CENTRIPETAL ACCELERATED PARTICLE SWARM OPTIMIZATION AND ITS APPLICATIONS IN MACHINE LEARNING

ZAHRA BEHESHTI

A thesis submitted in fulfilment of the requirements for the award of the degree of Doctor of Philosophy (Computer Science)

Faculty of Computing
Universiti Teknologi Malaysia

JANUARY 2013
This Thesis is dedicated to my beloved family for their endless support and encouragement.
ACKNOWLEDGMENTS

In the Name of Allah, Most Gracious, Most Merciful

First and foremost, I must thank Allah S.W.T. for His unlimited bounties and innumerable graces for helping me finishing this thesis to its best form. Here, I would like to express heartfelt gratitude to my supervisor Prof. Dr. Siti Mariyam Hj. Shamsuddin for her constant support during my study at UTM. She inspired me greatly to work in this project. Her willingness to motivate me contributed tremendously to our project. I have learned a lot from her and I am fortunate to have her as my mentor and supervisor. Also, I would like to thank the members of evaluation committee, Assoc. Prof. Dr. Siti Zaiton binti Mohd Hashim, Prof. Dr. Azuraliza binti Abu Bakar and Prof. Dr. Abdul Hanan bin Abdullah. I am thankful for their valuable comments and suggestions.

Next, I wish to extend my grateful appreciation to all those who have contributed directly and indirectly to the preparation of this study. I would like to express my deep gratitude and thanks to my father, my mother, my beloved family for their patience, support and prayers.

Besides, I would like to thank the authority of Universiti Teknologi Malaysia (UTM) for providing me with a good environment and facilities which I needed during the process.
ABSTRACT

Nowadays, meta-heuristic optimization algorithms have been extensively applied to a variety of Machine Learning (ML) applications such as classification, recognition, prediction, data mining and web mining, combinatorial optimization and so on. The majority of them imitate the behavior of natural phenomena to find the best solution. The algorithms find promising regions in an affordable time due to exploration and exploitation ability. Although the mentioned algorithms have satisfactory results in various fields, none of them is able to present a higher performance for all applications. Therefore, searching for a new meta-heuristic algorithm is an open problem. In this study, an improved scheme of Particle Swarm Optimization (PSO) based on Newtonian’s motion laws called Centripetal Accelerated Particle Swarm Optimization (CAPSO) has been proposed to accelerate learning process and to increase accuracy in solving ML problems. A binary mode of the proposed algorithm called Binary Centripetal Accelerated Particle Swarm Optimization (BCAPSO) has been developed for discrete (binary) search space. These algorithms have been employed for problems such as non-linear benchmark functions, Multi-Layer Perceptron (MLP) learning and the 0-1 Multidimensional Knapsack Problem (MKP). The results have been compared with several well-known meta-heuristic population-based algorithms in both continuous (real) and binary search spaces. From the experiments, it could be concluded that the proposed methods show significant results in function optimization for real and binary search spaces, MLP learning for classification problems and solving MKP for binary search space.
ABSTRAK

Kini pengoptimum algoritma meta-huristik sudah digunakan dengan meluasnya dalam pelbagai aplikasi mesin pembelajaran (ML) seperti pengklasifikasian, pengecaman, ramalan, pencarian data dan pencarian jaringan, pengoptimum kombinasi dan sebagainya. Kebanyakan aplikasi ini meniru keadaan fenomena semulajadi bagi mendapatkan penyelesaian terbaik. Algoritma akan mendapatkan ruang yang sangat sesuai dalam jangkamasa tertentu mengikut keupayaan eksplorasi dan eksploitasi. Walaupun algoritma tersebut memberi keputusan yang memuaskan dalam banyak bidang, namun tidak satu pun diantaranya dapat menghasilkan prestasi yang lebih tinggi untuk semua aplikasi. Maka, untuk mencari algoritma meta-huristik yang baru merupakan suatu cabaran yang nyata. Di dalam kajian ini, skim Particle Swarm Optimization (PSO) yang diperbaharui berdasarkan hukum gerakan Newtonian yang dipanggil Centripetal Accelerated Particle Swarm Optimization (CAPSO) telah dicadangkan bagi mempercepat proses pembelajaran dan meningkatkan ketepatan untuk menyelesaikan masalah-masalah ML. Mod binari algoritma yang dicadangkan yang dinamakan Binary Centripetal Accelerated Particle Swarm Optimization (BCAPSO) dibangunkan untuk pencarian ruang diskret (binari). Kesemua algoritma tersebut telah digunakan bagi mengatasi beberapa kesulitan seperti fungsi penanda aras bukan linear, pembelajaran Multi-Layer Perceptron (MLP) dan 0-1 Multidimensional Knapsack Problem (MKP). Keputusan telah dibandingkan dengan beberapa algoritma meta-huristik berdasarkan populasi yang terkenal carian ruang nyata dan binari. Daripada eksperimen, boleh disimpulkan bahawa kaedah yang dicadangkan menunjukkan hasil yang signifikan bagi fungsi pengoptimum untuk nyata dan pencarian ruang binari, pembelajaran MLP bagi masalah-masalah klasifikasi dan penyelesaian MKP untuk pencarian ruang binari.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECLARATION</td>
<td></td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td></td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td></td>
<td>iv</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td></td>
<td>v</td>
</tr>
<tr>
<td>ABSTRAK</td>
<td></td>
<td>vi</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td></td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td></td>
<td>xiii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td></td>
<td>xvi</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td></td>
<td>xix</td>
</tr>
<tr>
<td>LIST OF APPENDICES</td>
<td></td>
<td>xxiii</td>
</tr>
</tbody>
</table>

1  INTRODUCTION 1

1.1 Overview 1

1.2 Problem Background 3

1.3 Research Statement with Research Questions 6

1.4 Goal of the Research 7

1.5 Objectives of the Research 8

1.6 Scope of the Study 8

1.7 Importance of the Study 9

1.8 Organization of the Thesis 9

2  META-HEURISTIC ALGORITHMS 12

2.1 Introduction 12

2.2 Concept of meta-heuristic 12
2.3 Classification of meta-heuristic algorithms
   2.3.1 Nature-inspired against non-nature inspired
   2.3.2 Population-based against single point search
   2.3.3 Dynamic against static objective function
   2.3.4 Various neighborhood structures against single neighborhood
   2.3.5 Memory usage against memory-less methods

2.4 Related works

2.5 Population-based meta-heuristic algorithms
   2.5.1 Population-based meta-heuristic algorithms in real search space
      2.5.1.1 Genetic Algorithm (GA)
      2.5.1.2 Particle Swarm Optimization (PSO) in real search space
      2.5.1.3 Imperialist Competition Algorithm (ICA)
      2.5.1.4 Gravitational Search Algorithm (GSA) in real search space
   2.5.2 Population-based meta-heuristic algorithms in binary search space
      2.5.2.1 PSO in binary search space (BPSO)
      2.5.2.2 GSA in binary search space (BGSA)

2.6 Discussion

2.7 Summary

3 MACHINE LEARNING AND ITS APPLICATIONS
   3.1 Introduction
   3.2 Machine Learning (ML)
3.3 Meta-heuristic algorithms in Machine Learning (ML)

3.3.1 Function optimization

3.3.2 Artificial Neural Networks (ANNs)

3.3.2.1 Multi-Layer Perceptron (MLP) network

3.3.2.2 Back-Propagation (BP) algorithm for MLP training

3.3.2.3 Meta-heuristic algorithms for MLP training

3.3.3 Combinatorial Optimization Problems

3.3.3.1 The 0-1 Multidimensional Knapsack Problem (MKP)

3.3.3.2 Hybrid of meta-heuristic algorithms and the 0-1 MKP

3.4 Summary

4 RESEARCH METHODOLOGY

4.1 Introduction

4.2 General Research Framework

4.2.1 Phase 1: Research development

4.2.1.1 Data preparation for function optimization

4.2.1.2 Data preparation for MLP learning

4.2.1.3 Data preparation for the 0-1 MKP

4.2.1.4 Identifying the hybrid learning meta-heuristic algorithms for MLP network

4.2.1.5 Identifying the hybrid meta-heuristic algorithms for solving the 0-1 MKP
4.2.2  Phase 2: Design and development of algorithms 74
4.2.3  Phase 3: Validation process 75
4.2.3.1  Function optimization 75
4.2.3.2  Performance measure in classification problems using MLP network 76
4.2.3.3  Performance evaluation of MKP 77

4.3  Summary 78

5  CENTRIPETAL ACCELERATED PARTICLE SWARM OPTIMIZATION (CAPSO) FOR REAL AND BINARY SEARCH SPACES USED IN FUNCTION OPTIMIZATION 79

5.1  Introduction 79
5.2  The Newtonian’s motion laws used to design the proposed algorithms 79
5.3  CAPSO - The proposed algorithm for real search space 81
5.4  BCAPSO - The proposed algorithm for binary search space 83
5.5  Analysis and design of the proposed algorithms 84
5.6  Experimental results of the proposed methods for function optimizations 87
5.6.1  Analysis and discussion of function optimization in real search space 88
5.6.1.1  Comparison with different dimension 94
5.6.1.2  Comparison with other PSO algorithms 98
5.6.2  Analysis and discussion of function optimization in binary search space 101
5.6.3  Overall comparison of algorithms performance 109

5.7  Summary 111
# ENHANCEMENT OF MULTI-LAYER PERCEPTRON (MLP) LEARNING USING CENTRIPETAL ACCELERATED PARTICLE SWARM OPTIMIZATION (CAPSO)

6.1 Introduction 113

6.2 Hybrid learning of CAPSO and MLP network (CAPSO–MLP) 113

6.3 Results of CAPSO-MLP 115
   6.3.1 Experimental step 115
   6.3.2 Analysis and discussion of CAPSO-MLP 116

6.4 Summary 124

# BINARY CENTRIPETAL ACCELERATED PARTICLE SWARM OPTIMIZATION (BCAPSO) FOR SOLVING THE 0-1 MULTIDIMENSIONAL KNAPSACK PROBLEM (MKP)

7.1 Introduction 125

7.2 Hybrid of BCAPSO and the 0-1 MKP 125

7.3 Results and discussion of the 0-1MKP 129
   7.3.1 Analysis of Penalty Function (PF) technique 129
   7.3.2 Analysis of Check-and-Dropt (CD) algorithm 133
   7.3.3 Analysis of Improved Check-and-Repair Operator (ICRO) algorithm 136

7.4 Summary 143

# CONCLUSION AND FUTURE WORK

8.1 Introduction 144

8.2 Research summary 144

8.3 Research contributions 146

8.4 Future works 147
## LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>List of some meta-heuristic algorithms (1975-2012)</td>
<td>17</td>
</tr>
<tr>
<td>2.2</td>
<td>Advantages and disadvantages of some meta-heuristic algorithms</td>
<td>42</td>
</tr>
<tr>
<td>4.1</td>
<td>Unimodal test functions</td>
<td>66</td>
</tr>
<tr>
<td>4.2</td>
<td>Multimodal high-dimensional test functions</td>
<td>67</td>
</tr>
<tr>
<td>4.3</td>
<td>Multimodal low-dimensional test functions</td>
<td>68</td>
</tr>
<tr>
<td>4.4</td>
<td>Maximization function (Max-Ones) in binary search space</td>
<td>68</td>
</tr>
<tr>
<td>4.5</td>
<td>Fairly moderate MKP benchmarks</td>
<td>72</td>
</tr>
<tr>
<td>4.6</td>
<td>Hard MKP benchmarks</td>
<td>73</td>
</tr>
<tr>
<td>5.1</td>
<td>Minimization results of unimodal functions in Table 4.1 for real search space (Maximum iteration=1000 and n=30)</td>
<td>89</td>
</tr>
<tr>
<td>5.2</td>
<td>Minimization results of multimodal high-dimensional functions in Table 4.2 for real search space (Maximum iteration=1000 and n=30)</td>
<td>91</td>
</tr>
<tr>
<td>5.3</td>
<td>Minimization results of multimodal low-dimensional functions in Table 4.3 for real search space (Maximum iteration=500)</td>
<td>93</td>
</tr>
<tr>
<td>5.4</td>
<td>Minimization results of benchmark functions in Table 4.1 and Table 4.2 for real search space (Maximum iteration=2000 and n=100)</td>
<td>95</td>
</tr>
<tr>
<td>5.5</td>
<td>Minimization results of benchmark functions in Table 4.1 and Table 4.2 for real search space (Maximum iteration=3000 and n=200)</td>
<td>97</td>
</tr>
<tr>
<td>5.6</td>
<td>Seven PSO algorithms in the literature</td>
<td>99</td>
</tr>
</tbody>
</table>
Comparative results of seven PSO algorithms (Zhan et al., 2009) with CAPSO and LCAPSO on eleven benchmark functions of Table 4.1 and Table 4.2 (Maximum iteration=200000 and n=30)

Minimization results of unimodal functions in Table 4.1 for binary search space (Maximum iteration=500, n=5 and dim=75)

Minimization results of multimodal high-dimensional functions in Table 4.2 for binary search space (Maximum iteration=500, n=5 and dim=75)

Minimization results of multimodal low-dimensional functions in Table 4.3 for binary search space (Maximum iteration=500)

Maximization results of benchmark function in Table 4.4 for binary search space (Maximum iteration=1000)

Overall rank of Table 5.3 to Table 5.5 for real search space and Table 5.8 to Table 5.11 for binary search space

Description of datasets

MSE of CAPSO-MLP, PSO-MLP, GSA-MLP and ICA-MLP on all datasets

Comparison of datasets in term of accuracy (%)

Comparison of datasets in term of Sensitivity (%)

Comparison of datasets in term of Specificity (%)

Comparison of datasets in term of AUC

Experimental results on the benchmarks of Table 4.5 using PF technique

Experimental results on the benchmarks of Table 4.6 using PF technique

Comparison of BCAPSO and BCAPSO with PSO-P for the benchmarks of Table 4.6 with different penalty functions

Experimental results on the benchmarks of Table 4.5 using CD algorithm

Experimental results on the benchmarks of Table 4.6 using CD algorithm

Experimental results on the benchmarks of Table 4.5 using ICRO algorithm

Experimental results on the benchmarks of Table 4.6 using ICRO algorithm
7.8  Comparison of BCAPSO and BCAPSO using ICRO algorithm with PSO-R and MHPSO on the first seven benchmarks of Table 4.5

7.9  Comparison of BCAPSO and BCAPSO using ICRO algorithm with MHPSO on the benchmarks of Table 4.6
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE NO.</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Classification of meta-heuristic algorithms (Dreo, 2007)</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Trajectory-based method</td>
<td>14</td>
</tr>
<tr>
<td>2.3</td>
<td>Roulette wheel selections</td>
<td>22</td>
</tr>
<tr>
<td>2.4</td>
<td>GA crossover</td>
<td>24</td>
</tr>
<tr>
<td>2.5</td>
<td>GA pseudo code</td>
<td>25</td>
</tr>
<tr>
<td>2.6</td>
<td>PSO pseudo code</td>
<td>27</td>
</tr>
<tr>
<td>2.7</td>
<td>Movement of colonies toward their relevant imperialist</td>
<td>33</td>
</tr>
<tr>
<td>2.8</td>
<td>Imperialistic competition</td>
<td>34</td>
</tr>
<tr>
<td>2.9</td>
<td>ICA pseudo code</td>
<td>36</td>
</tr>
<tr>
<td>2.10</td>
<td>GSA pseudo code</td>
<td>38</td>
</tr>
<tr>
<td>3.1</td>
<td>Architecture of MLP network</td>
<td>49</td>
</tr>
<tr>
<td>3.2</td>
<td>An artificial neuron of MLP</td>
<td>50</td>
</tr>
<tr>
<td>3.3</td>
<td>Encoding a set of weights in a chromosome</td>
<td>56</td>
</tr>
<tr>
<td>3.4</td>
<td>Pseudo code of hybrid leaning of MLP and meta-heuristic algorithms</td>
<td>56</td>
</tr>
<tr>
<td>3.5</td>
<td>Flowchart of hybrid of meta-heuristic algorithms and the 0-1 MKP</td>
<td>60</td>
</tr>
<tr>
<td>4.1</td>
<td>General framework of the study</td>
<td>64</td>
</tr>
<tr>
<td>5.1</td>
<td>The object motion in the interval $\Delta t$</td>
<td>80</td>
</tr>
<tr>
<td>5.2</td>
<td>Graphical representation of (a) $\vec{A}_i$ and (b) $\vec{a}_i$</td>
<td>85</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>5.3</td>
<td>Flowchart of CAPSO for continuous and discrete search spaces</td>
<td></td>
</tr>
<tr>
<td>5.4</td>
<td>CAPSO pseudo code for continuous and discrete search spaces</td>
<td></td>
</tr>
<tr>
<td>5.5</td>
<td>Convergence performance of CAPSO, LCAPSO, GSA, PSO and LPSO on $F_3$ with $n=30$</td>
<td></td>
</tr>
<tr>
<td>5.6</td>
<td>Convergence performance of CAPSO, LCAPSO, GSA, PSO and LPSO on $F_{11}$ with $n=30$</td>
<td></td>
</tr>
<tr>
<td>5.7</td>
<td>Convergence performance of CAPSO, LCAPSO, GSA, PSO and LPSO on $F_{15}$ with $n=4$</td>
<td></td>
</tr>
<tr>
<td>5.8</td>
<td>Convergence performance of CAPSO, LCAPSO, GSA, PSO and LPSO on $F_4$ with $n=100$</td>
<td></td>
</tr>
<tr>
<td>5.9</td>
<td>Convergence performance of CAPSO, LCAPSO, GSA, PSO and LPSO on $F_7$ with $n=200$</td>
<td></td>
</tr>
<tr>
<td>5.10</td>
<td>Convergence performance of CAPSO and LCAPSO on the functions of $F_1$, $F_2$ and $F_3$ with $n=30$</td>
<td></td>
</tr>
<tr>
<td>5.11</td>
<td>Convergence performance of BCAPSO, LBCAPSO, BGSA, BPSO and LBPSO on $F_3$ with $n=5$</td>
<td></td>
</tr>
<tr>
<td>5.12</td>
<td>Convergence performance of BCAPSO, LBCAPSO, BGSA, BPSO and LBPSO on $F_9$ with $n=5$</td>
<td></td>
</tr>
<tr>
<td>5.13</td>
<td>Convergence performance of BCAPSO, LBCAPSO, BGSA, BPSO and LBPSO on $F_{22}$ with $n=4$</td>
<td></td>
</tr>
<tr>
<td>5.14</td>
<td>Convergence performance of BCAPSO, LBCAPSO, BGSA, BPSO and LBPSO on $F_{24}$ with $n=200$</td>
<td></td>
</tr>
<tr>
<td>5.15</td>
<td>Mean Absolute Error (MAE) of the best algorithms results for Table 5.3 to Table 5.5 in real search space</td>
<td></td>
</tr>
<tr>
<td>5.16</td>
<td>Mean Absolute Error (MAE) of the best algorithms results for Table 5.8 to Table 5.11 in binary search space</td>
<td></td>
</tr>
<tr>
<td>6.1</td>
<td>Flowchart of hybrid learning of CAPSO and MLP network</td>
<td></td>
</tr>
</tbody>
</table>
6.2 Training errors (MSE) of CAPSO-MLP, PSO-MLP, GSA-MLP and ICA-MLP for Hepatitis, Heart Disease, Diabetes, Breast Cancer, Parkinson’s disease and Echocardiogram

6.3 ROC Curve of training and testing data on Hepatitis and Heart Disease

6.4 Training Classification Accuracy for datasets

6.5 Testing Classification Accuracy for datasets

7.1 ICRO pseudo code

7.2 Flowchart of solving MKP using meta-heuristic algorithms

7.3 Performance of BCAPSO, LBCAPSO, BPSO, LBPSO and GA using PF technique on mknapcb1-5.100-02

7.4 Performance of BCAPSO, LBCAPSO, BPSO, LBPSO and GA using CD algorithm on mknapcb4-10.100-04

7.5 Performance of BCAPSO, LBCAPSO, BPSO, LBPSO and GA using ICRO algorithm on mknapcb1-5.100-02

7.6 Performance of BCAPSO, LBCAPSO, BPSO, LBPSO and GA using ICRO algorithm on mknapcb9-30.500-29

7.7 Average error of mean profit using PF, CD and ICRO methods on the benchmarks of Table 4.5 (Fairly moderate benchmarks)

7.8 Average error of mean profit using PF, CD and ICRO methods on the benchmarks of Table 4.6 (Hard benchmarks)
# LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td>Artificial Bee Colony</td>
</tr>
<tr>
<td>ACC</td>
<td>Accuracy</td>
</tr>
<tr>
<td>ACO</td>
<td>Ant Colony Optimization</td>
</tr>
<tr>
<td>AE</td>
<td>Average Error</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>AIS</td>
<td>Artificial Immune System</td>
</tr>
<tr>
<td>ANNs</td>
<td>Artificial Neural Networks</td>
</tr>
<tr>
<td>APSO</td>
<td>Adaptive Particle Swarm Optimization</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under Curve</td>
</tr>
<tr>
<td>BA</td>
<td>Bootstrap Algorithm</td>
</tr>
<tr>
<td>BCAPSO</td>
<td>Binary Centripetal Accelerated Particle Swarm Optimization</td>
</tr>
<tr>
<td>BGSA</td>
<td>Binary Gravitational Search Algorithm</td>
</tr>
<tr>
<td>BO</td>
<td>Bees Optimization</td>
</tr>
<tr>
<td>BP</td>
<td>Back-Propagation algorithm</td>
</tr>
<tr>
<td>BPSO</td>
<td>Binary Particle Swarm Optimization</td>
</tr>
<tr>
<td>CAPSO</td>
<td>Centripetal Accelerated Particle Swarm Optimization</td>
</tr>
<tr>
<td>CAPSO-MLP</td>
<td>Particle Swarm Optimization Multi-Layer Perceptron</td>
</tr>
<tr>
<td>CD</td>
<td>Check-and-Dropt</td>
</tr>
<tr>
<td>CEM</td>
<td>Cross Entropy Method</td>
</tr>
<tr>
<td>CLPSO</td>
<td>Comprehensive Learning Particle Swarm Optimization</td>
</tr>
<tr>
<td>COPs</td>
<td>Combinatorial Optimization Problems</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>CP</td>
<td>Charged Particle</td>
</tr>
<tr>
<td>CS</td>
<td>Cuckoo Search</td>
</tr>
<tr>
<td>CSS</td>
<td>Charged System Search</td>
</tr>
<tr>
<td>DSA</td>
<td>Differential Search Algorithm</td>
</tr>
<tr>
<td>DE</td>
<td>Differential Evolution</td>
</tr>
<tr>
<td>DMS-PSO</td>
<td>Dynamic Multi-Swarm Particle Swarm Optimization</td>
</tr>
<tr>
<td>FA</td>
<td>Firefly Algorithm</td>
</tr>
<tr>
<td>FFNN</td>
<td>Feed–Forward Neural Network</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>GA</td>
<td>Genetic Algorithm</td>
</tr>
<tr>
<td>GbSA</td>
<td>Galaxy-based Search Algorithm</td>
</tr>
<tr>
<td>GLS</td>
<td>Guided Local Search</td>
</tr>
<tr>
<td>GPSO</td>
<td>Global-topology Particle Swarm Optimization</td>
</tr>
<tr>
<td>GSA</td>
<td>Gravitational Search Algorithm</td>
</tr>
<tr>
<td>GSA-MLP</td>
<td>Gravitational Search Algorithm Multi-Layer Perceptron</td>
</tr>
<tr>
<td>GSO</td>
<td>Glowworm Swarm Optimization</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
</tr>
<tr>
<td>HMO</td>
<td>Honey-bee Mating Optimization</td>
</tr>
<tr>
<td>HPSO-TVAC</td>
<td>Hierarchical Particle Swarm Optimizer with Time-Varying Acceleration Coefficients</td>
</tr>
<tr>
<td>HS</td>
<td>Harmony Search</td>
</tr>
<tr>
<td>ICA</td>
<td>Imperialist Competition Algorithm</td>
</tr>
<tr>
<td>ICA-MLP</td>
<td>Imperialist Competition Algorithm Multi-Layer Perceptron</td>
</tr>
<tr>
<td>ICRO</td>
<td>Improved Check-and-Repair Operator</td>
</tr>
<tr>
<td>ILS</td>
<td>Iterated Local Search</td>
</tr>
<tr>
<td>IWD</td>
<td>Intelligent Water Drops</td>
</tr>
<tr>
<td>KH</td>
<td>Krill Herd</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LBCAPSO</td>
<td>Local-topology Binary Centripetal Accelerated Particle Swarm Optimization</td>
</tr>
<tr>
<td>LCAPSO</td>
<td>Local-topology Centripetal Accelerated Particle Swarm Optimization</td>
</tr>
<tr>
<td>LPSO</td>
<td>Local topology Particle Swarm Optimization</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean Absolute Error</td>
</tr>
<tr>
<td>MKP</td>
<td>Multidimensional Knapsack Problem</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>MLP</td>
<td>Multi-Layer Perceptron</td>
</tr>
<tr>
<td>MOGA</td>
<td>Multi-Objective Genetic Algorithm</td>
</tr>
<tr>
<td>MS</td>
<td>Monkey Search</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>PF</td>
<td>Penalty Function</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
<tr>
<td>PSO-MLP</td>
<td>Particle Swarm Optimization Multi-Layer Perceptron</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
</tr>
<tr>
<td>RFD</td>
<td>River Formation Dynamics</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristics</td>
</tr>
<tr>
<td>RSO</td>
<td>Reactive Search Optimization</td>
</tr>
<tr>
<td>SA</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>SD</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>SO</td>
<td>Spiral Optimization</td>
</tr>
<tr>
<td>SS</td>
<td>Scatter Search</td>
</tr>
<tr>
<td>TLBO</td>
<td>Teaching-Learning-Based Optimization</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
<tr>
<td>TS</td>
<td>Tabu Search</td>
</tr>
<tr>
<td>UCI</td>
<td>University of California at Irvine</td>
</tr>
<tr>
<td>VNS</td>
<td>Variable Neighborhood Search</td>
</tr>
<tr>
<td>Acronym</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>VPSO</td>
<td>Von–Neumann topology Particle Swarm Optimization</td>
</tr>
<tr>
<td>WNN</td>
<td>Wavelet Neural Network</td>
</tr>
</tbody>
</table>
# LIST OF APPENDICES

<table>
<thead>
<tr>
<th>APPENDIX</th>
<th>TITLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Details of Functions of Table 4.3</td>
<td>167</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

1.1 Overview

Machine Learning (ML) (Shavlik and Dietterich, 1990; Michie et al., 1994; Mitchell, 1997, Bishop, 2007; Marsland, 2009) is a branch of Artificial Intelligence (AI) concerned with many learning algorithms and problems. Different ML algorithms have been successfully employed to solve real-life problems. The goal of ML research is computer learning based on training data to recognize complex patterns of datasets, or to make intelligent decisions based on data. In ML, optimization provides a valuable framework for thinking about, formulating and solving many problems.

Optimization problems have located at the heart of most ML approaches. Many algorithms from the class of exact and approximate optimization algorithms have been presented to deal with ML applications. However, exact optimization algorithms such as dynamic programming, branch-and-bound and backtracking (Neapolitan and Naimipour, 2004; Tanaka et al. 2009; Ferrer et al., 2009; Manerba and Mansini, 2012; Smet et al., 2012) have shown good performance in addressing ML applications, they are not efficient in a high-dimensional search space. In the applications, the search space increases exponentially with the problem size, hence solving these problems using the algorithms (such as exhaustive search) is not practical. Therefore, many researchers are interested in utilizing approximate algorithms like meta-heuristic algorithms in this regard.
Artificial Immune System (AIS) (Farmer et al., 1986), Genetic Algorithm (GA) (Holland, 1975; Tang, 1996), Ant Colony Optimization (ACO) (Dorigo et al., 1996), Particle Swarm Optimization (PSO) (Kennedy and Eberhart, 1995; Shi and Eberhart, 1998), Artificial Bee Colony (ABC) (Karaboga, 2005), Imperialistic Competitive Algorithm (ICA) (Atashpaz-Gargari and Lucas, 2007), Gravitational Search Algorithm (GSA) (Rashedi et al., 2009) and Charged System Search (CSS) (Kaveh and Talatahari; 2010) are samples of meta-heuristic algorithms.

The meta-heuristic algorithms have applied as learning algorithm in for tackling complex problem such as neural network learning (Dehuri et al., 2011; Qasem and Shamsuddin, 2011), image processing (Lu and Chen, 2008; Yang, 2011), function optimization (Kaveh and Talatahari, 2010; Rashedi et al., 2010), data mining (Sousa et al., 2004; Freita and Timmis, 2007), pattern recognition (Senaratne et al., 2009; Zhao and Davis, 2011), control objectives (Baojiang and Shiyong, 2007; Karakuzu, 2009; Xie et al., 2009) and combinatorial optimization problems (Al-Dulaimi and Ali, 2008; Defersha and Chen, 2010; Angelelli et al., 2010).

Even though, they have been illustrated good performance, there is no a specific algorithm to find the best solution for all problems in continuous (real) and discrete (binary) search spaces. In other words, some algorithms have a better solution for a number of particular problems. Therefore, searching for a new meta-heuristic algorithm which can operate on two-valued functions, real and binary search spaces, would be beneficial.

In this thesis, the proposed methods of Centripetal Accelerated Particle Swarm Optimization (CAPSO), Local topology of Centripetal Accelerated Particle Swarm Optimization (LCAPSO), Binary Centripetal Accelerated Particle Swarm Optimization (BCAPSO) and Local topology of Binary Centripetal Accelerated Particle Swarm Optimization (LBCAPSO) are proposed for real and binary search spaces. The methods are evaluated by some ML applications in continuous and discrete search spaces such as function optimizations, Multi-Layer Perceptron (MLP) learning for classification problems and Multi-dimensional Knapsack Problem (MKP). The rationale of proposing this study is given in the problem background.
followed by the thesis statement with research questions, goal of the study, objectives, scope and importance of the research.

1.2 Problem Background

Traditional algorithms such as branch-and-bound, dynamic programming, backtracking which are in the class of exact algorithms are inefficient in solving many high-dimensional optimization problems of ML. In these problems, the search space grows exponentially with the problem size hence; the exhaustive search is not practical using the algorithms. Also, the algorithms are inflexible to adapt a solution with a problem (Chan and Tiwari, 2007). In these algorithms, a problem is modeled in such a way that can be solved by these algorithms. This generally requires making several assumptions which might not be easy to validate in many situations. Therefore, a set of more adaptable and flexible algorithms are required to overcome these limitations.

Based on this motivation, a numerous algorithms inspired by nature have been proposed in the literature. Among them, meta-heuristic algorithms have shown satisfactory abilities to handle such problems. In these algorithms, the goal is to explore efficiently the search space in order to find (near-) optimal solutions. These algorithms have various advantages (Jin and Branke, 2005; Du and Li, 2008, Zhan et al., 2009, Sarıçiçek and Çelik, 2011; Valdez et al., 2011, Mezmaz et al., 2011; Kim et al., 2012) to name a few:

1. They are robust and can adapt solutions with changing conditions and environment.
2. They can be applied in solving complex multimodal problems.
3. They may incorporate mechanisms to avoid getting trapped in local optima.
4. They are not problem-specific algorithm.
5. These algorithms are able to find promising regions in a reasonable time due to exploration and exploitation ability.
6. They can be easily employed in parallel processing.

To achieve the above advantages and to have better solution in different applications, many meta-heuristic population-based algorithms have been proposed so far and employed in many ML problems.

GA is one the oldest meta-heuristic algorithms. It has been widely used in ML (Goldberg, 1989; Shapiro, 2001). A combined ML with GA was proposed for controller design by Filipic (1999). Also, a general method was presented for identification of an optimal non-linear mixed effects model (Bies et al., 2006). This included structural, inter-individual random effects and residual error models using ML and GA. In other research, Sarkar et al. (2012) offered an accuracy-based learning system called DTGA (Decision Tree and GA) to enhance the prediction accuracy of classification problems. Moreover, a Two-stage Genetic Clustering Algorithm (TGCA) was suggested by He and Tan (2012) to determine the appropriate number of clusters and partition of dataset.

D’Souza et al. (2012) used several meta-heuristic algorithms such as Simulated Annealing (SA), PSO, GA and AIS to optimize Dial-A-Ride Problem (DARP). From the results, it could be concluded that AIS method provided more efficient optimal solutions. Al-Obeidat et al. (2010) developed PSO for PROAFTN which is a classification method and belongs to the class of supervised learning algorithms. The method applied PSO to elicit the PROAFTN parameters during the learning process. To evaluate the quality of approach, it was tested on some datasets and compared with several ML techniques. The method had considerably performance better than other ML techniques used. Furthermore, a hybrid of improved PSO algorithm with Wavelet Neural Network (WNN) was introduced (Yue-bo et al., 2012) to simulate the aerodynamic model for flight vehicles. The proposed method was compared with some well-known method such as the hybrid of GA with WNN and SVM. The simulated results indicated that the presented method has more efficiency than the others for aerodynamic modeling.

Another meta-heuristic algorithm applied in ML is ACO. Azar and Vybihal (2011) proposed a method using ACO to optimize the accuracy of software quality
predictive models for classification new data. In other study, Loyola et al. (2012) presented an approach to predict web user behavior using learning-based ACO.

Xu and Duan (2010) provided a shape-matching approach to visual target recognition for aircraft at low altitude using ABC algorithm. Also, Sulaiman et al. (2012) employed a hybrid of ABC and Least Square Support Vector Machine (LS-SVM) for solving real and reactive power tracing problem. The compared results with LS-SVM, the hybrid of GA and SVM demonstrated that the proposed method was more efficient than others in terms of determining the optimal values of hyper-parameters of LS-SVM.

Tayefeh-Mahmoudi et al. (2009) employed ICA to optimize the weights of MLP network for classification problems and compared the results with PSO, GA, Resilient Back-Propagation (RPROP) and Min Finder. The results illustrated that ICA performed better results.

Also, GSA was applied for function optimizations by Rashedi et al. (2009). The algorithm offered a better performance than PSO and GA in many cases. In another study, Bahrololoum et al. (2012) used GSA for a prototype classifier in multiclass datasets. The results of proposed method were compared with PSO, ABC and nine other classifiers on some well-known datasets. The results indicated that GSA was more efficient than the others.

Although the mentioned algorithms have obtained satisfactory results in various fields of ML, there are some unavoidable disadvantages. For instance, GA has the inherent drawbacks of prematurity convergence (Leung et al., 1997; Hrstka and Kučerová, 2004; Hong et al., 2011; Pavez-Lazo and Soto-Cartes, 2011) and unpredictable results. Also; it uses complex functions in selection and crossover operators and sometimes, the encoding scheme is difficult (Moslemipour et al., 2012). PSO suffers from trapping into local optima and slow convergence speed (Deep, M. Thakur, 2007 (a), 2007 (b); Tsoulos, 2008; Zhan et al., 2009; Zhan et al., 2011; Gao, 2012), whereas GSA and ICA take long computational time to achieve
the results. Furthermore, some of these algorithms have several parameters to tune and often parameters setting is a challenge for various optimization problems (Tashkova, 2011). Meanwhile, none of meta-heuristic algorithms are able to present a higher performance than others in solving all problems. Another noteworthy point is that many problems are expressed in a binary representation. In other words, some solutions are encoded binary form or some problems are binary in nature. Nevertheless, some meta-heuristic algorithms are designed for only continuous (real) or discrete (binary) search space and sometimes, they have good performance just on one of the search spaces. For example, ICA and the original of ACO have been designed for continuous and discrete search space respectively. Also, binary PSO (Kennedy and Eberhart, 1997) has some inherent disadvantages such as poor convergence rate and failure to achieve desired results (Nezamabadi-pour et al., 2008) which bring about a decrease in performance of algorithm in the binary search space. Therefore, the enhancement of performance of previous meta-heuristics or even introduction of new ones in minimizing the disadvantages seems to be necessary. Hence, a new optimization meta-heuristic algorithm has been proposed based on Newtonian’s motion laws and PSO algorithm to improve convergence speed and to avoid trapping into local optimum and setting many parameters. The algorithm is named Centripetal Accelerated Particle Swarm Optimization (CAPSO) and can be applied for both continuous and discrete high-dimensional search spaces.

1.3 Research Statement with Research Questions

Traditional optimization algorithms cannot provide proper results for ML problems with high-dimensional search space since the search space exponentially increases with the size of problem and exhaustive search is impractical. Also, existing meta-heuristic algorithms suffer from different drawbacks such as lack of providing optimum solution for all problems, getting stuck in local optima, tuning many parameters, slow convergence rate and high run-time. Also, some meta-heuristic algorithms are designed for only continuous (real) or discrete (binary) search space and sometimes, they have good performance only in one of the search
spaces. However, the algorithms are robust and have the ability of adapting with changing environment.

Therefore, more works are still required to develop the performance of meta-heuristic algorithms in ML. Hence, new meta-heuristic algorithms are introduced in the study for both continuous and discrete search spaces to cope with the shortcomings.

Consequently, based on the above issues, the main research question is:

Are the proposed meta-heuristic algorithms beneficial for learning process enhancement in ML?

Thus, the following issues need to be addressed:

1. Could the proposed methods improve the learning process and accelerate the convergence rate in ML?
2. Is it possible that the algorithms need no parameters setting?
3. Could the proposed algorithms have good performance in both real and binary search spaces?

1.4 Goal of the Research

The aim of this research is to propose an improved scheme of Particle Swarm Algorithm (PSO) based on the Newtonian’s motion laws, which is called Centripetal Accelerated Particle Swarm Optimization (CAPSO) to accelerate the learning and convergence procedure of classifiers in real and binary search spaces.
1.5 Objectives of the Research

In order to answer the above questions, the objectives of this thesis have been identified as:

1. To propose efficient meta-heuristic algorithms for both real and binary search spaces.
2. To improve the performance of meta-heuristic algorithms for optimizing non-linear functions in both real and binary search spaces.
3. To enhance ANN learning using the proposed method.
4. To evaluate the performance of combinatorial optimization problems in binary search space.

1.6 Scope of the Study

To achieve the mentioned objectives, the scope of this study is bounded as follows:

1. Twenty three unimodal and multimodal high-dimensional non-linear benchmark functions have been chosen to validate and to compare the performance of proposed algorithms with some meta-heuristic algorithms in real search space (Yao et al., 1999; Rashedi et al., 2009).
2. Twenty four unimodal and multimodal high-dimensional non-linear benchmark functions have been selected to assess the efficiency of proposed algorithms in binary search space (Yao et al., 1999; Rashedi et al., 2010).
3. Six datasets on binary class classification problems (http://www.ics.uci.edu/~mlearn/MLRepository.html) have been used to validate the hybrid learning of proposed algorithm with MLP. The datasets are: Hepatitis, Heart Disease, Pima Indian Diabetes, Wisconsin Prognostic Breast Cancer, Parkinson’s disease and Echocardiogram (Heart attack). The performance of the proposed method is measured
based on convergence towards error, Sensitivity, Specificity, and classification accuracy.

4. Twenty five datasets for MKP (OR-Library: http://people.brunel.ac.uk/~mastjjb/jeb/orlib/mknapinfo.html) are applied to test the performance of proposed methods for combinatorial optimization problems in binary search space.

5. All meta-heuristics used in the study are in the class of population-based global search meta-heuristic algorithms.

6. The programs have been customized, developed and applied to the problems using MATLAB R2011a software.

1.7 Importance of the Study

The study investigates the capabilities of meta-heuristic algorithms in Machine Learning (ML). The performance of the proposed methods is evaluated using some applications in ML such as function optimization, Multi-Layer Perceptron (MLP) learning for pattern classification tasks and solving the 0-1 Multidimensional Knapsack Problem (MKP). The approaches are tested to detect whether the methods are efficient in the applications.

1.8 Thesis Organization

This thesis consists of eight chapters. The first is the introductory chapter. The second and third chapters describe the background as well as the previously published work in the field of meta-heuristic algorithms and Machine Learning (ML). The fourth chapter describes the research methodology of this study. Chapter 5, 6 and 7 provide the proposed methods and their analysis of results on some ML applications. Finally, the summary of this study is presented in Chapter 8. The details of each chapter are as follows:
Chapter 2, *Meta-heuristic Algorithms*, provides a review on concept and techniques applied in meta-heuristic algorithms. Also, related works are elucidated in real and binary search spaces. Finally, the discussion and summary of this chapter are given.

Chapter 3, *Machine Learning and Its applications*, presents ML algorithms and the related problems. Some ML applications are reviewed in this chapter such as ANN learning, combinatorial optimization problems in binary search space and the optimization of unimodal and multimodal high-dimensional function. Moreover, a broad overview about the basic concepts and traditional techniques of ANN learning are described especially, the hybrid learning of MLP network with meta-heuristics is elucidated in details. Furthermore, the hybrid of the 0-1 MKP and meta-heuristics is discussed in this chapter. Lastly, the chapter will be finished by a summary.

Chapter 4, *Research Methodology*, comprises of research methodology, a general framework for each phase of the study and descriptions about the overall solving-tools and standard techniques adopted.

Chapter 5, *Centripetal Accelerated Particle Swarm Optimization (CAPSO) in Real and Binary Search Spaces*, presents the encoding of the proposed algorithms and evaluates their performance using some non-linear benchmark functions in the search spaces.

Chapter 6, *Enhancement of Multi-Layer Perceptron (MLP) Learning Using Centripetal Accelerated Particle Swarm Optimization (CAPSO)*, uses the hybrid learning of proposed algorithm and MLP network to improve the ability of the network in term of accuracy for classification problems. Finally, the results and discussion of the proposed method on several medical datasets are compared with some previous methods in the literature.

Chapter 7, *Binary Centripetal Accelerated Particle Swarm Optimization (BCAPSO) For Solving 0-1 Multidimensional Knapsack Problem (MKP)*, presents
the methods of solving the 0-1 MKP using meta-heuristic algorithms. Three methods of Penalty Function (PF) technique, Check-and-Dropt (CD) and Improved Check-and-Repair Operator (ICRO) algorithms are proposed to improve the 0-1 MKP solutions. Also, the performance of each method is compared, analyzed and benchmarked with previous methods.

Chapter 8, *Conclusion and Future Works*, discusses and highlights the contributions and findings of the research work and provides suggestions and recommendations for future studies.
REFERENCES


