + W_G^k)\frac{r}{2}$$

$$C_p^{k+1} = \left(C_P^k + C_G^k\right)\frac{r}{2}$$

$$C2_P^{k+1} = (C2_P^k + C2_G^k)\frac{r}{2}$$

(d.) The new fitness is greater than the existing global best fitness $F_p^k > F_G$ this implies that a new global best has been found thus the particle will learn from its current parameters. This means the parameters for the next iteration will be its current parameter multiplied by a uniform number.

$$W_P^{k+1} = \left(W_P^k\right)r$$

$$C_p^{k+1} = (C_p^k)r$$

$$C2_P^{k+1} = (\ C2_P^k\ )r$$

Where

F_p^k =	*Fitness of Particle P at iteration k*
F_{pb}^{k-1}	*Fitness of particle best of particle P at iteration k-1*
F_G	*Fitness of global best particle*
W_P^{k+1}	*Inertial weight of particle P at iteration k+1*
W_P^k	*Inertial weight of particle P at iteration k*
C_p^{k+1}	*Cognitive learning factor of particle P for iteration k+1*
C_p^k	*Cognitive learning factor of particle P for iteration k*
$C2_P^{k+1}$	*Social learning factor of particle P for iteration k+1*
$C2_P^k$	*Social learning factor of particle P for iteration k*
r	*Uniform number between 0 and 1*
W_G^k	*Inertial weight of Global best particle at iteration k*
C_G^k	*Cognitive learning factor of Global best particle at iteration k*
$C2_G^k$	*Social learning factor of Global best particle at iteration k*
W_{Pb}^k	*Inertial weight of particle best of particle P at iteration k*
C_{Pb}^k	*Cognitive learning factor of particle best of particle P at iteration k*
$C2_{Pb}^k$	*Social learning factor of particle best of particle P at iteration k*

After initialisation of particle, particle best and global best crossover and mutation are performed, update of particle position, pbest, gbest, parameters and all other personal and global variables is performed. Velocity is updated, and new position calculated, new updates performed, and this loop continues depending on the type of hybridization used in the algorithm. However, the algorithm of adaptive hybridization types A and type B are stated below.

4.4.6 Adaptive MGAPSO Algorithm with Adaptive Hybridisation Type A

Initialization (particles, particle parameters, gbest, pbest)

Microbial genetic algorithm

Select particle mate

Perform Crossover and mutation

Update Pbest, Gbest

While termination condition (GFitcount<Iteration)

For every particle

If (pfitcount> = 1% of iteration) {

Crossover, mutation; reset pfitcount=0}

EndIf

Calculate velocity, Update position,

Calculate new fitness, Update pfitcount, Update pbest,

Update gbest and GFitcount

Update particle parameters for next iteration

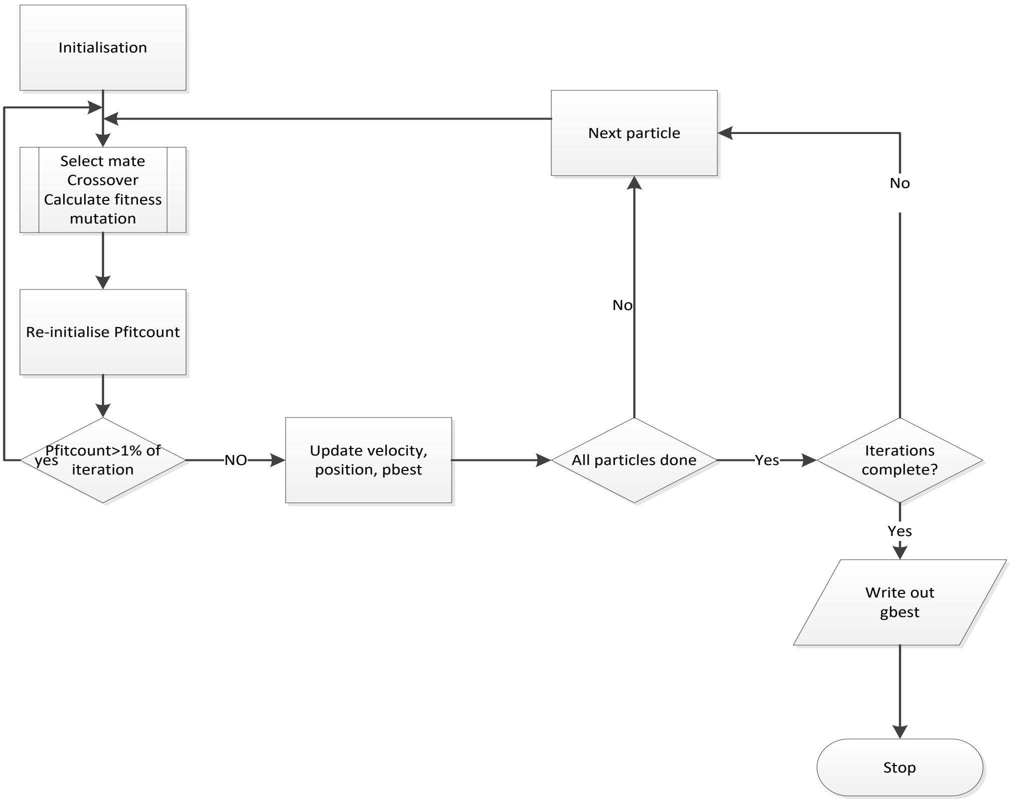
EndFor

EndWhile

Write Gbest to file;

End

Stop



4.4.7 Adaptive MGAPSO Algorithm with Adaptive Hybridisation Type B

Initialization (particles, particle parameters, gbest, pbest)

Microbial genetic algorithm

Select particle mate

Perform Crossover and mutation

Update Pbest, Gbest

While termination condition (GFitcount< generations)

For every particle

Select mate

Perform Crossover, Perform mutation;

Normalise particle, gbest, pbest

EndFor

While (gbestcount2 <10% of generations)

Calculate velocity, Update position

Calculate new fitness, Update pfitcount

Update pbest, Update gbest and GFitcount,

Update particle parameters for next iteration

Endwhile

Update generations

EndWhile

Write Gbest to file;

End

Stop

4.5 Experimental Design

4.5.1 Overview of Experiment

In this study hybridization Type-A was used. The hybridization type A was used because it took less time and the experiment was conducted with a computer with processor: AMD A8=7410 APU with AMD Radeon R5 Graphics 2.0 GHz; Installed memory (RAM) 8.00 GB (6.95 GB usable) System Type 64-bit Operating system, x64-based processor.

The experiment was conducted to investigate the ability of the proposed new adaptive hybrid algorithm could be used in the group formation; the sample sizes of 200, 500 and 1000. Fifty runs were performed with each algorithm with each sample size and the mean of each was used to represent the result for each algorithm.

4.5.2 Purpose of Experiment

The purpose of this experiment is

- To investigate the use of the new adaptive AMGAPSO could be used in forming groups
- To compare the new AMGAPSO with the MGAPSO algorithm and PSO
- To ascertain the possibility of the adaptive hybridization method in the hybridization of MGA and PSO

Data used for this experiment was data used in the experiments in chapter three of this thesis. Fifty runs were performed with each sample size stated and with each data distribution and the average of the results obtained from the 50 runs for all the algorithms are shown in the result using box and whisker plot iterations/generations are tabulated and the average for all runs per iteration/generation was used to plot the evolution curve of the algorithm. Figures showing the result with box and whisker and the evolution curve for the algorithms are shown below.

4.5.3 Results of Experiment

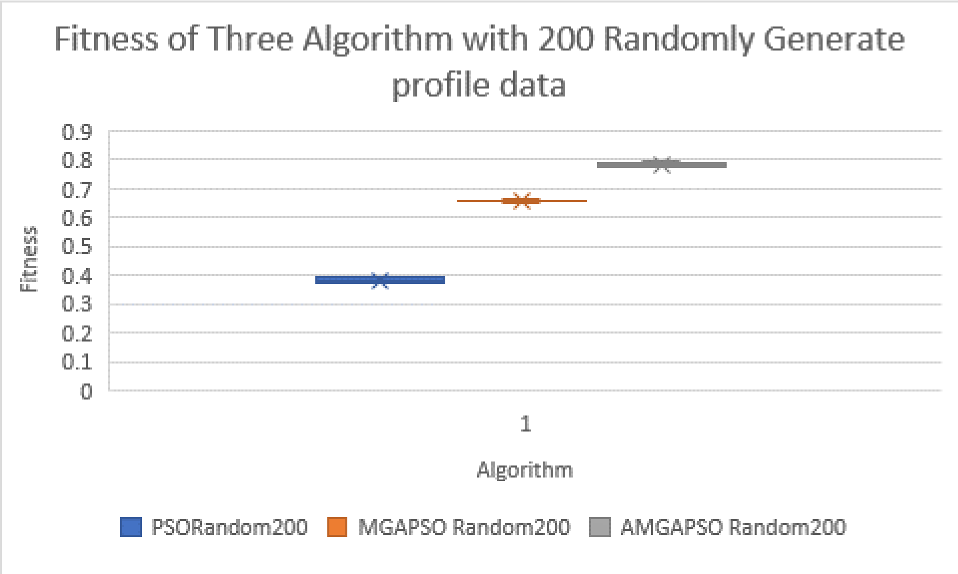


Figure 4.1 Showing Box plot for PSO, MGAPSO and AMGAPSO for 200 Uniformly generate profile

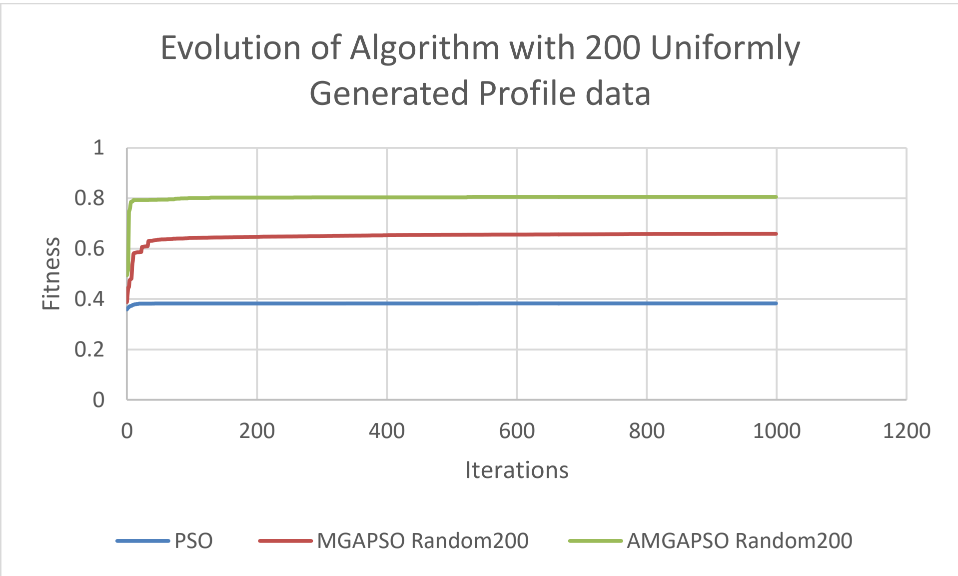


Figure 4.2 Showing Graph for PSO, MGAPSO and AMGAPSO for 200 Uniformly generate profile

Figure 4.1 shows the average global best of the AMGAPSO compared to the MGAPSO and the PSO however, the MGAPSO have been compared in chapter three thus the AMGAPSO is of primary concern. The AMGAPSO had a higher fitness for the global best

In figure 4.2 the shows the evolution of the global best particle gbest. The AMGAPSO found a much higher fitness within the first few generations and maintained that fitness all through. This might be because of the small number of participants involve the algorithm could find the best it could find within few generations.

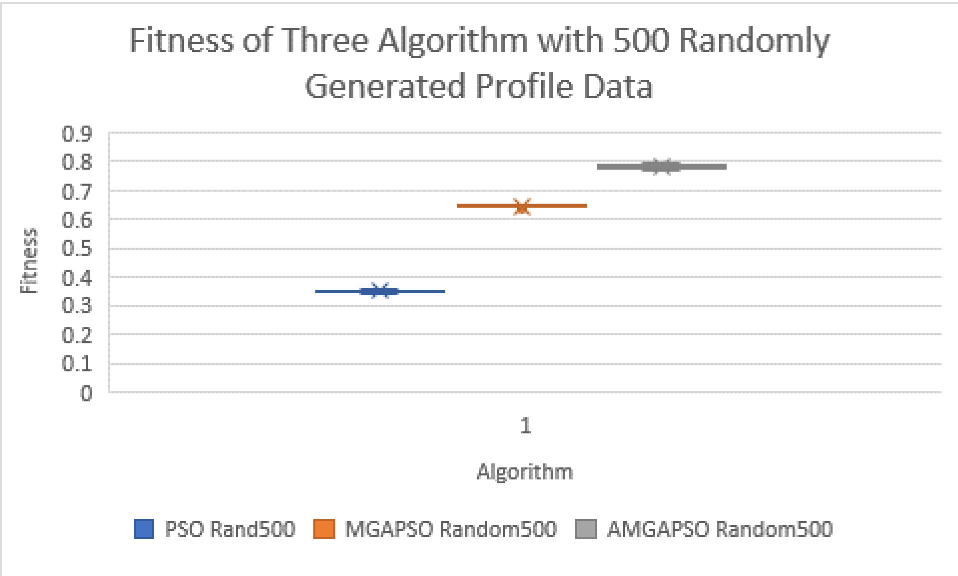


Figure 4.3 Showing Box plot for PSO, MGAPSO and AMGAPSO for 500 Uniformly generate profile

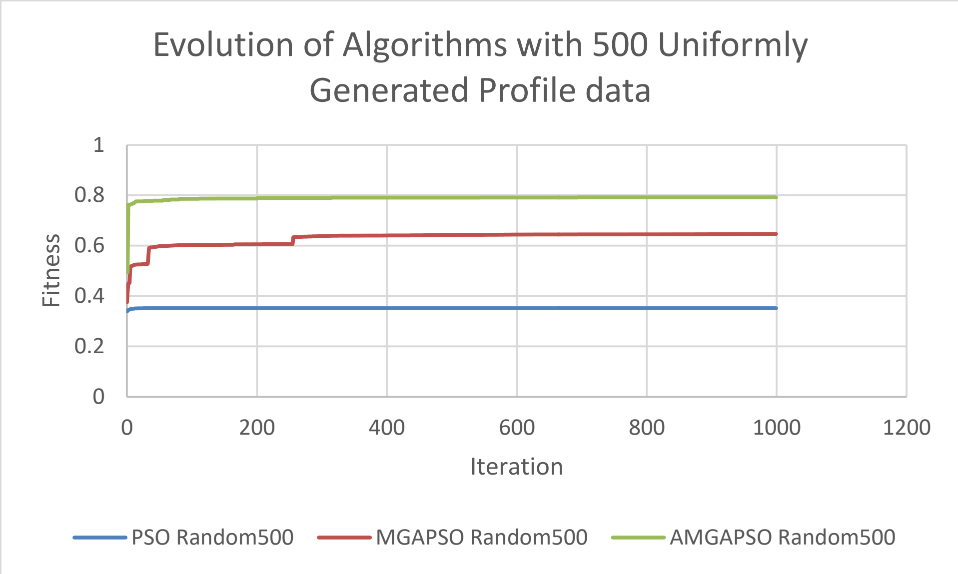


Figure 4.4 Showing Graph of PSO, MGAPSO and AMGAPSO for 500 Uniformly generate profile

Figure 4.4 shows the evolution of the gbest when experimented with 500 uniformly generate profile data with Gaussian distribution. The AMGAPSO found a higher fitness within the first few generations which it maintained.

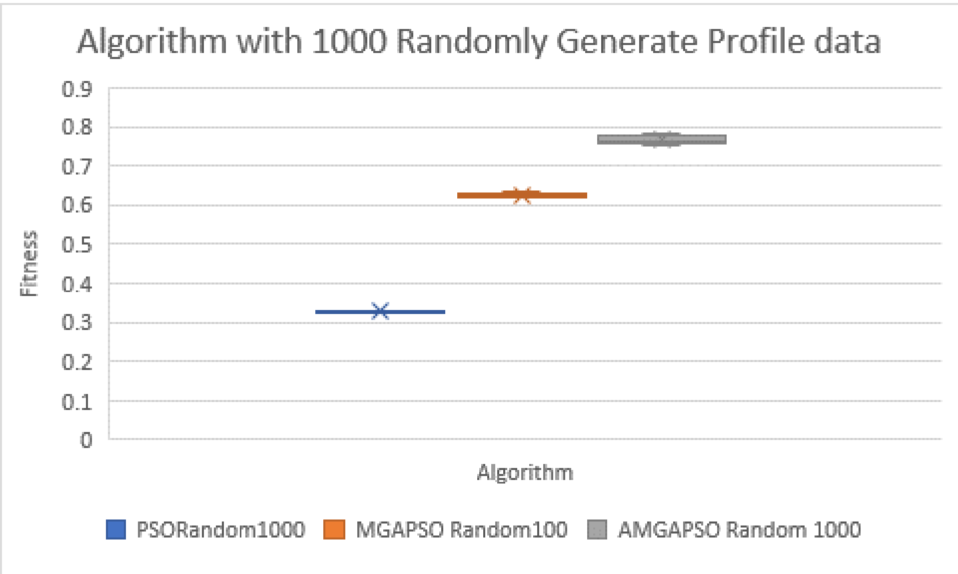


Figure 4.5 Showing Box plot for PSO, MGAPSO and AMGAPSO for 1000 Uniformly generate profile

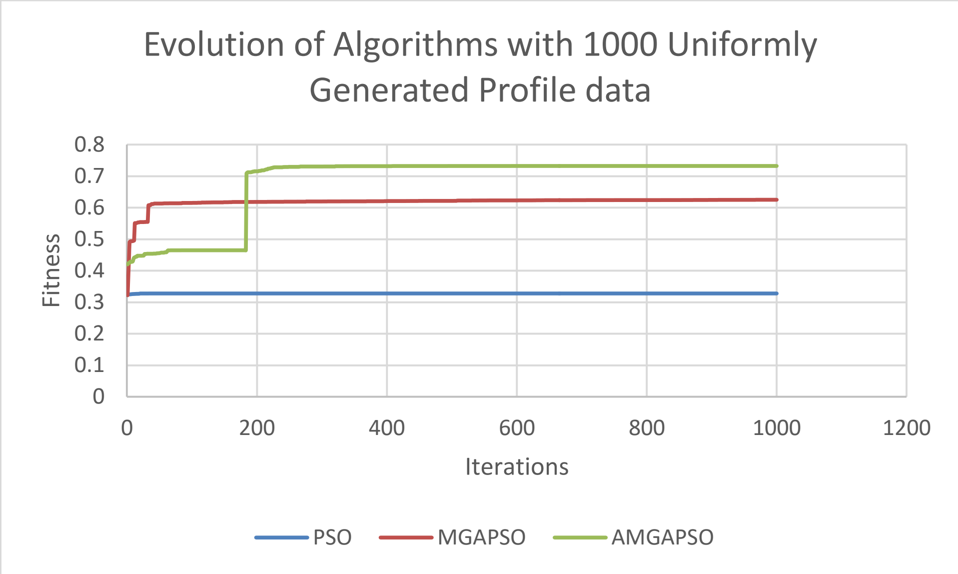


Figure 4.6 Showing Graph of PSO, MGAPSO and AMGAPSO for 1000 Uniformly generate profile

Figure 4.5 shows the fitness of the AMGAPSO and the MGAPSO and the PSO, the AMGAPSO outperformed MGAPSO with higher fitness however in figure 4.6 which shows the evolution of the gbest the AMGAPSO was lower than the MGAPSO but the graph of AMGAPSO algorithm made a vertical movement and found a higher fitness than the MGAPSO; slight improvements were still made after the vertical movement. The algorithm later found a value which it maintained.

Table 4.1 Showing Mean Gbest fitness for PSO, MGAPSO and AMGAPSO

Mean Fitness for Uniform Data

	PSO	MGAPSO	AMGAPSO
200	0.3829	0.658	0.7827
500	0.351	0.6461	0.7816
1000	0.3277	0.6251	0.71565

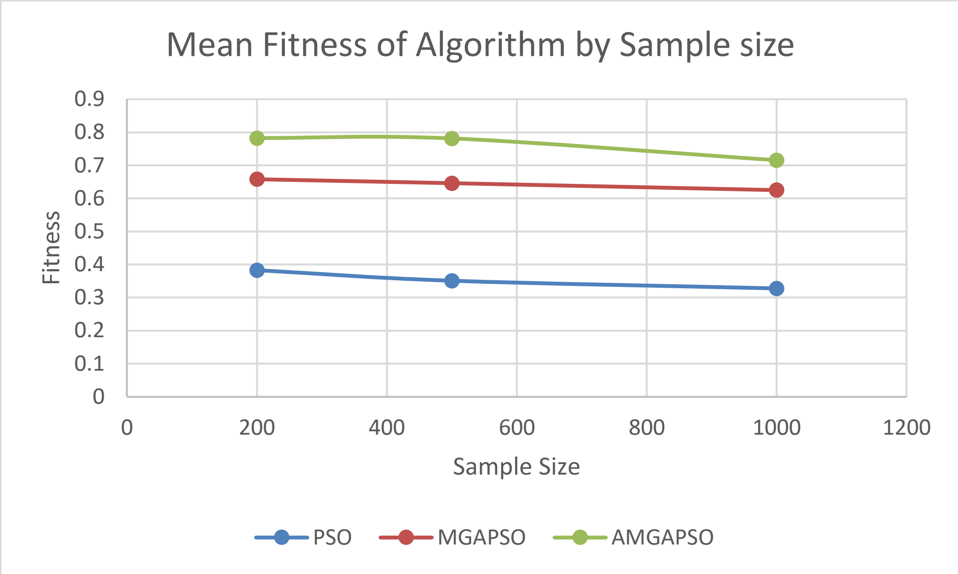


Figure 4.7 Graph showing means of Algorithms

The graph in figure 4.7 shows the means gbest of three algorithms, the PSO, MGAPSO and the AMGAPSO for 200, 500 and 1000. In all three sample sizes the AMGAPSO outperformed the MGAPSO and the PSO.

Table 4.2 Showing SD of PSO, MGAPSO and AMGAPSO Algorithms

	Uniform Data	Standard Deviations	
	200	500	1000
PSO	0.01611	0.025863	0.023451
MGAPSO	0.005419	0.004078	0.003655
AMGAPSO	0.033554	0.03799	0.044338

The standard deviations of the three algorithms particle swarm optimization (PSO), static hybrid MGAPSO and the adaptive MGAPSO algorithms show the static MGAPSO having the lowest standard deviation in all three-sample sizes.

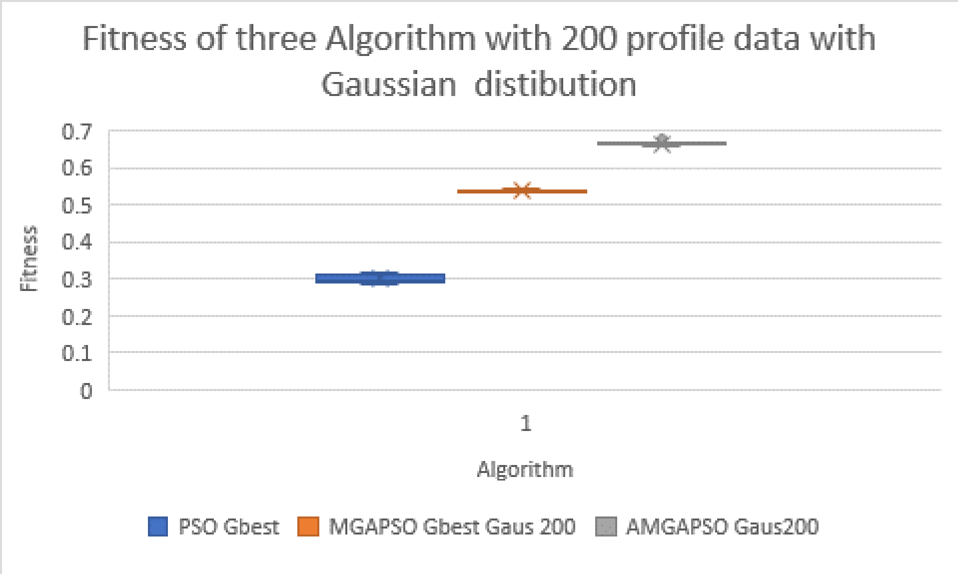


Figure 4.8 Showing Fitness of algorithm with 200 profile data with Gaussian distribution

The AMGAPSO and the MGAPSO algorithms have thinner boxes than the PSO however, the AMGAPSO had higher fitness than MGAPSO and the PSO.

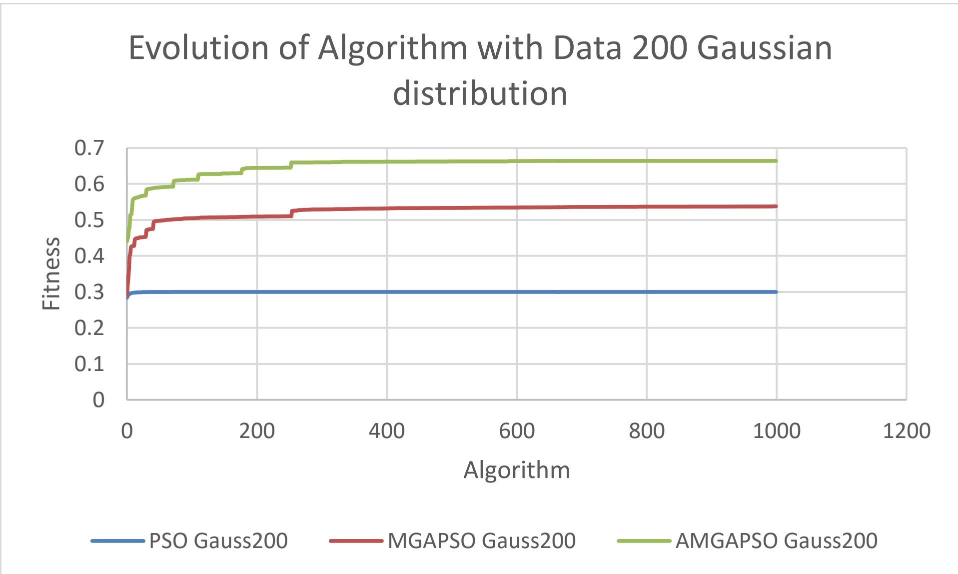


Figure 4.9 Showing Box plot for PSO, MGAPSO and AMGAPSO for 200 generated profile with Gaussian distribution

In figure 4.9 the AMGAPSO have a zigzag climb. In this graph the AMGAPSO gradually ascended, the graph of the AMGAPSO moves up slightly and maintained that fitness for a while and moved up again. It continued till it found its maximum level.

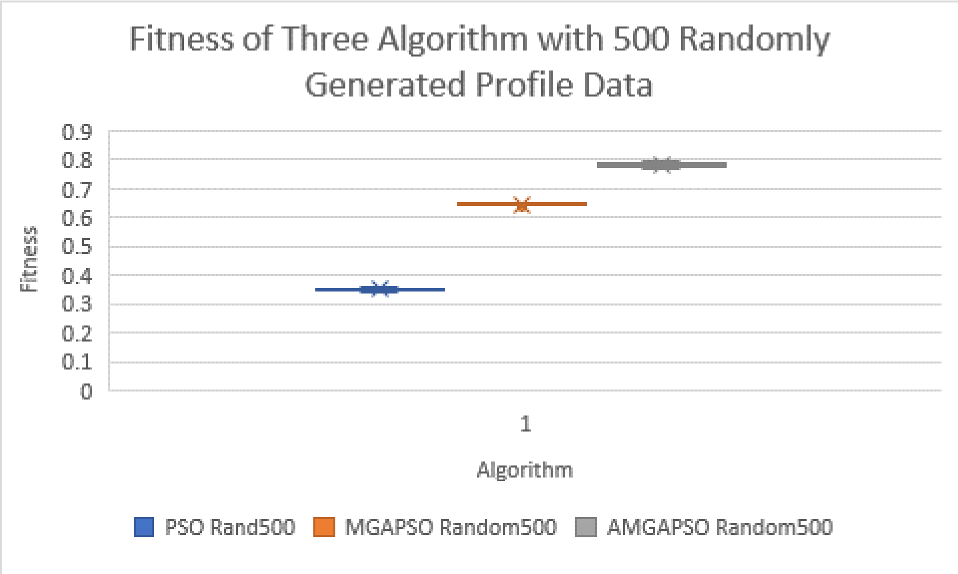


Figure 4.10 Showing fitness of Algorithm with 500 profile data with Gaussian distribution

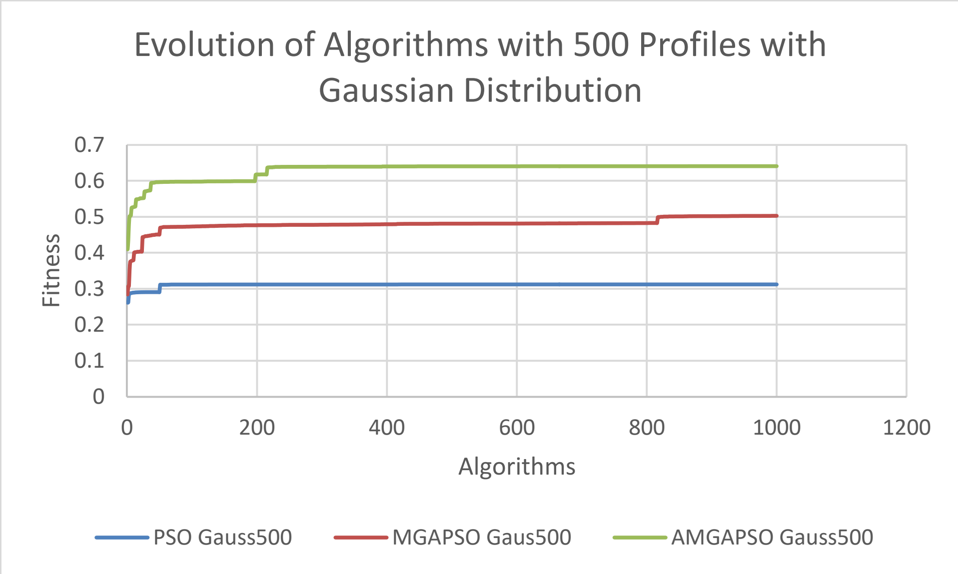


Figure 4.11 Evolution of Gbest for three algorithms with 500 profile data with Gaussian distribution

Figure 4.11 shows the graph of the of the gbest for 500 profile data with Gaussian distribution, the first few generations of the AMGAPSO shows the curve finding increments. Then it maintained a value and the curve parallel with the horizontal axis for some generation and then made some increment again to find the value it maintained till the end of the experiment. Below also is figure 4.12 showing bar graph of the fitness of the used algorithm. The figure 4.12 shows AMGAPSO outperforming both the MGAPSO and PSO when 1000 randomly generate learners data was used.

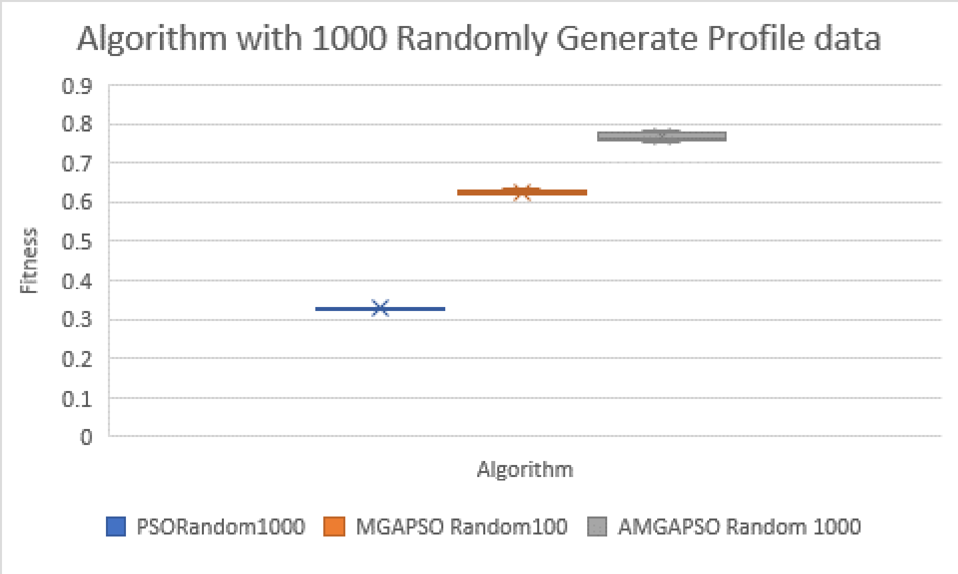


Figure 4.12 Showing fitness for Algorithms with 1000 profiles with Uniform distribution

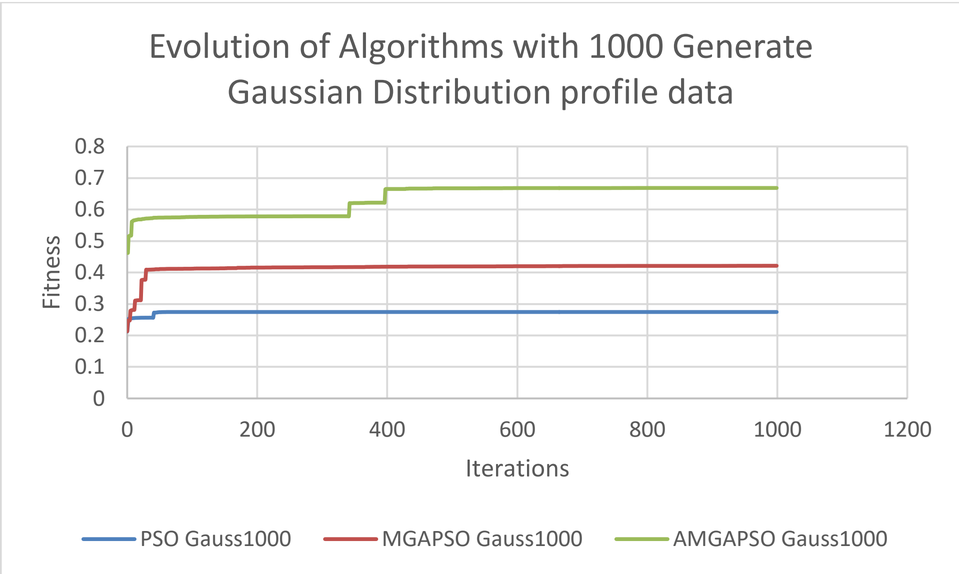


Figure 4.13 Showing Evolution of Algorithms with 1000 profile with Gaussian distribution

Figure 4.13. shows the evolution of the gbest for 1000 profile data with Gaussian distribution, the AMGAPSO found gbest higher than 0.5 within the first few generations and the graph remained parallel to the horizontal axis till after 300 it made slight vertical jumps twice, at 400 it made a vertical movement after which it remained parallel to the horizontal.

Table 4.3 Showing mean fitness for Gaussian data

Mean Fitness for Gaussian Data Distribution			
	PSO	MGAPSO	AMGAPSO
200	0.2998	0.5377	0.6637
500	0.312	0.5026	0.6406
1000	0.2744	0.4212	0.6633

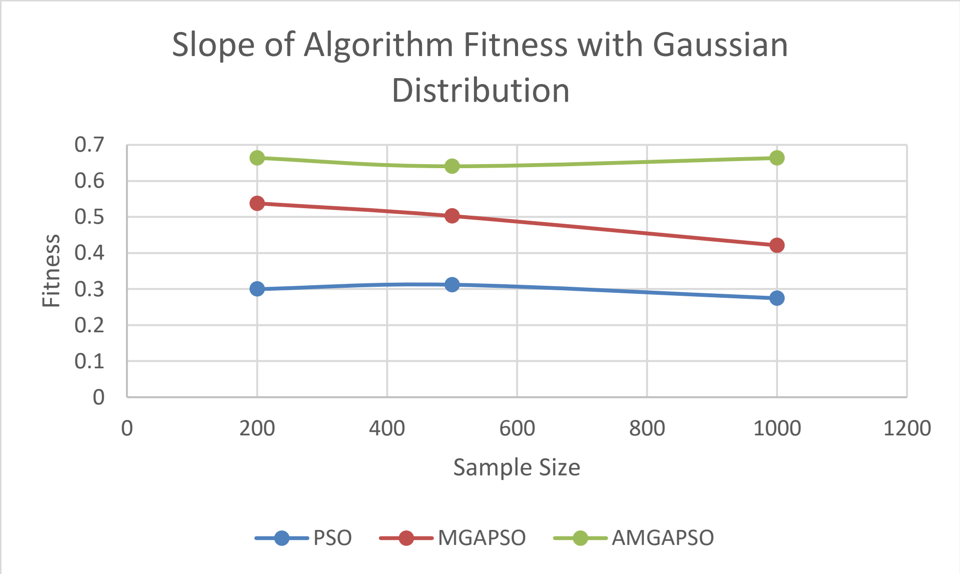


Figure 4.14 Showing Algorithm Fitness with Gaussian distribution

The graph titled "Graph of means of Gbest fitness for 3 algorithms" Fig 4.14 above shows the gbest fitness of the three algorithms for the grouping of 200, 500 and 1000 learners with profile data with Gaussian distribution. The graph of the AMGAPSO tends to move higher when the number of participants increased to 1000.

4.5.2 Stability Test

Table 4.4 Standard Deviation of Gaussian distribution

	Gaussian	Standard deviation	
Size	PSO	MGAPSO	AMGAPSO
200	0.010178	0.002728	0.006
500	0.08470	0.004586	0.0075
1000	0.09609	0.004113	0.006

The standard deviation for the fitness of the three algorithms for population sizes of 200, 500 and 1000 showed a decrease in the standard deviation from adaptive MGAPSO to PSO followed by the hybrid MGAPSO which is an indication of higher stability of the hybrid MGAPSO than all other two algorithms. Similarly, the PSO showed a higher stability that the adaptive hybrid MGAPSO

4.6 Evaluation of Experiment

The experiments above have shown that the AMGAPSO outperformed the MGAPSO in all three sample sizes for each data distribution. However, there is the need to test if there is significant difference in the means obtained by the two algorithms the AMGAPSO and the MGAPSO. A null hypothesis was stated for all data sizes for the data with the uniform distribution and the data with the Gaussian distribution used in the experiments above. The null hypothesis was tested using ANOVA one-way test at a significance level of 0.05 degree of freedom.

The null hypothesis and the alternative hypothesis in all case were stated as follows

H0: the means of the two algorithms are equal (there is no significant difference in the means of the two algorithms).

HA: the means of the two algorithms are not equal (there is significant difference in the means of the two algorithm)

Results of ANOVA Analysis

The p-value, f-value and the F-critical values for the one-way test of ANOVA at a 0.05 significance level are summarised in the table 4.5 below.

Table 4.5 Summary of ANOVA Results

ANOVA Results for MGAPSO AND AMGAPSO				
Data	F value	P value	F critical	Remark
Uniform 200	1432.236	1.3 E -18	4.413873	Reject Null Hypothesis and accept the alternative hypothesis
Uniform 500	954.0098	2.79 E -14	4.60011	Reject Null Hypothesis and accept the alternative hypothesis
Uniform 1000	318.3115	6.54 E -09	4.964603	Reject Null Hypothesis and accept the alternative hypothesis
Gaussian 200	3283.443	7.92 E -22	4.413873	Reject Null Hypothesis and accept the alternative hypothesis
Gaussian 500	2218.127	2.64 E -20	4.413873	Reject Null Hypothesis and accept the alternative hypothesis
Gaussian 1000	4947.819	1.86 E -12	5.317655	Reject Null Hypothesis and accept the alternative hypothesis

Table 4.5 above shows a summary of all the hypotheses tested. In all cases the F-values were greater than their corresponding F-critical values and all P-values are less than 0.05. Both conditions indicate conditions for a rejection of the null hypothesis and the acceptance of the alternative hypothesis. This is an indication of a very high significant difference in the means of the adaptive hybrid microbial genetic algorithm particle swarm (AMGAPSO) and MGAPSO with simulated student profile with Gaussian and uniform distribution. This shows that the AMGAPSO outperformed the MGAPSO

4.7 Discussion of the Chapter

Figure 4.1 shows a box of the AMGAPSO shows a thicker box than the MGAPSO which mean the range of the gbest obtained in the AMGAPSO is higher, however, the least fitness of the AMGAPSO from the box plot show the AMGAPSO outperforming the MGAPSO. In figure 4.2 the MGAPSO showed increments in the fitness while in AMGAPSO the slope at the beginning is parallel to the vertical axis meaning a jump meaning the algorithm jump out of the initial local optimum very fast, it was able to converge very fast. Figure 4.3 shows the AMGAPSO having higher fitness while in figure 4.4 the AMGAPSO converges to a local optimum very fast but the algorithm was still able to improve slightly till at 400 it converged at an optimum which it maintained. The graph in figure 4.6 is of interest because the evolution is slightly different from the graphs previously examined. In figure 4.6 the AMGAPSO made slight improvements then it made a sudden jump from the local optimum, at the new optimum it made slight improvement in the fitness and it then maintained an optimum till the end. This means the algorithm trapped in a local optimum but the algorithm was able to jump out which might have been due to the generation of new offspring resulting in a better offspring generation which were better than their parents.

Figure 4.9 shows the AMGAPSO found many local optimums. The graph could be an indication of the ability of the algorithm to jump out of local optimum, it means many local optimums were found and the algorithm was able to jump out of each shortly. The search capabilities of the AMGAPSO is shown in figure 4.11 similarly to figure 4.9, in figure 4.11 the algorithm shows the jump out of local optimum. The algorithm after been trapped in local optimum the algorithm jumped out. Figure 4.13 shows the AMGAPSO after three areas where the graph is parallel to the horizontal axis each of which is a local optimum, the algorithm might have jumped out due to the death of the parent generation and replacement by the next generation which might have been better than the parent generation.

Table 4.4 shows the standard deviations of the three algorithm using profile data with Gaussian distribution, the table shows MGAPSO as having standard deviations lower than the standard deviations of the AMGAPSO. However, the standard deviations of the AMGAPSO were lower than the standard deviation of PSO showing higher convergence.

Results of ANOVA analysis conducted to test six null hypotheses stated are shown in table 4.5 that at the significance level of 0.05 all null hypothesis were reject which is an indication of the acceptance of the alternative hypothesis meaning there is high significance in the means of the AMGAPSO and MGAPSO. This means the AMGAPSO performs better than the MGAPSO

4.8 Conclusion of Chapter

In this chapter a new adaptive hybrid algorithm was proposed, the proposed algorithm was implemented and compares with the static hybrid algorithm proposed in chapter three. Two levels of adaptation were applied in the algorithm; two new types of adaptive hybridization methods were proposed. An Experiment was conducted to compare the two algorithms, the new adaptive hybrid algorithm (AMGAPSO) was found to outperform the static hybrid algorithm (MGAPSO). The experiment also showed that the adaptive hybridization method proposed could be used in the hybridization of the particle swarm optimization algorithm with the microbial genetic algorithm and that the new adaptive AMGAPSO could be used in the forming collaborative learning groups.

An ANOVA one-way test was conducted to test for statistical significance in the means of the two algorithms AMGAPSO and MGAPSO. The ANOVA one-way test showed high significance in the means of the two algorithms in all dataset used with varying data sizes. The AMGAPSO have thus been shown to outperform the MGAPSO and the PSO however, this algorithm will be tested with real student's data in the next chapter of this thesis.

Chapter 5: Evaluation and Discussion

5.1 Introduction

In the previous chapter of this thesis a new adaptive hybrid AMGAPSO algorithm was proposed and a comparative experiment was conducted to investigate the ability of the algorithm relative to other existing algorithms (particle swarm optimisation algorithm and the static hybrid MGAPSO algorithm) and the new adaptive hybrid AMGAPSO was found to have outperform the other two algorithms in the grouping problem using simulated learners' profiles. However, what has not been done is to evaluate the performance of the algorithm using real learners' data and to confirm the performance of the new adaptive AMGAPSO relative to other algorithms. The use of the real data is essential because it is actual data which might not be of the same distribution as the Gaussian or uniform.

In this chapter, a comparative experiment is conducted using real learners profile data to compare the new adaptive AMGAPSO, static MGAPSO and PSO to ascertain the best among these algorithms in the grouping of real learner's profile data. The resultants groups formed with the new adaptive hybrid algorithm are further analysed using a one-way ANOVA test.

The rest of the chapter is arranged as follows section 5.2 contains the experimental design, section 5.3 discusses the quality of groups formed while discussion of the research is contained in section 5.4. In section 5.5.and 5.6 the finding and the benefits of the research are discussed and the chapter is concluded with section 5.8 before which is section 5.7 which discusses alternative methods that could be applied in the research.

5.2 Experimental Design

5.2.1 Overview of Experiment

The experiment was a comparative experiment in which the new adaptive hybrid AMGAPSO was compared to the PSO, the static MGAPSO. Fifty (50) runs for each algorithm (PSO, static MGAPSO, adaptive MGAPSO) was conducted, the results obtained were plotted together for comparison.

To test for the quality of the groups all groups generated by the algorithm were outputted, each attribute of each group was separated then the means of each attribute for all the groups was analysed to test if the means of the groups have significant difference.

5.2.2 Purpose of Experiment

The purpose of this experiment is

- To compare the new adaptive AMGAPSO algorithm with PSO, and the static MGAPSO algorithms using real learners' profiles
- To ascertain the quality of groups formed by the new adaptive AMGAPSO

5.2.3 Data Source

Data used in this experiment is the data of 500 students collected from the Niger Delta university in Nigeria. I chose to collect the data from this university because it will be easier for me to obtain such data from that university than from any other university as this is a university I work for and it will be easier for me to obtain approval for the collection of the learners' profile data.

Leaners Interest in a learning object have been found to be characterised by increased attention, concentration and affect performance positively (Hidi, 2006). Interest levels in this scenario was measured as the number of take home assignment given to learners in which learners were told they will never be graded. This was a way the university from which the data was collected rated interest levels of their student. The argument is that learners who are interested may try to complete such assignments due to their interest when they know the assignments will not be graded. Knowledge level in this case was taken as a measure of the performance of the learners in learning topics in in which the learners were graded. Lecturers had announced test for assessment weeks before the assessment were conducted and the grades for such announced assessment was taken as the knowledge level for each learner.

To obtain this learners data an application was sent to the Head of department of the department of mathematics and computer science of the Niger Delta University in Nigeria. Upon receipt of approval/ permission (Appendix B) to use the required data ethical approval was sorted for from the ethics committee of the department of computer science in the college of engineering, design and applied sciences of Brunel university. The letter of approval from the ethics committee (Appendix D) was submitted to the head of department of mathematics and computer science of Niger Delta university, Nigeria, who then authorised the release of the learners' data. The learners' data were obtained from the module leader of a module GST 100 who anonymised the data by removing everything in the data that could lead to the identification of any student. The registration numbers (unique identity numbers) of the learners, the year of assessment and everything which could lead to the identification of any student were removed before handing over the learners' profile data.

5.2.4 Data Pre-processing

The data received from the university were not pre-processed in any form rather they were used in the experiment exactly in the form they were received. Sample of data received from the university is shown in table 5.1

Table 5.1 Showing Sample learners data from university

Topic1 interest level	Topic1 understanding level	Topic2 interest level	Topic 2 understanding level	Topic3 interest level	Topic3 understanding level
4	7	6	5	6	5
5	4	5	4	6	7
5	5	4	5	4	6
6	3	5	5	4	4
5	6	6	2	2	5
6	5	5	6	5	4
7	4	5	5	3	3
8	6	5	6	5	4
6	4	4	6	5	3
7	7	5	6	5	3

The students' profile data was examined to ascertain the distribution of the learners' profile data. The EasyFit software was employed to test the distribution of best fit using Anderson Darling test. The Anderson Darling fitness test has the advantage that it is used to test the fitness of a data set to of any distribution, another is that it uses the entire data set and it is sensitive to all types of deviation from normality and more sensitive to deviations at the tails. In addition, with the EastFit software the critical values for samples need not be calculated. Other distribution test including Kolmogorov Smirnov test, Shapiro-Wilk test (Shapiro & Wilk; 1965) were examined by the software. Interested readers could see (Anderson & Darling; 1952)

A null hypothesis and an alternative hypothesis were formulated as

The null and the alternative hypotheses are:

- H_0: the data follow the normal distribution;
- H_A: the data do not follow the normal distribution.

The result of the test for the distribution of the learners' profile data is displayed in the table 5.2 below.

Table 5.2 Result of the Test for the Distribution of the Learners' Profile Data

Distribution	Parameters
Normal	σ =0.95082 μ =4.9167

However, the calculated value of the data show σ =0.95082 μ =4.9167 where μ is the mean and σ is the variance of the dataset. Reference to tables of critical values for the Anderson-Darling Test found at http://www.real-statistics.com/statistics-tables/anderson-darling-test-table/ at the 0.05 confidence level the result shows that the null hypothesis be rejected haven gotten a value of σ =0.95082 which is greater than the critical value. Appendix C shows detail result of Anderson Darling test result as generated from the EasyFit software. The result indicates that the learners profile data is not normally distributed.

5.2.5 Results of Chapter Experiment

The learners profile that was then used in the comparative experiment of the three algorithms (Particle swarm optimization; hybrid Microbial genetic algorithm +particle swarm optimization algorithm (MGAPSO) and the adaptive hybrid AMGAPSO. Ten runs of the experiment were conducted with each algorithm and a box plot of the gbest obtained all three algorithms is shown in figure 5.1.

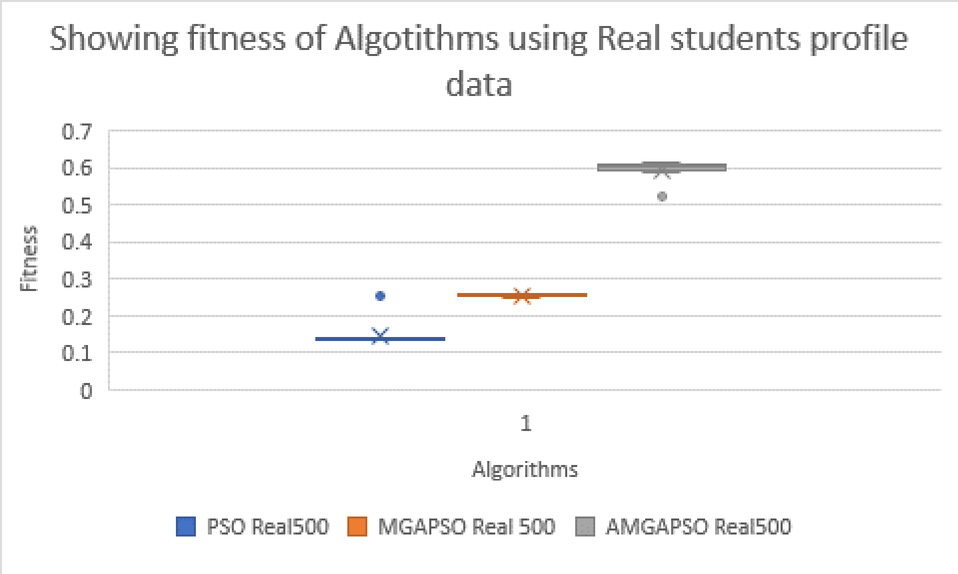


Figure 5.1 Showing Fitness of Algorithms with Real Learners Profile data

Figure 5.1 shows the box plot of the performance of PSO, MGAPSO and AMGAPSO in a comparative experiment, the result showed that the AMGAPSO outperformed the PSO and the MGAPSO.

Example of groups formed by the algorithm are shown below in figure 5.2 and 5.3

Topic1 interest level	Topic1 understanding level	Topic2 interest level	Topic 2 understanding level	Topic3 interest level	Topic3 understanding level
5	7	5	5	1	5
2	3	4	4	3	1
4	6	2	1	7	7
3	4	1	3	2	6
6	5	6	7	5	3

Figure 5.2 showing Group of 5 Generate by Algorithm

Topic1 interest level	Topic1 understanding level	Topic2 interest level	Topic 2 understanding level	Topic3 interest level	Topic3 understanding level
4	7	6	5	6	5
5	6	5	3	5	2
1	4	4	5	2	6
6	1	2	7	4	5

Figure 5.3 Showing Group of 4 Generates by Algorithm

5.3 Quality of Groups

To ascertain the quality of the groups formed by the adaptive hybrid MGAPSO algorithm the groups formed with 500 students profile was analysed. The five hundred students' data profile were used due to my inability to obtain an actual class of 1000 student. One hundred unequal groups were formed with group size ranging from four (4) to six (6) The student profiles had 6 features (interest levels for 3 topics and knowledge level for 3 topics) and for all groups the features of the profiles were separated and for all one hundred groups each feature was separated. A null hypothesis and an alternative hypothesis was tested for each interest level and knowledge level for all three topics thus six hypothesis and alternative hypothesis were stated as

ANOVA RESULTS FOR REAL STUDENTS GROUPING TESTING FOR QUALITY OF GROUPS

ATTRIBUTE 1

H1: There is significant difference in the means among the means of the interest level in topic 1 in all one hundred groups

H0: There is no significant difference in the means of all groups interest level in topic 1 in all one hundred groups

ATTRIBUTE 2

H1: There is significant difference in the means among the means of the knowledge level in topic 1 in all one hundred groups

H0: There is no significant difference in the means of all groups knowledge level in topic 1 in all one hundred groups

Hypotheses were stated and tested for interest levels and knowledge level were stated for all three topics and excel was used in the ANOVA one-way test. The p-values, F-value and F-critical values obtain are tabulated in table 5.3.

Table 5.3 Summary of ANOVA-One Way Test Result

	F-Values	P-Values	F-Critical Values	Remarks
Interest level topic 1	0.474283	0.998981	1.386324	Accept the Null Hypothesis
Knowledge level topic 1	0.789337	0.995108	1.390105	Accept the Null Hypothesis
Interest level topic 2	0.776741	0.856764	1.390105	Accept the Null Hypothesis
Knowledge level topic 2	0.724596	0.913399	1.390105	Accept the Null Hypothesis
Interest level topic 3	0.880869	0.696906	1.390105	Accept the Null Hypothesis
Knowledge level topic 3	0.643277	0.968308	1.390105	Accept the Null Hypothesis

All ANOVA analyses were carried out using Microsoft excel 2013, with significance level was set at 0.05.

Table 5.1 shows the ANOVA analysis where the mean of the knowledge level of topic 1 for all groups formed, with p-value obtained of 0.998981 which is greater than the significance level of 0.05. This means the null hypothesis is accepted, the F-value greater than the significance level, it also showed a F-values of 0.474283 which is less than the F-critical of 1.38632. both conditions of p-value and F-values satisfy conditions for the acceptance of the null hypothesis. This means there is no significant difference in the means of the interest levels in topic 1 for all 100 groups.

Table 5.3 row 2 shows the ANOVA one-way test analysis of the understanding levels for all groups on topic 1, with a F value of 0.789337 and a F-critical value of 1.390105 similarly, the p-value obtained was 0.995108 greater than the significance level. This indicates an acceptance of the null hypothesis showing that there is no significant difference in the means of the interest levels in topic 1 in all one hundred groups.

The ANOVA test for the interest levels for topic 2, the means of the interest levels for topic 2 of the one hundred groups were tested for the hypothesis with the null hypothesis that all the means are equal. The results show that F value as 0.776741 with F-critical value of 1.390105 showing a F value less than the F-critical similarly, the p-value obtained was 0.856764 which is greater than the significance value of 0.05. Both p-value and F values indicate conditions for acceptance of the null hypothesis meaning there is no significant difference in the means of the interest levels of topic 2 for all one hundred groups. This shows that the interest levels for topic 2 of all one hundred groups are equal.

Table 5.3 shows the test for the knowledge levels for topic 2. This tables also show conditions for acceptance of the null hypothesis with the p-value as equal to 0.913399 greater than the significance level and a F value of 0.724596 which is less than the F-critical (1.390105) of the test. Both conditions support the acceptance of the null hypothesis meaning the knowledge levels for topic 2 of all one hundred groups are equal.

The ANOVA analysis for interest level and knowledge level for the topic three from the table show p values of 0.696906 and 0.968308 respectively and their F-values as 0.880869 and 0.643277 respectively while both have F-critical value of 1.390105. The results for the interest levels and knowledge levels for topic show conditions for the acceptance of the null hypothesis because the p-values for both hypotheses are greater than 0.05. The calculated F values support the acceptance of the null hypothesis as the F value is equal to 0.880869 and 0.643277 respectively which are all less than the F-critical in both cases; this is also a condition for the acceptance of the null hypothesis. The p-value for all hypothesis tested were greater than 0.05. In addition, the F values for all hypothesis were less than their corresponding F-critical values. Thus, all null hypothesis was accepted. This means for all one hundred groups there was no significant difference in the means of all their attributes which implies that all the groups formed by the algorithm are similar

5.4 Discussion of the Research

The benefits of collaborative learning were discussed in chapter two of this thesis. However, these benefits can only be enjoyed when collaborative learning groups are balanced whereby there is diversity within the groups. Maintaining diversity within the groups while maintaining similarity among groups becomes an NP-hard problem particularly when the number of participants increases. The study set out with the aim of developing algorithms for automatically forming balanced learning groups with the following objectives set to contribute towards achieving the stated aim.

a. To identify the gap in the literature
b. To identify in the literature existing algorithms in learning groups formation

c. To develop an algorithm for forming learning group with large number of participants
d. To compare the proposed algorithm with algorithm found in the literature
e. To evaluate the propose algorithm using learners' data

In chapter two of this thesis, a review of the literature was done and the gap in the literature was identified that there is currently no existing algorithm for forming balanced collaborative learning groups with large number of participants however, particle swarm optimization algorithm was identified as the most used algorithm for collaborative learning groups formation in the literature thus the first two objectives were achieved in chapter two of the thesis.

In chapter three a new algorithm (a hybrid algorithm of microbial genetic algorithm and the particle swarm optimization algorithm MGAPSO) was proposed. A fitness function was designed in chapter three for the algorithm which was used by the new MGAPSO and PSO. The fitness function in the study have been use as a measure of the dis-similarity within the groups and similarity among groups. This means that learners in the same group have to vary in interest levels and performances levels however, the different groups have to be similar with the assumptions as stated in chapter three of this thesis that learners interest levels in various topics differ and their performance levels vary.

The MGAPSO was compared with the particle swarm algorithm in a comparative experiment, the results showed the new MGAPSO outperformed the PSO however, a one-way ANOVA was conducted to test for significance in the means of the two algorithms. The result of the hypothesis showed that there was high significance in the means of the MGAPSO and the PSO. Although objectives c and d have been achieved in chapter 3 with the MGAPSO outperforming the PSO, the MGAPSO was not adaptive thus need the parameters to be set by the user. The resulted in the need for an adaptive algorithm. In chapter four an adaptive hybrid microbial genetic algorithm and particle swarm (MGAPSO) was proposed. The new AMGAPSO was compared with the MGAPSO and the PSO in a comparative experiment, the results showed the AMGAPSO outperformed the MGAPSO and the PSO. Although, there can be various measures of success in group selection which includes improvement of the top performers, improvement in the overall mean grade of the group and reduction in the number of failures, the results of the study have been based on the assumption implied by the fitness measure of the study.

The weights used in the particle swarm algorithm and the hybrid particle swarm and microbial algorithms were the standard weights used by the research community and no changes were made however, in the adaptive hybrid microbial and particle swarm algorithm no weight changes were made by researcher. Weights were altered by the adaptive hybrid microbial genetic and particle swarm algorithm based on the changes in the environments.

During the initial experiments 50 runs 100 runs and 10 runs were conducted and it was discovered the mean fitness of the runs was a difference within the range of 0.0002 and 0.0001. In addition, the time duration for a single experiment was an average of 72 hours when the number of simulated data was 1000. Thus the number of runs for the static algorithm was set at 10.

Learning styles and personality measures are additional measures which could be used for group selection. However, these measures were not used in this study due to the difficult of obtaining this measures. Obtaining a measure of personality would require a personality test to be conducted on the students which will require direct contact with the learners in addition to the cost of the researcher travelling severally to Nigeria where the data was obtained. It was also challenging to obtain this data from the UK universities, hence the use of interest levels and understanding levels as stated earlier in the research.

Although the data of actual students used for the experiment with the proposed adaptive hybrid algorithm was limited to 500 because the number of learners in the modules from which this data was collected had only 500 learners, there will be no additional cost of collecting addition data due to the data collection method used.

5.5 The Findings of the Study

The findings in chapter three indicated that the static hybrid MGAPSO algorithm outperformed the PSO and the ANOVA (one way) showed that the mean fitness of the hybrid MGAPSO differ significantly from the mean fitness of the particle swarm algorithm with all simulated learners profile data. A possible explanation for this might be that the microbial genetic algorithm which re-initialise the particles at every iteration as the particle location/position changes after every genetic operation increased the diversity of the particle. This finding suggests that the hybridisation of the PSO and MGA could be responsible for better exploration of a hybrid algorithm.

An adaptive hybrid AMGAPSO algorithm was proposed in chapter four of this thesis and a comparative experiment of the adaptive algorithm with PSO and MGAPSO using simulated learners' data was conducted. The result showed the new adaptive hybrid AMGAPSO outperformed the static hybrid MGAPSO and the PSO. It is possible that particles in the adaptive algorithm could search the solution space better than the particle in the static hybrid MGAPSO algorithm since only the particle that seem to be stuck at a given location over given iterations experienced the microbial genetic algorithm. Two new hybridization techniques were introduced in chapter four of this thesis. However, only one of the two was experimented. The particles move in the adaptive AMGAPSO algorithm more than in the MGAPSO as the particle use their velocity to keep searching and they only die after they can no longer change, which is contrary to the technique in the MGAPSO where particles die after one move. Another possible explanation is that parameters of the particle modify their parameters based on their position which may have resulted in better parameters resulting and a better search by the particles.

It can thus be suggested based on this finding that the model of hybridisation and the ability of the particles to learn from the parameters of their particle best parameter, global best particle parameter and the ability of the particles to learn from itself when it obtains the global best position might be responsible for the improved performance of the new AMGAPSO

Although the new adaptive hybrid AMGAPSO had outperformed PSO and MGAPSO using simulated data, learners' profile data collected from Niger Delta University in Nigeria was used in comparative experiment to ascertain the behaviour of these algorithms relative to one another with actual learners' (500 learners) data. Anderson-Darling test was conducted to identify the distribution of the learners' data, the results of the showed that the learners' profile data does not follow any distribution. The adaptive AMGAPSO outperformed the MGAPSO and the PSO with the learners' data from the university which confirmed that the adaptive AMGAPSO might outperform the MGAPSO and the PSO regardless of the distribution.

The initial aim of this thesis was to automatically form learning groups which are similar in composition thus groups formed by the new adaptive hybrid MGAPSO were analysed. The ANOVA (one way) test results showed no significant difference in the means of all groups for all six learners profile attributes in all one hundred groups formed by the algorithm. The understanding at this point is that the data distribution may not affect the performance of the new adaptive hybrid MGAPSO and the new adaptive hybrid algorithm (AMGAPSO) fulfil the aim of the research to form balanced collaborative learning groups which are diverse within the groups and the groups are similar. The equality of the means of all attributes in all hundred groups in the ANOVA one-way test showed that the groups are similar in composition.

This study also agreed with earlier observations which showed that the particles of PSO get stuck when the number of participant becomes large (Ullmann, 2015) however, subsequent experiments and code modification showed that this could be corrected by improving on their error handling methods to be able to handle cases when null or empty particles are passed to such methods. A notable lesson learnt at this point was that the when null objects are passed into methods this could result in the objects being stuck and control stopped which can be handled by proper better error handling (throw and Caught) methods. In addition, the model of interaction between the particle swarm and the microbial genetic algorithm could affect the performance of the new hybrid MGAPSO; this knowledge could be extended to other hybrid algorithms.

5.6 Benefits of the Research

Collaborative learning promotes social, psychological and academic benefits among learners and improves retention (Johnson and Johnson; 1986) in addition, collaborative learning promotes critical thinking, increases interest among learners (Gokhale, 1995; Totten et. al.; 1991) and creates room for discussion among peers thereby reducing anxiety among learners (Totten et. al., 1991). However, these enormous benefits of collaborative learning can only be enjoyed when learners work in balanced groups which are very hard for instructors to form when the number of learners is large like in online classes. This algorithm will be of benefit to educational institutions and the research opens some areas of interest to the research community.

5.6.1 Educational Institutions

With the advent of online (MOOCS) and distance learning classes the number of participant in a single class tends to be very large because participants could be attending the class from any part of the globe, in such online classes instructors know very little about participants thus, forming groups for balanced collaborative learning among such learners becomes an enormous task for instructors which have deprived such learners the benefit of collaboration which learners in conventional classrooms benefit from. The algorithm from this study will enable instructors of such large classes automatically form balanced collaborative learning groups. Similarly, instructors of offline classes with large number of participants as obtained in universities in Nigeria could use the algorithm for forming balanced learning groups with minimal human intervention.

Some organisations have offices all around the globe and some of these organisations do have teams composed of members from different parts of the world, however selecting members to form teams have never been easy as most times officers composing the teams might not know much of other staff at the other end. This algorithm will help in such team’s formation.

5.6.2 Computer Science Research Community

The new adaptive AMGAPSO outperformed the particle swarm algorithm. Thus the study opens a new direction of research in optimization. The research community could research into using this new algorithm for large scale optimisation problems in business and engineering and all other areas of research in which the particle swarm optimisation algorithm has been applied

5.7 Additional / Alternative Methods

Alternative techniques could be used in the development of the new adaptive algorithm however these alternatives could be done at varying points or stages of the algorithm. To understand the alternatives, the alternatives are analysed as follows

5.7.1 Algorithm

In the algorithm design, other genetic algorithms could be used in place of the microbial genetic algorithm however, this is open to research as it is not clear how hybrid algorithm formed with PSO or other swarm algorithm (Bee, Fish school, ant colony) with genetic algorithm will behave in the collaborative learning group formation problem.

The mate selection mechanism could be modified where a particle mate is always better than the particle, this could be experimented to ascertain the outcome of the algorithm. The evolutionary property of the AMGAPSO could be improved by selecting the best particle among the parent, and two children produced during crossover and mutation to replace the parent in the next generation thus a particle will only be replaced when a better offspring is produced.

Although two adaptive hybridization methods were proposed in this research only one was used of these methods was used in the experiment thus the second proposed hybridization method is open to experimentation.

5.7.2 Data Collection

Learners data collection was not done personally during this research; this was not a better way as data collection by some other people without the supervision of the researcher could contain some abnormalities. In addition, the data collected for learners’ interest may not reflect the exact interest of the learners as participation in a task may not actually be a measure to the amount of interest in a topic similarly in the data for interest of learners all learners who partook in a task were score the same. This can be reasonably argued as this does not indicate equality in the interest levels in that topic

5.8 Conclusion of Chapter

In this chapter the new adaptive hybrid AMGAPSO algorithm, Comparative experiments were then conducted with real learner's data collected from Niger Delta University in Nigeria. In the experiment 500 students' data was used and the adaptive hybrid AMGAPSO outperformed all other algorithms in this experiment. With five hundred students' data one hundred unequal groups ranging from 4 to 6 were formed by the adaptive AMGAPSO algorithm, to ascertain the quality of the groups, the knowledge level and interest level for all three topics were analysed using analysis of variance. The statistical result showed that there was no significant difference in the means of knowledge levels for all groups similarly, there was no significant difference in the means of interest levels for all one hundred groups in all three topics consider, which means that the compositions of the groups were similar.

Chapter 6: Conclusion

6.1 Introduction

Although the findings should be interpreted with caution, this study has several strengths one of which is that it provides a simple and adaptive hybridisation method for the hybridisation of particle swarm and the basic microbial genetic algorithm which may be extended to the hybridisation of particle swarm and other genetic algorithms. The study also minimised the human intervention as no parameter setting of any kind was needed in the adaptive AMGAPSO experiments conducted. To conclude this thesis summary of the research is presented in section 6.2 while in section 6.3 and 6.4 contains contributions of the study and importance of the study respectively the limitation and future work are presented in section 6.5 and section 6.6 respectively.

6.2 Research summary

Although the problem of forming learning groups with large number of participants has been identified as key issue (objective one) within the research community due to the importance of collaborative learning groups in the teaching and learning process, very little work has been done in the forming of groups with large number of participants using intelligent algorithms. Lin et. al., (2010) and Ullmann (2015) were identified as notable research in the automatic learning group formation in which swarm intelligent algorithm were used, both researches were conducted with the particle swarm however, these research groups used limited sample size of learners and actual learners' data were not used in the evaluation of their research. Ullmann et. al., (2015) further suggested the investigation of the automatic learning groups formation with large number of participant because the particle swarm experienced stagnation after some iterations.

This indicates a need for development of an algorithm for automatic composition of learning groups with large number of participant.

With particle swarm been identified as the algorithm used in the existing literature for the group formation problem (objective two) and the reported problems (Ullmann et. al., 2015) there is need for a new algorithm. In chapter three of this thesis a new hybrid algorithm was developed from the hybridisation of the particle swarm and the basic microbial genetic algorithm (MGAPSO). A comparative experiment was conducted in which the new static MGAPSO outperformed the particle swarm algorithm with very high significant difference in the means of the two algorithms. However, an algorithm with minimal human intervention was still required. Thus an adaptive AMGAPSO was developed in chapter four of the thesis. The adaptive hybrid AMGAPSO outperformed the particle swarm and the MGAPSO in the comparative experiment with a high significant difference in the means of the adaptive AMGAPSO and MGAPSO. Thus objective 3 and objective 4 of the thesis were achieved.

In chapter five the new adaptive AMGAPSO was compared to the PSO and MGAPSO using real learners' data in a comparative experiment. The results confirmed the results obtained in the previous chapter in which the adaptive hybrid AMGAPSO outperformed the PSO and the MGAPSO. Analysis of the groups formed with the new adaptive algorithm using ANOVA (one-way test) showed there was no significant difference in the mean of all corresponding learners profile attributes in all groups formed which indicates that the groups formed using the algorithm were all well balanced groups thus the primary aim of the thesis have been achieved.

6.3 Importance of the Study

This study has raised important questions about the nature of hybrid algorithms. One such question is the effect of the model of hybridisation on the performance of the resulting hybrid algorithm. Secondly, the possibility of extending the use of the proposed hybrid algorithm for other optimisation problems.

The grouping of learners into learning groups in large classes can be achieved with easy, distant learning instructors can now group their learners with very little knowledge about the learners into balance collaborative learning groups.

Balanced collaborative groups and teams outside educational environment can be formed using this algorithm based on their interest and knowledge levels. The algorithm can be used on MOOCS for the formation of learning groups enabling users of MOOCS enjoy the benefits of collaboration and working in learning groups.

6.4 Limitations of the Study

The scope of this study was limited by the absence of very large number of real learners' profile data thus only 500 learners profile data was used in the validation of the algorithm for grouping. Thus, scalability of the algorithm was not investigated. The proposed algorithm was compared against PSO and not other state of the art algorithms this was because these algorithms were commercial and their details were not specified. Similarly, the proposed was not compared with other adaptive algorithms reviewed in the literature.

Understanding level was measured as average performance on test for each topic while Interest was measured by the number of class work hand-ins out of ten class work given, each hand-in of assignment was scored as one while ten such assignments were given. However, students were told that the assignments will not be used for assessments purpose.

One issue with this study was the challenges of obtaining the data the learners profile data directly by myself. It was very difficult to obtain the interest levels of the learners directly, the outcome was to rely on the data collected by a lecturer in the university thus the quality of the data cannot be guaranteed.

Another limitation was to obtain the profile data of large number of students who belong to a class, neither was it easy to obtain the profile data of online students thus data of students in the same class in Nigeria (mention this in chapter 5) was used in the experiment, this number was limited to five hundred (500).

The study did not consider equality of group sizes Secondly, the effect of the mate selection on the algorithm behaviour was also not investigated.

An additional limitation is that this research is evaluated only in terms of the group formation but the effectiveness of the group selection has not been evaluated against the overall learning improvement achieved by the groups formed by the algorithm.

6.4.2 Analysis

The analysis in this thesis was limited to the use of ANOVA (one way) with no other statistical tool been used in the analysis of the groups formed and in the analysis of the means of the groups.

The new adaptive AMGAPSO was compared to only the particle swarm optimisation algorithm, other swarm intelligent algorithms were not used in the comparison neither were other genetic algorithms used to compare with the new adaptive hybrid AMGAPSO.

6.5 Future Work

This research has raised many questions that need to be further investigated which are that

- Research is needed to ensure the new adaptive hybrid AMGAPSO algorithm produce groups of equal sizes.
- Further experimental investigation needs to be conducted to examine the effect of mate selection on the performance of the adaptive hybrid AMGAPSO and the MGAPSO.
- A broader research is also recommended to examine the performance of other hybrid algorithms which could likely be form by the hybridisation of swarm intelligent algorithms and genetic algorithms
- Further studies need to be done to investigate the possibility of adapting the new adaptive AMGAPSO in optimisation problems and compare the performance of the new algorithm with established benchmark optimisation algorithms
- Recommend for further investigation of the effect of the hybridisation model on the performance of hybrid algorithms particularly the hybrid algorithms formed by the hybridisation of PSO and genetic algorithms.
- A further investigation into the hybridization of Microbial Genetic Algorithm with other swarm intelligent algorithms (Bee, Fish school, ant colony).

It will also be interesting to compare the new adaptive hybrid algorithm AMGAPSO and PSO in other optimization problems.

References

Abnar, S., Orooji, F. and Taghiyareh, F., (2012), "An evolutionary algorithm for forming mixed groups of learners in web based collaborative learning environments". In Technology Enhanced Education (ICTEE), 2012 IEEE International Conference on (pp. 1-6).

Aghababa, M.P., Shotorbani, A.M. and Shotorbani, R.M., (2010). "An adaptive particle swarm optimization applied to optimum controller design for AVR power systems". International Journal of Computer Applications, 11(10), pp.22-29.

Agrawal, R., Golshan, B., Terzi, E. (2014) "Grouping students in educational settings". In: SIGKDD, pp. 1017–1026. ACM (2014)

Ahmed, Z.H., (2010) "Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator". International Journal of Biometrics & Bioinformatics (IJBB), 3(6), p.96.

Alberola, J.M., Del Val, E., Sanchez-Anguix, V., Palomares, A. and Teruel, M.D., (2016) "An artificial intelligence tool for heterogeneous team formation in the classroom". Knowledge-Based Systems, 101, pp.1-14.

Angert, E.R., (2005) "Alternatives to binary fission in bacteria" Nature Reviews Microbiology, 3(3), pp.214-224.

Ani, Z.C., Yasin, A., Husin, M.Z. and Hamid, Z.A., (2010) "A method for group formation using genetic algorithm". International Journal on Computer Science and Engineering, 2(09), pp.3060-3064.

Ardizzon, G., Cavazzini, G. and Pavesi, G., (2015) "Adaptive acceleration coefficients for a new search diversification strategy in particle swarm optimization algorithms". Information Sciences, 299, pp.337-378.

Arumugam, M.S. & Rao, M.V.C. 2008, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems", Applied Soft Computing Journal, vol. 8, no. 1, pp. 324-336.

Barr, R. S., Golden, B. L., Kelly, J. P., Resende, M. G. C., and Stewart, W. R. Designing and Reporting on Computational Experiments with Heuristic Methods. Journal of Heuristics 1 (1995), 9–32

Beauchamp, G.K. and Yamazaki, K., (1997). “HLA and Mate Selection in Humans: Commentary”. American journal of human genetics, 61(3), p.494.

Beheshti, Z., Shamsuddin, S.M. and Hasan, S., (2015). Memetic binary particle swarm optimization for discrete optimization problems. Information Sciences, 299, pp.58-84.

Carlisle, A. and Dozier, G., (2000), “Adapting particle swarm optimization to dynamic environments”. In International conference on artificial intelligence (Vol. 1, pp. 429-434).

Cavanaugh, C., Gillan, K.J., Kromrey, J., Hess, M. and Blomeyer, R., (2004). “The effects of distance education on K-12 student outcomes: A meta-analysis”. Learning Point Associates/North Central Regional Educational Laboratory (NCREL).

Cavanaugh, R., Ellis, M., Layton, R. and Ardis,. M. (2004) “Automating the process of assigning students to cooperative- learning teams”. In Proceedings of the 2004 American Society for Engineering Education Annual Conference & Exposition, June 2004.

Chatterjee, A. and Siarry, P., (2006). “Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization”. Computers & operations research, 33(3), pp.859-871.

Christodoulopoulos, C.E. and Papanikolaou, K.A., (2007), October. A group formation tool in an e-learning context. In Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on (Vol. 2, pp. 117-123). IEEE.

Cocea, M. and Magoulas, G.D., (2010), September. Group formation for collaboration in exploratory learning using group technology techniques. In International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (pp. 103-113). Springer, Berlin, Heidelberg.

Dascalu, M., Bodea, C., Lytras, M., de Pablos, P. & Burlacu, A. (2014), "Improving e-learning communities through optimal composition of multidisciplinary learning groups", COMPUTERS IN HUMAN BEHAVIOR, vol. 30, pp. 362-371.

DeCarvalho, R.J., (1991). The humanistic paradigm in education. The Humanistic Psychologist, 19(1), p.88.

Dong, X., Li, Q., Zhao, Q. and Ren, Z., (2016), June." Time-varying group formation control for multi-agent systems with second-order dynamics and directed topologies". In Intelligent Control and Automation (WCICA), 2016 12th World Congress on (pp. 350-355). IEEE.

Dong, X., Li, Q., Zhao, Q. and Ren, Z., (2016). "Time-varying group formation analysis and design for general linear multi-agent systems with directed topologies". International Journal of Robust and Nonlinear Control.

Eberhart R C, Kennedy J. (1995) "A new optimizer using particle swarm theory"; Proceedings of the sixth international symposium on micro machine and human science. 1995, 1: 39-43.

Eberhart, R.C. and Shi, Y.,(2000). "Comparing inertia weights and constriction factors in particle swarm optimization". In Evolutionary Computation, 2000. Proceedings of the 2000 Congress on (Vol. 1, pp. 84-88). IEEE.

Feng, Y., Teng, G., Wang, A. & Yao, Y. (2007); "Chaotic Inertia Weight in Particle Swarm Optimization", IEEE, pp. 475

Fukuyama, Y. and Yoshida, H., (2001). “A particle swarm optimization for reactive power and voltage control in electric power systems”. In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on (Vol. 1, pp. 87-93). IEEE.

Gaudioso, E. and Boticario, J.G., 2003, July. Towards web-based adaptive learning communities. In Proceedings of the 11th International Conference on Artificial Intelligence in Education (AIED’2003). Sidney, Australia.

Ghosh, S., Das, S., Kundu, D., Suresh, K., Panigrahi, B.K. and Cui, Z., 2012. An inertia-adaptive particle swarm system with particle mobility factor for improved global optimization. Neural Computing and Applications, 21(2), pp.237-250.

Gokhale, A.A. (1995). “Collaborative learning enhances critical thinking”. Journal of Technology education. 7(1), Retrieved 9th Jun. 2017, from: http://scholar.lib.vt.edu/ejournals/JTE/v7n1/gokhale.jte-v7n1.html

Goldberg D E, Lingle R. Alleles, (1985) “Loci, and the traveling salesman problem” Proceedings of the first international conference on genetic algorithms and their applications. Lawrence Erlbaum Associates, Publishers, 1985: 154-159.

Graf S, Bekele R. (2006) “Forming heterogeneous groups for intelligent collaborative learning systems with ant colony optimization Intelligent Tutoring Systems”. Springer Berlin Heidelberg, 2006: 217-226.

Ho, T., Shyu, S.J., Wang, F. & Li, C.T. (2009), "Composing High-Heterogeneous and High-Interaction Groups in Collaborative Learning with Particle Swarm Optimization", IEEE, pp. 607.

Holland, J.H., 2000. Building blocks, cohort genetic algorithms and hyperplane defined

Hsu, C.C., Chen, H.C., Huang, K.H., Huang, K.K. and Huang, Y.M., (2014). “The development of an adaptive group composition system on Facebook for collaborative learning using an artificial bee colony algorithm”. Appl. Math, 8(1L), pp.157-164.

Hübscher, R. (2010), "Assigning Students to Groups Using General and Context-Specific Criteria", IEEE Transactions on Learning Technologies, vol. 3, no. 3, pp. 178-189.

Isotani, S., Inaba, A., Ikeda, M. & Mizoguchi, R. (2009). An Ontology Engineering Approach to the Realization of Theory- Driven Group Formation. *International Journal of Computer Supported Collaborative Learning*, 4(4).

J. Bais, J. Giletto, J. Balmaceda, S. Schia_no, J. A. D__az (2013), “An Assistant for Group Formation in CSCL Based on Constraint Satisfaction”, Proceedings of ASAI (2013) 12{23.

Jagadish, D., (2014). Grouping in collaborative e-learning environment based on interaction among students. In Recent Trends in Information Technology (ICRTIT), 2014 International Conference on (pp. 1-5). IEEE.

Jiao, B., Lian, Z. & Gu, X. (2008), "A dynamic inertia weight particle swarm optimization algorithm", Chaos, Solitons and Fractals, vol. 37, no. 3, pp. 698-705.

Johnson, D. W., Johnson, R. T., and Smith, K. A. (1998). Active learning: Cooperation in the college classroom. Edina, MN: Interaction Book Company.

Johnson, R. T., and Johnson, D. W. (1994). An overview of cooperative learning. In Thousand, J., Villa, A. & Nevin, A. (Eds.), Creativity and collaborative learning (p.2). Baltimore, Maryland; USA. Brookes Publishing

Johnson, D.W. (1971). Effectiveness of role reversal: actor or listener. Journal of Psychological Reports. 28(1), 275-282.

Johnson, D.W., & Johnson, R.T. (1989). Cooperation and Competition Theory and Research. Edina, Minnesota, USA: Interaction Book Co.

Johnson, D.W., & Johnson, R.T. (2009). An Educational Psychology Success Story: Social Interdependence Theory and Cooperative Learning. *Journal of Educational researcher,* 38(5), 365-379

Johnson, R.T., & Johnson, D.W. (1986). Action Research: Cooperative Learning in the Science Classroom. Journal of Science and Children,

Kargupta, H. (1996), "The Gene Expression Messy Genetic Algorithm", , proceedings of the IEEE International Conference on Evolutionary Computation pp. 814.

Khandaker, N., Soh, L.K., Miller, L.D., Eck, A. and Jiang, H., (2011). Lessons learned from comprehensive deployments of multiagent CSCL applications I-MINDS and ClassroomWiki. IEEE Transactions on Learning Technologies, 4(1), pp.47-58.

Konert, J., Bellhäuser, H., Röpke, R., Gallwas, E. and Zucik, A., (2016), "MoodlePeers: Factors Relevant in Learning Group Formation for Improved Learning Outcomes, Satisfaction and Commitment in E-Learning Scenarios Using GroupAL". In European Conference on Technology Enhanced Learning (pp. 390-396). Springer International Publishing.

Laal, M. & Laal, M. (2012), "Collaborative learning: What is it?" Journal of Procedia-Social and Behavioural Sciences, 31, pp. 491.

Laal, M. and Ghodsi, S.M., (2012). Benefits of collaborative learning. Procedia-Social and Behavioral Sciences, 31, pp.486-490.

Laal, M., Laal, M., & Khattami-Kermanshahi, Zh. (2012). 21st century learning; learning in collaboration. *Journal of Procedia - Social and Behavioral Sciences*, 47, 1696 – 1701

Laal, M., Naseri, A.S., Laal, M. and Khattami-Kermanshahi, Z., (2013). What do we achieve from learning in Collaboration? Procedia-Social and Behavioral Sciences, 93, pp.1427-1432.

Liang, J.J., Qin, A.K., Suganthan, P.N. and Baskar, S., 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE transactions on evolutionary computation, 10(3), pp.281-295.

Liang, X., Li, W., Zhang, Y. and Zhou, M., (2015). An adaptive particle swarm optimization method based on clustering. Soft Computing, 19(2), pp.431-448.

Lin Y T, Huang Y M, Cheng S C. (2010) “An automatic group composition system for composing collaborative learning groups using enhanced particle swarm optimization” [J]. Computers & Education, 2010, 55(4): 1483-1493.

Liu, C.Y., Chen, C.M., Kuo, C.H. and Taipei City, T.R., (2016), “An Optimized Group Formation Scheme Considering Knowledge Level, Learning Roles, and Interaction Relationship for Promoting Collaborative Problem-Based Learning Performance”. In ICEL2016-Proceedings of the 11th International Conference on e-Learning: ICEl2016 (p. 238). Academic Conferences and publishing limited.

Maia, C. (2008). “Is there a future for distance learning in Brazil?” Proceedings of the 6th International Conference on Networked Learning, Halkidiki, Greece, pp.843-844.

Moreno J, Ovalle D A, Vicari R M. (2012). “A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics” [J]. Computers & Education, 58(1): 560-569.

Mujkanovic, A., Lowe, D.B. and Willey, K., (2012). “Adaptive group formation to promote desired behaviours”. In 23rd Annual Conference for the Australasian Association for Engineering Education-The Profession of Engineering Education: Advancing Teaching, Research and Careers. Swinburne University of Technology.

Nickabadi, A., Ebadzadeh, M.M. and Safabakhsh, R., (2011). “A Novel Particle Swarm Optimization Algorithm with Adaptive Inertia Weight”. Applied Soft Computing, 11(4), pp.3658-3670.

Ounnas, A., Davis, H.C. and Millard, D.E., (2007). “Semantic modeling for group formation”.

Ounnas, A., Davis, H.C. and Millard, D.E., (2009). “A Framework for Semantic Group Formation in Education”. Journal of Educational Technology & Society, 12(4).

Panigrahi, B.K., Pandi, V.R. and Das, S., (2008). “Adaptive particle swarm optimization approach for static and dynamic economic load dispatch”. Energy conversion and management, 49(6), pp.1407-1415.

Panitz, T. (1996). “A Definition of Collaborative Vs Cooperative Learning. Deliberations”, London Metropolitan University; UK., Retrieved 2016 June 1, from: http://www.londonmet.ac.uk/deliberations/collaborative-learning/panitz-paper.cfm.

Paredes, P., Ortigosa, A. and Rodriguez, P., (2010). A Method for Supporting Heterogeneous-Group Formation through Heuristics and Visualization. J. UCS, 16(19), pp.2882-2901.

Pinninghoff, M.A., Ramírez, M., Arriagada, R.C. and Lagos, P.S., (2015), June. Collaborative Group Formation Using Genetic Algorithms. In International Work-Conference on the Interplay Between Natural and Artificial Computation (pp. 330-338). Springer International Publishing.

Pornsing, C., Saichareon, K. and Kerot, T., (2015). “A modified particle swarm optimization for engineering constrained optimization problems”. In Proc. International Conference on Innovative Computational Technologies and Industrial (pp. 1-5).

Potts, J.C., Giddens, T.D. and Yadav, S.B., (1994). The development and evaluation of an improved genetic algorithm based on migration and artificial selection. IEEE Transactions on Systems, Man, and Cybernetics, 24(1), pp.73-86.

Premalatha, K. and Natarajan, A.M., (2009). Hybrid PSO and GA for global maximization. Int. J. Open Problems Compt. Math, 2(4), pp.597-608.

R. Eberhart, Y. Shi. (2001). Particle swarm optimization: developments, applications and resources. IEEE Int. Conf. on Evolutionary Computation, 2001: 81-86

Rezazadeh, I., Meybodi, M.R. and Naebi, A., (2011), September. Adaptive Particle Swarm Optimization Algorithm in Dynamic Environments. In Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on (pp. 74-79). IEEE.

Rogers and Freiberg, 1994). Rogers, C., & Freiberg, H. J. (1994). Freedom to learn (3rd Ed.). New York: Macmillan.

Rubens, N., Vilenius, M. and Okamoto, T., (2009), September. Automatic group formation for informal collaborative learning. In Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on (Vol. 3, pp. 231-234). IEEE.

Sadeghi H, Kardan A A. (2015) A novel justice-based linear model for optimal learner group formation in computer-supported collaborative learning environments [J]. Computers in Human Behavior, 2015, 48: 436-447.

Sadeghi H, Kardan A A. (2015) Toward effective group formation in computer-supported collaborative learning [J]. Interactive Learning Environments, 2014: 1-14.

Sha, D. Y., & Hsu, C. Y. (2006). A hybrid particle swarm optimization for job shop scheduling problem. Computers & Industrial Engineering, 51(4), 791–808.

Shayeghi, H., Jalili, A. and Shayanfar, H.A., (2008). Multi-stage fuzzy load frequency control using PSO. Energy Conversion and Management, 49(10), pp.2570-2580.

Sheng, L. and Gu, X., (2014). A genetic algorithm with combined operators for permutation flowshop scheduling problems. In Information and Automation (ICIA), 2014 IEEE International Conference on (pp. 65-70). IEEE.

Shi Y, Eberhart RC.(1998)" Parameter selection in particle swarm optimization". In: Evolutionary programming VII: proceedings of the seventh annual conference on evolutionary programming, New York, 1998, p. 591–600.

Shi, Y., (2001). "Particle swarm optimization: developments, applications and resources". In evolutionary computation, 2001. Proceedings of the 2001 Congress on (Vol. 1, pp. 81-86). IEEE.

Shi, Y., Eberhart. R. (2001) "Fuzzy adaptive particle swarm optimization". IEEE Int. Conf. on Evolutionary Computation, 2001: 101-106

Srba, I. and Bielikova, M., 2015. Dynamic group formation as an approach to collaborative learning support. IEEE Transactions on Learning Technologies, 8(2), pp.173-186.

Srinivas, H. (2011 Oct. 21, last updated). What is Collaborative Learning? The Global Development Research Centre, Kobe; Japan . Retrieved

Srinivas, M. and Patnaik, L.M., 1994. Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Transactions on Systems, Man, and

Suresh, K., Ghosh, S., Kundu, D., Sen, A., Das, S. & Abraham, A. 2008, "Inertia-Adaptive Particle Swarm Optimizer for Improved Global Search", IEEE, pp. 253

Tacadao, G.S. and SJ, R.P.S.T., 2015, July. A Generic Model for the Group Formation Problem Using Constraint Logic Programming. In Advanced Learning Technologies (ICALT), 2015 IEEE 15th International Conference on (pp. 307-308). IEEE.

Ullmann, M.R.D., Ferreira, D.J., Camilo-Junior, C.G., Caetano, S.S. & de Assis, L. 2015, "Formation of learning groups in cMoocs using Particle Swarm Optimization", IEEE, pp. 3296

Vygotsky, L. (1978). Mind in society: The development of higher psychological processes. Cambridge: Harvard University Press.

Wang, D.; Lin, S.; Sun, C. 2010. DIANA: A computer-supported heterogeneous grouping system for teachers to conduct successful small learning groups. Computers in Human Behavior vol. 23, pp. 1997–2010.

Wang, G., Lu, M., Dong, Y. & Zhao, X. 2016, "Self-adaptive extreme learning machine", Neural Computing and Applications, vol. 27, no. 2, pp. 291-303.

Wang, X., Guo, Y. & Liu, G. 2013, "Self-adaptive particle swarm optimization algorithm with mutation operation based on K-Means", pp. 2194.

Whitley, D., 1994. A genetic algorithm tutorial. Statistics and computing, 4(2), pp.65-85.

Wichmann, A., Hecking, T., Elson, M., Christmann, N., Herrmann, T. and Hoppe, H.U., (2016). "Group Formation for Small-Group Learning: Are Heterogeneous Groups More Productive?". In Proceedings of the 12th International Symposium on Open Collaboration (p. 14). ACM.

Yang, X., Yuan, J., Yuan, J. and Mao, H., (2007). "A modified particle swarm optimizer with dynamic adaptation". Applied Mathematics and Computation, 189(2), pp.1205-1213.

Yannibelli, V. and Amandi, A., (2012). A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. Expert Systems with Applications, 39(10), pp.8584-8592.

Zhang, Y., Wang, S. and Ji, G., (2015). "A comprehensive survey on particle swarm optimization algorithm and its applications". Mathematical Problems in Engineering, 2015.

Zheng, Y., Liu, Y., Lu, W. and Li, C., (2016). "A hybrid PSO-GA method for composing heterogeneous groups in collaborative learning". In Computer Science & Education (ICCSE), 2016 11th International Conference on (pp. 160-164). IEEE.

Zheng, Z. and Pinkwart, N., (2014), "A discrete particle swarm optimization approach to compose heterogeneous learning groups". In Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on (pp. 49-51). IEEE.

Zhou, M., Ye, Y., Ren, Y. and Xu, Y., (2016). "Collaborative Learning Group Formation with Density Clustering". International Journal of Grid and Distributed Computing, 9(6), pp.117-126.

Appendix A Approval Letter

NIGER DELTA UNIVERSITY

WILBERFORCE ISLAND, BAYELSA STATE

DEPARTMENT OF MATHEMATICS / COMPUTER SCIENCE

FACULTY OF SCIENCE

Dr. M. A Orukari, B. Sc (RSUST), M. Sc (IBADAN), Ph. D (RSUST)

E-mail: orukarimer@gmail.com

GSM: +234 (0) 8037079340, +234(0) 8056941605

Jan. 19, 2017

The Ethics Committee

School of Engineering,

Design and Computer Science

Brunel University London

UB8 3PH, UK

Dear Sir**,**

APPROVAL FOR THE USE OF STUDENTS DATA

The Department of Computer Science, Niger Delta University, Bayelsa State, Nigeria has approved the use of students' data for a purpose of research by Mr. Nicholas Dienagha. However, the data would be made anonymous and must be used strictly for the research only.

By this approval, Mr. Nicholas Dienagha would be granted access to the data of Five Hundred (500) students and any other material that will be needed for his research. He is to meet with the database administrator of the department to obtain the required data.

Yours sincerely,

Dr.(Mrs). Mercy ORUKARI

Ag. HOD

Appendix B

Results of the Anderson-Darling Test

#	Distribution	Parameters
1	Beta	α_1=3314.4 α_2=1913.6 a=-85.962 b=57.399
2	Burr	k=1.6578 α=7.8387 β=5.306
3	Burr (4P)	k=1.3149 α=11.13 β=6.6921 γ=-1.5969
4	Cauchy	σ=0.38921 μ=4.9655
5	Chi-Squared	ν=4
6	Chi-Squared (2P)	ν=3 γ=1.9701
7	Dagum	k=0.69087 α=10.53 β=5.1403
8	Dagum (4P)	k=1.5189 α=1.8201E+7 β=1.0936E+7 γ=-1.0936E+7
9	Erlang	m=26 β=0.18388
10	Erlang (3P)	m=103 β=0.09352 γ=-4.7009
11	Error	k=1.6469 σ=0.95082 μ=4.9167
12	Error Function	h=0.74368
13	Exponential	λ=0.20339
14	Exponential (2P)	λ=0.34286 γ=2.0
15	Fatigue Life	α=0.20241 β=4.8179
16	Fatigue Life (3P)	α=0.06011 β=15.759 γ=-10.87
17	Frechet	α=5.662 β=4.3545
18	Frechet (3P)	α=3.0918 β=2.6619 γ=1.6814
19	Gamma	α=26.739 β=0.18388
20	Gamma (3P)	α=107.22 β=0.09178 γ=-4.9248
21	Gen. Extreme Value	k=-0.18429 σ=0.83832 μ=4.5641
22	Gen. Gamma	k=0.99458 α=26.263 β=0.18388

23	Gen. Gamma (4P)	k=1.5951 α=30.642 β=0.98377 γ=-3.4581
24	Gen. Pareto	k=-0.78495 σ=2.5015 μ=3.5152
25	Gumbel Max	σ=0.74135 μ=4.4887
26	Gumbel Min	σ=0.74135 μ=5.3446
27	Hypersecant	σ=0.95082 μ=4.9167
28	Inv. Gaussian	λ=131.47 μ=4.9167
29	Inv. Gaussian (3P)	λ=4369.7 μ=15.793 γ=-10.876
30	Johnson SU	γ=-0.86595 δ=3.417 λ=3.0096 ξ=4.112
31	Kumaraswamy	α_1=3.7433 α_2=647.65 a=1.5908 b=22.325
32	Laplace	λ=1.4874 μ=4.9167
33	Levy	σ=4.7213
34	Levy (2P)	σ=2.4619 γ=1.8617
35	Log-Gamma	α=61.33 β=0.02565
36	Log-Logistic	α=8.4307 β=4.8193
37	Log-Logistic (3P)	α=21.348 β=11.333 γ=-6.4571
38	Log-Pearson 3	α=10.598 β=-0.06171 γ=2.2272
39	Logistic	σ=0.52421 μ=4.9167
40	Lognormal	σ=0.20077 μ=1.5732
41	Lognormal (3P)	σ=0.06152 μ=2.7338 γ=-10.503
42	Nakagami	m=6.7312 Ω=25.077
43	Normal	σ=0.95082 μ=4.9167
44	Pareto	α=1.1363 β=2
45	Pareto 2	α=229.59 β=1197.0
46	Pearson 5	α=23.839 β=112.55
47	Pearson 5 (3P)	α=248.27 β=3722.0 γ=-10.135
48	Pearson 6	α_1=27.183 α_2=536.98 β=96.968

49	Pearson 6 (4P)	α_1=156.67 α_2=2114.2 β=154.63 γ=-6.5473
50	Pert	m=4.8879 a=1.8786 b=8.1353
51	Power Function	α=1.2505 a=1.9853 b=8.0
52	Rayleigh	σ=3.9229
53	Rayleigh (2P)	σ=2.1878 γ=1.9722
54	Reciprocal	a=2.0 b=8.0
55	Rice	ν=4.82 σ=0.9603
56	Triangular	m=5.0 a=1.9328 b=8.0523
57	Uniform	a=3.2698 b=6.5635
58	Weibull	α=5.9325 β=5.3091
59	Weibull (3P)	α=3.7497 β=3.6745 γ=1.5887
60	Johnson SB	No fit
61	Student's t	No fit

APPENDIX C

Tables of Critical Values for the Anderson-Darling Test.

		0.01	0.025	0.05	0.10	0.15				
Specified		3.853	3.070	2.492	1.933	1.610				
		0.0025	0.005	0.01	0.025	0.05	0.10	0.15	0.20	0.25
Expon		2.534	2.244	1.959	1.591	1.321	1.162	0.916	0.816	0.736
		0.01	0.025	0.05	0.10	0.25				
Weibull		1.038	0.877	0.757	0.637	0.474				
	k\α	0.005	0.01	0.025	0.05	0.10	0.25			
Gamma	1	1.227	1.092	0.917	0.786	0.657	0.486			
	2	1.190	1.062	0.894	0.768	0.643	0.477			
	3	1.178	1.052	0.886	0.762	0.639	0.475			
	4	1.173	1.048	0.883	0.759	0.637	0.473			
	5	1.170	1.045	0.881	0.758	0.635	0.472			
	6	1.168	1.043	0.880	0.757	0.635	0.472			
	8	1.165	1.041	0.878	0.755	0.634	0.471			
	10	1.164	1.040	0.877	0.754	0.633	0.471			
	12	1.162	1.038	0.876	0.754	0.633	0.471			
	15	1.162	1.038	0.876	0.754	0.632	0.470			
	20	1.161	1.037	0.875	0.753	0.632	0.470			
	>20	1.159	1.035	0.873	0.752	0.631	0.470			

	α	0.005	0.01	0.025	0.05	0.10	0.20			
Normal	a	1.1578	1.0348	0.8728	0.7514	0.6305	0.5091			
	b	1.063	1.013	0.881	0.795	0.750	0.756			
	d	1.34	0.93	0.94	0.89	0.80	0.39			

http://www.real-statistics.com/statistics-tables/anderson-darling-test-table/

accessed 23/11/2017

Appendix D

Goodness of Fit - Summary

#	Distribution	Kolmogorov Smirnov		Anderson Darling		Chi-Squared	
		Statistic	Rank	Statistic	Rank	Statistic	Rank
1	Beta	0.2362	30	44.596	18	416.43	22
2	Burr	0.22519	12	45.171	23	420.17	30
3	Burr (4P)	0.22661	13	45.541	25	421.3	34
4	Cauchy	0.2385	32	55.525	39	326.19	7
5	Chi-Squared	0.55288	57	316.11	57	2412.3	54
6	Chi-Squared (2P)	0.39267	50	153.13	48	672.07	45
7	Dagum	0.22717	15	45.771	26	422.63	38
8	Dagum (4P)	0.23151	22	45.838	27	421.39	35
9	Erlang	0.28731	44	54.419	38	91.056	1
10	Erlang (3P)	0.22513	11	44.209	12	416.34	20
11	Error	0.22776	16	45.349	24	418.59	29
12	Error Function	0.99253	59	11989.0	59	54804.0	56
13	Exponential	0.51561	56	281.22	55	2366.2	52
14	Exponential (2P)	0.45516	51	205.17	50	863.26	46
15	Fatigue Life	0.24382	36	44.495	16	420.5	32
16	Fatigue Life (3P)	0.21986	4	43.967	10	416.29	16
17	Frechet	0.30417	46	64.585	42	115.15	2
18	Frechet (3P)	0.27417	42	86.9	44	N/A	
19	Gamma	0.23135	20	44.346	13	417.18	24
20	Gamma (3P)	0.21929	2	43.908	6	416.31	17
21	Gen. Extreme Value	0.23162	23	47.305	32	414.97	9
22	Gen. Gamma	0.23144	21	43.887	5	417.54	25
23	Gen. Gamma (4P)	0.22191	7	43.966	9	416.17	14
24	Gen. Pareto	0.22134	6	226.66	51	N/A	

25	Gumbel Max	0.27657	43	56.887	40	385.27	8
26	Gumbel Min	0.30019	45	74.411	43	547.8	44
27	Hypersecant	0.22298	10	49.607	35	424.67	39
28	Inv. Gaussian	0.23204	27	47.998	33	420.32	31
29	Inv. Gaussian (3P)	0.23179	25	44.558	17	416.4	21
30	Johnson SU	0.22209	8	44.615	19	418.26	28
31	Kumaraswamy	0.23774	31	43.92	7	415.1	12
32	Laplace	0.2294	18	52.841	37	427.2	42
33	Levy	0.58295	58	376.06	58	3912.1	55
34	Levy (2P)	0.50563	55	286.62	56	1772.9	51
35	Log-Gamma	0.25878	39	47.201	31	425.06	40
36	Log-Logistic	0.24807	37	44.679	20	430.18	43
37	Log-Logistic (3P)	0.22918	17	45.942	29	422.58	37
38	Log-Pearson 3	0.23448	28	44.936	21	415.04	10
39	Logistic	0.22701	14	47.099	30	421.88	36
40	Lognormal	0.24274	35	44.478	15	420.69	33
41	Lognormal (3P)	0.21955	3	43.968	11	416.32	18
42	Nakagami	0.21926	1	43.652	1	416.14	13
43	Normal	0.23175	24	44.465	14	416.33	19
44	Pareto	0.50396	54	273.46	54	1740.9	50
45	Pareto 2	0.49401	53	264.5	53	2396.1	53
46	Pearson 5	0.2538	38	45.907	28	426.18	41
47	Pearson 5 (3P)	0.22212	9	43.675	2	416.81	23
48	Pearson 6	0.23196	26	43.882	4	417.8	26
49	Pearson 6 (4P)	0.22008	5	43.965	8	416.27	15
50	Pert	0.24218	34	52.364	36	220.35	4
51	Power Function	0.34508	48	140.07	46	885.47	47
52	Rayleigh	0.36427	49	151.75	47	974.26	48
53	Rayleigh (2P)	0.30808	47	90.895	45	250.34	6
54	Reciprocal	0.45889	52	198.7	49	1380.4	49

55	Rice	0.2302	19	45.026	22	187.25	3
56	Triangular	0.26545	41	57.547	41	220.56	5
57	Uniform	0.24137	33	230.02	52	N/A	
58	Weibull	0.26298	40	48.977	34	418.03	27
59	Weibull (3P)	0.23586	29	43.868	3	415.09	11
60	Johnson SB	No fit					
61	Student's t	No fit					