

**THE DEVELOPMENT AND APPLICATION OF EVOLUTIONARY  
COMPUTATION-BASED LAYERED ENCODING CASCADE  
OPTIMIZATION MODEL**

**by**

**NEOH SIEW CHIN**

**Thesis submitted in fulfilment of the  
requirements for the degree  
of Doctor of Philosophy**

**Jan 2010**

## **ACKNOWLEDGEMENTS**

First of all, I would like to express my gratitude to my supervisor; Dr Zalina Abdul Aziz and co-supervisors; Associate Professor Dr Norhashimah Morad and Professor Lim Chee Peng for their insightful guidance and encouragement throughout the research. Thanks to their supervision and invaluable assistance in accomplishing this research work.

My appreciation also goes to En. Arjuna Marzuki, a lecturer from the School of Electrical and Electronic Engineering, USM for allowing the use of the MMIC Circuit Design that was originated by him and his colleagues. Besides, I am also grateful to the Intel Corporation for giving me an opportunity to be involved in the Intel-USM collaboration program.

I would like to thank my beloved parents; Neoh Yong Seng and Lim Beng Choo for their patience, encouragement, and support. In addition, I wish to extend my gratitude to my only sister, Neoh Siew Bee for motivating me for all these years. In this special session, I would also like to thank all my colleagues and friends for their friendship, assistance, and enlightening discussions on many occasions.

Finally, I wish to thank the Ministry of Science, Technology, and Innovation, Malaysia for its financial support in the form of National Science Fellowship.

# TABLE OF CONTENTS

<b>ACKNOWLEDGEMENTS</b> .....	ii
<b>TABLE OF CONTENTS</b> .....	iii
<b>LIST OF TABLES</b> .....	ix
<b>LIST OF FIGURES</b> .....	xi
<b>LIST OF ABBREVIATION</b> .....	xiv
<b>ABSTRAK</b> .....	xvi
<b>ABSTRACT</b> .....	xviii
<b>CHAPTER 1 – INTRODUCTION</b>	
1.1 Preliminaries.....	1
1.2 Knowledge, Representation and Search .....	3
1.3 Hybridization.....	5
1.4 Problems and Motivation .....	6
1.5 Research Objectives .....	8
1.6 Research Scope and Approach .....	10
1.7 Thesis Outline.....	13
<b>CHAPTER 2 – LITERATURE REVIEW</b>	
2.1 Introduction .....	15
2.2 Genetic Algorithms .....	15
2.2.1 GA Terminology and Basic Mechanism.....	16

2.2.1(a)	Chromosome Encoding and Representation .....	17
2.2.1(b)	Fitness Function .....	19
2.2.1(c)	Penalty Function.....	20
2.2.1(d)	Selection Function and Genetic Operators.....	21
2.2.2	Theory of Schemata.....	24
2.2.3	GA basic solution approaches.....	26
2.2.4	GA advantages and limitations .....	29
2.3	Particle Swarm Optimization .....	30
2.3.1	PSO Characteristic.....	31
2.3.2	The Basic PSO Algorithm .....	31
2.3.3	PSO Parameter Setting.....	32
2.3.4	Attractive Properties of PSO.....	33
2.4	Search Strategies .....	33
2.4.1	Elitist Strategy.....	34
2.4.2	Hill-Climbing.....	35
2.4.3	Divide and Conquer .....	36
2.4.3(a)	Decomposition .....	38
2.5	Knowledge Representation.....	39
2.6	Hybrid Intelligent Systems.....	42
2.7	Summary .....	45

### **CHAPTER 3 – THE DEVELOPMENT OF THE EVOLUTIONARY LAYERED**

#### **ENCODING CASCADE OPTIMIZATION APPROACH**

3.1	Introduction .....	46
3.2	Encodings .....	47
3.3	Multidimensionality and Slicing .....	49

3.4	The Development of Layered Encoding Structure .....	50
3.5	Layered Encoding Structure for Memetic and Hybrid Algorithms .....	56
3.6	The Property of Layered Encoding Structure .....	58
3.7	The Development of Layered Encoding Cascade Optimization .....	58
3.7.1	The Cascade Architecture, Strategy and Optimization .....	59
3.7.2	Multi-Population Genetic Algorithm .....	64
3.8	The LECO Model .....	67
3.9	Summary .....	70

## **CHAPTER 4 – GA AND PSO-BASED LECO MODEL FOR MULTI-DECISION**

### **REPRESENTATION AND PLANNING**

4.1	Introduction .....	72
4.2	Decision Making and Multi-decision Approach .....	73
4.3	Thermal Power Generation Scheduling .....	75
4.3.1	Problem Description .....	77
4.3.2	Constraints and Mathematical Formulation .....	78
4.3.2(a)	Demand Constraint .....	79
4.3.2(b)	Operation Constraint of the Generator .....	80
4.3.2(c)	Start-up Generator Constraint .....	80
4.3.2(d)	Evaluation Function .....	81
4.3.3	Layered Encoding Structure for Thermal Power Scheduling .....	81
4.3.4	Cascade First Layer-Second Layer Optimizer Operation .....	82
4.3.5	Results and Discussion .....	88
4.4	High-Mix-Low-Volume (HMLV) Product-Mix Planning .....	91
4.4.1	Functional Relationship in HMLV Product Mix Planning .....	94
4.4.2	LECO for HMLV Product-Mix Optimization .....	95

4.4.3	Constraints Formulation .....	100
4.4.3(a)	Demand Constraint.....	100
4.4.3(b)	Production Loading Constraint .....	100
4.4.3(c)	Capacity Constraint.....	101
4.4.4	First Layer and Second Layer Optimizers .....	101
4.4.4(a)	GA Operations .....	102
4.4.4(b)	PSO Operations.....	104
4.4.4(c)	Termination Criterion.....	105
4.4.5	Evaluation Functions .....	106
4.4.5(a)	Normalization of the Objective Function.....	110
4.4.6	Results and Discussion .....	111
4.4.6(a)	Case 1 .....	116
4.4.6(b)	Case 2.....	117
4.4.6(c)	Most Probable Capacity Distribution .....	117
4.4.6(d)	Improvement .....	122
4.4.6(e)	Computational Time.....	122
4.5	Summary .....	123

## **CHAPTER 5 – GA-PSO LECO MODEL FOR MULTI-RESOLUTION**

### **REPRESENTATION, PARAMETER OPTIMIZATION AND**

### **HUMAN-MACHINE INTERACTION**

5.1	Introduction .....	125
5.2	Multi-resolution Approach.....	126
5.3	Parameter Optimization.....	128
5.4	Interactive Optimization.....	129
5.5	Tennessee Eastman Chemical Process Optimization .....	130
5.5.1	Layered Encoding Structure for Resolution Representation.....	134

5.5.2	GA-PSO LECO Model for Tennessee Eastman Parameter Optimization	135
5.5.3	Results and Discussion	137
5.6	MMIC Low Noise Amplifier Circuit Design Optimization using ADS	139
5.6.1	Layered Encoding Structure for MMIC Design Variables	144
5.6.2	Interactive GA-PSO LECO Approach	145
5.6.3	Results and Discussion	150
5.7	Summary	152

**CHAPTER 6 – EVOLUTIONARY MODEL FOR MULTI-OBJECTIVE PARETO OPTIMIZATION**

6.1	Introduction	153
6.2	Multi-objective Optimization	154
6.3	Knapsack Problem	156
6.4	String-based Layered Encoding Structure for Schema Representation	157
6.5	The Evolutionary Model for Pareto Optimization	159
6.5.1	Hill-Climbing	161
6.5.2	Pareto Dominance Optimization	162
6.5.3	GA-PSO LECO Operation	163
6.5.4	Non-dominated Spread Lengthen	166
6.5.5	Specification and Termination Criterion	168
6.5.6	Results and Discussion	169
6.6	Summary	177

**CHAPTER 7 – CONCLUSIONS AND FUTURE WORK**

7.1	Contributions and Summary of the Research	178
7.2	Suggestions of Further Work	182

<b>REFERENCES</b> .....	184
 <b>APPENDICES</b>	
Appendix A: Power production cost for different combinations of GA and PSO cascade optimization in 30 repetition runs.....	204
Appendix B: HMLV data (specifications).....	205
Appendix C: HMLV results for different combinations of GA and PSO cascade optimization (33 items and 4 weeks) .....	219
Appendix D: HMLV results for different combinations of GA and PSO cascade optimization (100 items and 4 weeks) .....	226
Appendix E: Tennessee Eastman parameter setting (30 repetition runs).....	233
Appendix F: MMIC parameter setting (5 repetition runs).....	234
Appendix G: Pareto hypervolume covered by different multi-objective optimizers for knapsack problem (30 repetition runs) .....	235
<b>LIST OF PUBLICATIONS</b> .....	239

## LIST OF TABLES

		<b>Page</b>
Table 2.1	Crossover operation in binary string chromosomes	23
Table 2.2	Mutation operation in binary string chromosomes	24
Table 3.1	An example of strings and its fitness value (Adapted from Goldberg, 1989)	54
Table 4.1	Daily electricity load demand according to time period	78
Table 4.2	Information of thermal power generator	78
Table 4.3	Population size and maximum generation for all combinations of LECO model	88
Table 4.4	Power production cost for different GA and PSO optimizations	89
Table 4.5	Comparison of power production cost for GA-PSO <sup>IM</sup> and Williams (1999)	90
Table 4.6	Population size and the maximum number of generation for different models	102
Table 4.7	Genetic parameter setting for crossover and mutation	104
Table 4.8	Original capacity distribution and performance analysis for Case 1 and Case 2	114
Table 4.9	Performance comparison for all different models in Case 1 and Case 2	115
Table 4.10	Product-Mix quality for different models	116
Table 4.11	Improvements brought by different models as compared to the original product-mix	123
Table 5.1	Process operating constraints (Adapted from Downs and Vogel, 1993)	133
Table 5.2	Parameter range used for TE problem in this study	133
Table 5.3	Parameter comparison between GA-PSO LECO model and the original Ricker's decentralized model	138
Table 5.4	Synthesis setup of design variables	142
Table 5.5	Required specifications for amplifier optimization	142
Table 5.6	Optimized design variables for MMIC amplifier using interactive GA-PSO LECO approach	150

Table 5.7	Parameter value of different optimizers	150
Table 5.8	Interactive GA-PSO LECO model performance in five repetition runs	151
Table 6.1	Specification and termination criterion for each operation in the proposed evolutionary model	169
Table 6.2	The amount of hypervolume that column optimizer cover the row optimizer—case 100 items, 2 knapsacks	176
Table 6.3	The amount of hypervolume that column optimizer cover the row optimizer—case 250 items, 2 knapsacks	176
Table 6.4	The amount of hypervolume that column optimizer cover the row optimizer—case 500 items, 2 knapsacks	176
Table 6.5	The amount of hypervolume that column optimizer cover the row optimizer—case 750 items, 2 knapsacks	176

## LIST OF FIGURES

	<b>Page</b>	
Figure 1.1	Flow chart of the overall work stages involved in the research	12
Figure 2.1	An example of schema	25
Figure 2.2	Topographic view of hill-climbing search (shown by arrow) used to improve the current state space (Modified from Russel and Norvig, 2003)	36
Figure 2.3	Divide and Conquer (Adapted from Aho <i>et al.</i> , 1974)	37
Figure 2.4	Representation of knowledge using in a semantic network (Adapted from Turban <i>et al.</i> , 2005)	40
Figure 2.5	An example of frame structure with slots (Adapted from Padhy, 2005)	41
Figure 2.6	Hierarchy of frames describing vehicles (Adapted from Turban <i>et al.</i> , 2005)	42
Figure 3.1	Decomposition of 3D representation into layers	51
Figure 3.2	A layered encoding structure for system parameter settings in different periods	53
Figure 3.3	Layered encoding structure: Communication and Optimization	57
Figure 3.4	The cascade architecture (Adapted from Fahlman and Lebiere, 1991)	60
Figure 3.5	Ring migration topology (Adapted from Chipperfield <i>et al.</i> , 1994)	65
Figure 3.6	Neighborhood migration topology (Adapted from Chipperfield <i>et al.</i> , 1994)	65
Figure 3.7	Unrestricted migration topology (Adapted from Chipperfield <i>et al.</i> , 1994)	66
Figure 3.8	The general flow of operation for LECO model	68
Figure 4.1	The layered matrix encoding structure for power generation schedule in the proposed model	79
Figure 4.2	Cascade first layer-second layer optimizer model for solving unit commitment and power dispatch problem	84
Figure 4.3	Randomly Selected Crossover	86
Figure 4.4	Randomly Selected Mutation	86
Figure 4.5	Functional relationship in the development of product-mix	95

Figure 4.6	The product-mix representation structure in the LECO model	97
Figure 4.7	A flow chart of the LECO model for HMLV	98
Figure 4.8	Randomly selected crossover for HMLV product mix planning	103
Figure 4.9	Randomly selected mutation for HMLV product mix planning	103
Figure 4.10	Weekly mutation for HMLV product mix planning	104
Figure 4.11	An example of capacity consumption of week $k$	108
Figure 4.12	Use of the $UBP_k$ function to provide a balanced workload	109
Figure 4.13	Original capacity distribution of Case 1 and Case 2	113
Figure 4.14	Most probable capacity distribution of Case 1 in GA1, GA2, GA-GA, GA-PSO, PSO-PSO, and PSO-GA	118
Figure 4.15	Most probable capacity distribution of Case 1 in GA-PSO <sup>IM</sup>	119
Figure 4.16	Most probable capacity distribution of Case 2 in GA1, GA2, GA-GA, GA-PSO, PSO-PSO, and PSO-GA	120
Figure 4.17	Most probable capacity distribution of Case 2 in GA-PSO <sup>IM</sup>	121
Figure 5.1	Schematic diagram for TE test problem (Adapted from Downs and Vogel, 1993)	131
Figure 5.2	Decentralized control model of the TE process developed by Ricker (1996)	132
Figure 5.3	Layered encoding structure for representing parameter resolution in TE	135
Figure 5.4	GA-PSO LECO model for TE process optimization	136
Figure 5.5	Mean total operating cost for 30 sample runs of GA-PSO LECO model	139
Figure 5.6	RC feedback amplifier circuit	141
Figure 5.7	ADS simulink model for MMIC low noise amplifier circuit design	143
Figure 5.8	The layered representation of design variables in amplifier optimization	145
Figure 5.9	The interactive GA-PSO LECO flow for the MMIC circuit design	147
Figure 6.1	Separation of chromosome string into two-layered encoding representation structure	158
Figure 6.2	The LGAPSO model for solving multi-objective 0/1 knapsack problem	160

Figure 6.3	GA-PSO LECO operation flow for 0/1 multi-objective knapsack problem	165
Figure 6.4	Effect of local search trap towards Pareto front	166
Figure 6.5	End points of Pareto front	167
Figure 6.6	Best Pareto front of each optimizer in different size knapsacks	170
Figure 6.7	Box-plot hypervolume comparison of each optimizer based on the 30 repetition runs	173

## LIST OF ABBREVIATION

<b>AC</b>	Available Capacity
<b>ACO</b>	Ant Colony Optimization
<b>ADS</b>	Agilent Advanced Design System
<b>AI</b>	Artificial Intelligence
<b>CVC</b>	Conversion Capacity
<b>GA</b>	Genetic Algorithm
<b>GAP</b>	Gap between <i>AC</i> and <i>UC</i>
<b>GGAP</b>	Generation Gap
<b>HMLV</b>	High-Mix-Low-Volume
<b>IEC</b>	Interactive Evolutionary Computation
<b>LECO</b>	Layered Encoding Cascade Optimization
<b>LGAPSO</b>	Layered and Pareto-based GA-PSO Model
<b>LGP</b>	Lexicographic Goal Programming
<b>LP</b>	Linear Programming
<b>MCDM</b>	Multi-Criteria Decision Making
<b>MDDM</b>	Multi-Decision Decision Making
<b>MMIC</b>	Microwave Monolithic Integrated Circuit
<b>MOEAs</b>	Multi-Objective Evolutionary Algorithms
<b>MPGA</b>	Multi-Population Genetic Algorithm
<b>NPGA</b>	Niched-Pareto Genetic Algorithm
<b>NSGA</b>	Non-dominated Sorting Genetic Algorithm
<b>OVP</b>	Overloaded Penalty
<b>PAES</b>	Pareto Archived Evolution Strategy
<b>PCS</b>	Personal Communication Service
<b>Pos</b>	Position of Particle in PSO

<b>PSO</b>	Particle Swarm Optimization
<b>RWS</b>	Roulette Wheel Selection
<b>SPEA</b>	Strength Pareto Evolutionary Algorithm
<b>SUS</b>	Stochastic Universal Sampling
<b>TE</b>	Tennessee Eastman
<b>TOC</b>	Theory of Constraints
<b>TWP</b>	Twice Penalty
<b>UBP</b>	Unbalanced workload penalty
<b>UC</b>	Used Capacity
<b>VEGA</b>	Vector Evaluation Genetic Algorithm
<i>vel</i>	Velocity of particle in PSO
<b>WLAN</b>	Wireless Local Area Network

# **PEMBANGUNAN DAN APLIKASI MODEL PENGOPTIMUMAN BERLAPISAN BERDASARKAN PERKOMPUTERAN EVOLUSI**

## **ABSTRAK**

Thesis ini mempersembahkan satu model umum pengoptimuman berlapisan berdasarkan perkomputeran evolusi yang dapat menyelesaikan pelbagai masalah pengoptimuman berkaitan pelbagai keputusan, pelbagai resolusi, interaktif, hibrid dan pelbagai objektif telah dipersembahkan. Dalam model yang dicadangkan, tumpuan diberi kepada algoritma genetik (GA) dan pengoptimuman partikel (PSO) dalam mekanisme pencarian evolusi. Asas model ini adalah pembangunan struktur wakil kod berlapisan yang menggabungkan idea-idea seperti strategi asing dan menjajah, teori skima, konsep hirarki dari jaringan semantic, dan perwakilan berpandukan rangka. Berpandukan ilham yang diperoleh daripada pengevolusian pelbagai populasi, korelasi, strategi dan pembinaan berkaskad, dan cara-cara pengoptimuman, model umum pengoptimuman berlapisan berdasarkan perkomputeran evolusi (LECO) telah dibangunkan. Pembinaan LECO menyokong pergabungan teknik pengoptimuman yang berlainan. Kombinasi yang berlainan dari GA dan PSO dalam LECO dikaji untuk mendapatkan kombinasi yang paling sesuai untuk LECO. Satu siri kajian yang mengandungi masalah panduan dan masalah sebenar telah dijalankan untuk mengkaji keupayaan, kebolehubahsuaian, dan keberkesanan LECO untuk mengendalikan pelbagai jenis masalah pengoptimuman. Selain daripada data panduan dan data sebenar, data perkembangan penyelidikan juga diguna bagi mengkaji kemampuan LECO. Daripada keputusan yang diperolehi, GA-PSO LECO didapati berupaya untuk menyelesaikan masalah yang terdiri daripada pengoptimuman pelbagai keputusan, pengoptimuman parameter pelbagai resolusi dan pengoptimuman Pareto pelbagai objektif. Struktur LECO yang membenarkan analisis pada lapisan tertentu menyokong pengoptimuman interaktif. Secara keseluruhan, ciri-ciri struktur kod berlapisan yang dapat menunjukkan warisan pengetahuan, mengasingkan perwakilan dalam lapisan, membolehkan

pencarian yang seimbang, dan berupaya untuk membataskan kawasan pencarian, telah menjadikan LECO satu model pengoptimuman yang umum dan berkesan.

# **THE DEVELOPMENT AND APPLICATION OF EVOLUTIONARY COMPUTATION-BASED LAYERED ENCODING CASCADE OPTIMIZATION MODELS**

## **ABSTRACT**

In this thesis, the research on a generic evolutionary-based layered encoding cascade optimization (LECO) model that is able to solve different kinds of optimization problems on multi-decision, multi-resolution, interactive, hybridized and multi-objective is presented. In the proposed model, particular attention is given to genetic algorithm (GA) and particle swarm optimization (PSO) in the development of evolutionary-based search mechanism. The foundation of the proposed model is the development of layered encoding representation structure that integrates the ideas of divide and conquer strategy, schema theorem, and hierarchical concepts of semantic network and frame-based representation. Then, based on the insightful mechanisms of the multi-population evolution, cascade correlation, architecture, and strategy, as well as optimization methods, the LECO model is developed. The architecture of the LECO endorses hybridization of different optimization techniques. Different combinations of GA and PSO in LECO are studied to investigate the most appropriate combination of evolutionary mechanism for LECO. A series of empirical studies comprising benchmark and real-world problems is employed to assess the capability, flexibility, and effectiveness of LECO to handle different kinds of optimization problems as well as to be integrated with other heuristic techniques. Besides the datasets given in benchmark and real-world problems, hypothetical data is also included to investigate the performance of LECO towards larger scale problems. The experimental results demonstrate that GA-PSO LECO is able to optimize combinatorial multi-decision scheduling problem, multi-resolution parameter optimization as well as multi-objective Pareto optimization. In addition, the LECO structure that allows particular layer to be easily analyzed and evaluated promotes interactive optimization whereby human

intervention can be applied on layers for feature extraction. In all, the cascade layered encoding structure that is able to show inheritance of information, separating representation into layers, enhancing balance global-local search, and narrowing down the search space makes the LECO model a flexible, generic, and powerful tool for optimization problems.

# 1 INTRODUCTION

## 1.1 Preliminaries

In the past decades, much attention has been paid to the philosophy of nature in the study of computational research and optimization. The natural law of organism survival has brought to the increase of human curiosity towards the association of computer science with cognition, biology and psychology as a problem solving tool and optimization technique. Artificial Intelligence (AI) is a field of research that encompasses computational techniques with human intelligence on reasoning, learning and perception.

AI is defined as the study of mental faculties through the use of computational models (Charniak and McDermott, 1985). Luger and Stubblefield (1993) claimed that AI is the branch of computer science that is concerned with the automation of intelligent behavior whereas Winston (1992) described it as the study of the computations that make it possible to perceive, reason, and act. According to Padhy (2005), the development of symbolic representation in order to build formal structures capable of being solved by computers is one of the main directions of AI.

The fundamental hypothesis of AI on knowledge representation and reasoning inspired the construction of intelligent systems. Beside technical knowledge-engineering development of AI, the commercial reality of computers and improved technology makes AI a tremendous application to various fields such as factory automation, complex optimization, robotic control, engineering design and etc. Current trends of intelligent systems enthused by the development of AI include expert systems, fuzzy systems, neural networks, evolutionary algorithms, and swarm intelligence.

Among all intelligent systems, evolutionary algorithm and swarm intelligence are two of the major contributions of AI in the search of optimization. Optimization is an act of optimizing. It is described as the quantitative study of optima and methods of finding them (Wilde and Beightler, 1967). According to Goldberg (1989), the most important goal of optimization is improvement. Thus, even when the path to the ideal optimum is blocked or obscured, optimization theory often shows how existing conditions can be improved.

With the idea that a population of candidate solutions could evolve to solve optimization problem using operators analogous to the natural genetic variation and selection, evolutionary systems has been studied by a number of computer scientists since 1950s. In the 60's, evolutionary programming that was introduced by Lawrence J. Fogel and evolutionary strategies that was devised by Rechenberg and Schwefel are two well established evolutionary paradigms forming the backbone of evolutionary computation. Evolutionary programming represents candidate solutions as finite-state machines whereas evolutionary strategies process one individual along many generations incorporating concept of gene deletion and duplication (Fogel, 1962; Rechenberg, 1994; Schwefel, 1995; Zebulum *et al.*, 2001). Genetic algorithms (GAs) were conceived by John Holland and his fellow colleagues in the 1960s and 1970s. In 1975, theoretical framework for adaptation under genetic algorithm (GA) was presented (Holland, 1975; Padhy, 2005). In 1990's, evolutionary algorithm was used to evolve computer program by John Koza introducing genetic programming as the fourth stream of evolutionary computation (Koza, 1992).

The emergence of computational intelligence in optimization has led to the development of swarm intelligence based on collective behavior of decentralized and self-organized systems. The expression of swarm intelligence was introduced by Beni and Wang (1989) in the context of cellular robotic systems. Ant colony optimization (ACO) and particle swarm optimization (PSO) are two popular swarm intelligent systems. ACO, invented by Marco Dorigo was inspired by ant colonies behavior whereas PSO, devised by

Eberhart and Kennedy was stimulated by the social behavior of birds flocking and fish schooling (Dorigo and Di Caro, 1999; Eberhart and Kennedy, 1995). Both evolutionary computation and swarm intelligence are widely used meta-heuristic optimization algorithms.

This section discusses briefly the growth of AI and the rising of cognitive techniques such as GA and PSO in the area of optimization. The following section provides introduction to two major areas of AI: the representation and discovery of knowledge and hybridized intelligence. The issues of multi-decision, multi-objective and interactive optimization problems and the value of representation structure and hybridized intelligence that motivated this research are discussed followed by a description of the research scope, research objectives, and the research methodology. Finally, at the end of this chapter, the outline of this thesis is given.

## **1.2 Knowledge, Representation and Search**

According to Feigenbaum and McCorduck (1983), the art of bringing the principles and tools of AI research to bear on difficult application problems requires expert knowledge. Knowledge representation is concerned with the choice of symbolical knowledge expression in computational system that the operational model is supposed to manipulate and reason with (Wrobel, 1994). It is the part of AI that relates the way of thinking to the intelligent behavior which is crucial for both theoretical and practical successes.

A number of knowledge representation schemes were discussed in Turban *et al.* (2005), e.g. production rules, semantic networks, and frames. Production rules are represented in the form of condition-action pairs whereas semantic networks were composed by nodes and links focusing on the relationships between different concepts (Cox, 2001). Frame, on the other hand, is a data structure that provides structural representation of knowledge focusing on properties of certain objects.

In GAs, the matter of representation is raised in the issue of encoding. The way a solution of a problem is encoded into a chromosome is a key issue in GA. Encoding methods can be classified into binary encoding, real number encoding, integer or literal permutation encoding, and general data structure encoding (Gen and Cheng, 2000). As for PSO, representation arises in the identification of structure for particles in which particles can be represented by array in  $D$  dimensions. Dependent on the complexity of a problem, an appropriate encoding or data structure is required to capture the nature of the problem. Therefore, the way to transfer thinking to computational intelligence is a vital task to be accomplished in order to assure successful knowledge representation, transmission and acquisition in optimization algorithms.

The process of using computers to extract knowledge from data is called knowledge discovery. Knowledge discovery was known as machine learning in the early 1990s. According to Roiger and Geatz (2003), typical knowledge discovery methods include inductive learning, neural computing, and GAs. Inductive learning induce rules from existing cases with known results and store the rules in knowledge base for consultation whereas neural computing mimic the human brain, derive solutions to new problem using historical cases and store knowledge in the connection between artificial neurons. GA employs the principle of natural process of evolution to gradually find the best combination of knowledge from known cases. The basic operations of GA for discovering knowledge are reproduction, crossover and mutation.

When solving optimization problems, the choice of appropriate solution searching approaches is important to enhance the finding of the optimum or improved solution. Search methods can be categorized as analytical, exhaustive and heuristic. Analytical techniques use mathematical formula to derive optimal solution. Exhaustive search considers all the alternatives in the search without guide while heuristic is an informed and guided method that uses rules and problem-specific knowledge in the search process.

Stochastic search is a heuristic search methodology that incorporates probabilistic elements in problem solving and machine learning (Spall, 2003).

Evolutionary algorithms and swarm intelligence are heuristic search mechanisms that integrate the concept of survival-of-fittest and collective organism behavior respectively. These heuristic strategies are step by step procedures that repeat until a satisfactory solution is found. Such a search is very much cheaper and faster than exhaustive search, and is able to find solutions near to the best ones.

### **1.3 Hybridization**

All AI techniques have different spectrums of searching methodology, each with different advantages and limitations. The booming applications of AI nowadays have led to the fusion of AI techniques to solve complicated and combinatorial problems, overcoming limitations of individual techniques.

Each intelligent technique has individual computational properties that are particularly suited to some problems and not for others. As mentioned by Goonatilake and Khebbal (1995), neural networks are good at pattern recognition but not good at explaining how they reach their decisions. Conversely, fuzzy logic systems which are able to reason with imprecise information do well at explaining decisions, but they cannot automatically acquire the rules they use to make those decisions. On the other hand, GAs are good in optimization. However the property of randomize search do not always give the same result at the end of the run. As for particle swarm algorithm, it consists of directional search that learns from the best solution, updating the distance between the best and the others.

Many industrial applications and academic researches are solved using hybridization techniques. For instance, a hybrid intelligent system of neural network and expert system

was developed by Kobe Steel Plant in Japan to control blast furnaces for making iron and steel (Otsuka *et al.*, 1992). Karr (1991) used GA to design fuzzy membership functions whereas Yoon *et al.* (1994) elaborated on the investigation of the use of GAs for training neural networks. In Chambers (1995), a hybrid approach using neural networks, simulation, GAs and machine learning for real-time sequencing and scheduling problems was described.

#### **1.4 Problems and Motivation**

Optimization problems can be generally categorized as combinatorial multi-decision optimization, single and multi-objective optimization, multi-resolution optimization, human-machine interactive optimization and hybrid intelligence optimization. Each of them encompasses different intricacies that make the task of problem solving challenging.

In combinatorial multi-decision problems, optimization requires consideration on the impact of each alternative course of action because a decision made may have significant sequential effects over other decision. Similar to multi-objective problems, almost every real-world decision involves multiple and conflicting objectives in which optimization is required to find a particular decision that could fulfill two or more objective functions. The impact of one decision to other decision and the conflicting objectives of each decision make the process of optimization difficult. The situation becomes more complicated when the problem consists of multiple constraints, resolutions and variables.

In the 1990s, significant research interest has been paid to human-machine interaction problems in which human behavior and subjective evaluation are integrated into AI computation in optimization. In human-machine interactive optimization, human fatigue and the mechanism of transferring human evaluation into mathematical and logic computation are the main concerns. As for hybrid intelligence systems, it is important to

obtain advantages from the hybridized approach in which limitations of single intelligent technique is coped.

The dilemmas of multi-decision, multi-objective, multi-resolution, interactive and hybrid optimizations motivated the development of a generic optimization model that is capable of handling different aspects of each optimization case. The growth of AI and the inspiration to develop generic optimization model stimulate the research towards AI-based optimization tool.

In the design of AI-based optimization model, the choice of knowledge representation and search method are the key to the achievement of problem solving. Therefore, the way to represent knowledge and mechanism of solution search should also be generic in order to develop a generic optimization model that could deal with different problem conditions.

In knowledge representation, the design of representation structure has significant effect to the problem computation, analysis, visualization, search, and human-machine interface. Often in multi-decision and multi-objective optimization, when the dimension of search increases considerably, the process of problem evaluation will become tedious.

Moreover, for interactive and hybrid intelligence problems that involve intervention and interaction of the representation structure, the computation of the problem is complicated. Thus, an appropriate representation structure that could simplify the problem analysis, allows human-machine interaction, promotes hybrid integration, and facilitates understanding of the structure, is required to accomplish a generic optimization tool.

Besides representation structure, the mechanism of search is one of the main issues in optimization. To ensure the effectiveness of search, the concept of global exploration

and local exploitation need to be implied in balance. Exploration often refers to randomized global search whereas exploitation is more towards local search improvement. Too much exploration may slow down the search because available information (best available solution) may be improperly used (Dumitrescu *et al.*, 2000). On the contrary, too much exploitation may bring about premature convergence of the search. As different optimization techniques consist of different searching limitations, the aim to obtain balanced global and local searches motivates the incorporation of hybrid intelligence techniques in the design of generic optimization model.

In all, subjects that stimulate the ideas of developing a generic evolutionary-based optimization model can be highlighted as follows:

- Different intricacies of different optimization problems have complicated the design of a generic optimization model.
- Knowledge representation is the key to the successful interpretation of problem knowledge into expression that is capable to be reasoned by computation system, e.g. ease of analysis, visualization, data retrieval, etc. A generic knowledge representation structure that is generally applicable to different optimization problems is inspired.
- Search mechanism is vital in the design of an optimization model. With the recent popularity of evolutionary-based optimization algorithms, a generic evolutionary-based optimization model that could incorporate generic knowledge representation structure, capable of optimizing problem with different intricacies, is enthused.

## **1.5 Research Objectives**

In the previous section, issues of multi-decision, multi-objective, multi-resolution, interactive and hybrid optimization problems are discussed. The development of a generic optimization algorithm in solving multi-condition optimization problems requires

development in generic problem representation structure as well as generic searching mechanism. This research proposed a layered encoding representation structure and hybrid genetic algorithm and particle swarm approach to enhance knowledge representation and promote intelligent search of optimization problems. The developed optimization model has been applied to case studies that consist of different optimization difficulties. Where possible, comparisons are made with other conventional techniques to observe the applicability of the developed model. The main objectives of the research are as follows:

- to develop a generic optimization model to deal with multi-decision, multi-objective, multi-resolution, interactive, and hybrid intelligence optimization problems
- to introduce the layered encoding representation structure as a generic knowledge representation in problem solving
- to develop an evolutionary-based layered encoding cascade optimization model for solving various kind of optimization problems

GA and PSO are integrated into the layered encoding optimization model to promote hybrid heuristic search mechanism in which the performance of different combinations of GA and PSO in using layered encoding structure is assessed. Based on the three main objectives, the feasibility of layered encoding structure in allowing communication of solutions, incorporating multi-heuristic techniques, simplifying knowledge extraction, and endorsing balance global-local search is determined. In order to investigate the applicability and capability of the proposed evolutionary-based layered encoding optimization model, simulated, benchmark as well as real-world case studies are carried out in this research. Besides that, hypothetical data has also been generated for the assessment of the model's capability towards large scale problem. Note that the term "large scale" refers to bigger size problem with more complexities.

Different fields of case studies are taken to assess the performance of the proposed model in optimizing multi-objective, multi-resolution, interactive and multi-decision problems.

Multi-objective optimization is a process of optimizing two or more conflicting objectives, e.g. maximize the performance and minimize the cost of a manufacturing process (Sawaragi *et al.*, 1985). As for multi-decision optimization, it is related to multiple stage decision making for two or more sequential decisions that are interdependent (Bather, 2000). Interactive optimization, on the other hand, concerns human-machine interaction in which human integrates subjective evaluation into the computing process of optimization (Takagi, 1998) whereas multi-resolution optimization refers to optimization with multiple parameter precision in this research.

## **1.6 Research Scope and Approach**

The scope of this research is confined to the design and development of the layered encoding hybrid evolutionary model to solve general optimization problem of multi-decision, multi-objective, interactive and hybrid intelligence. The capabilities of the proposed optimization model to handle large-scale and high resolution problems are also investigated. Literature review is concentrated on the evolutionary search mechanisms of GA and PSO, as well as the evolutionary knowledge and problem representation structure. The study focuses on the potential of using layered encoding structure as the generic knowledge representation in the examined optimization problems. The evolutionary mechanism is focused on GA and PSO whereby these two algorithms are used to integrate random and directional search in solving optimization problems. Combinations of GA and PSO in different layers of the layered encoding structure are investigated. In addition, the support of layered encoding structure towards hybridization and integration of multi-heuristic techniques is also demonstrated.

This research starts with an objective to develop a generic optimization model. A thorough literature review is done on the various AI techniques which are widely applied by researchers to solve optimization problems. Particular attention is paid to knowledge

representation and search mechanism of GA and PSO in extracting knowledge and increasing the effectiveness of optimization algorithm.

Various knowledge representation schemes and structures are reviewed to understand the advantages and limitations in the development of optimization algorithms. The idea of conventional representation schemes and heuristic strategy of divide and conquer form the foundation of the proposed layered encoding structure. The capability of the layered encoding structure to deal with different problem conditions is investigated.

Hybridization of intelligent techniques is studied in an effort to comprehend the limitations that exist in the mechanism of optimization search. GA and PSO approaches are combined with the proposed layered encoding structure to allow integration of both randomized and directional searches. The proposed hybridized layered encoding model is implemented using MATLAB software.

The flexibilities of the proposed model to handle different optimization problems are assessed. As the research aims to develop a generic optimization model, different case studies are taken to observe the applicability of the proposed model. General optimization problems that are targeted to be addressed in this research are multi-decision, multi-objective, multi-resolution, and interactive. For hybrid intelligent problem, GA and PSO are hybridized in different combinations with layered encoding structure to study the most appropriate optimization model. In addition to the application on benchmark and real-world case studies, a hypothetical experiment is also included to test the performance of the proposed model towards large-scale problem.

In order to have a clearer illustration on the research flow, Figure 1.1 shows the overall flow of the research work. The flow allows the research to be carried out

systematically. The research is carried out in such a way to develop, to investigate, to analyze, and to compare.

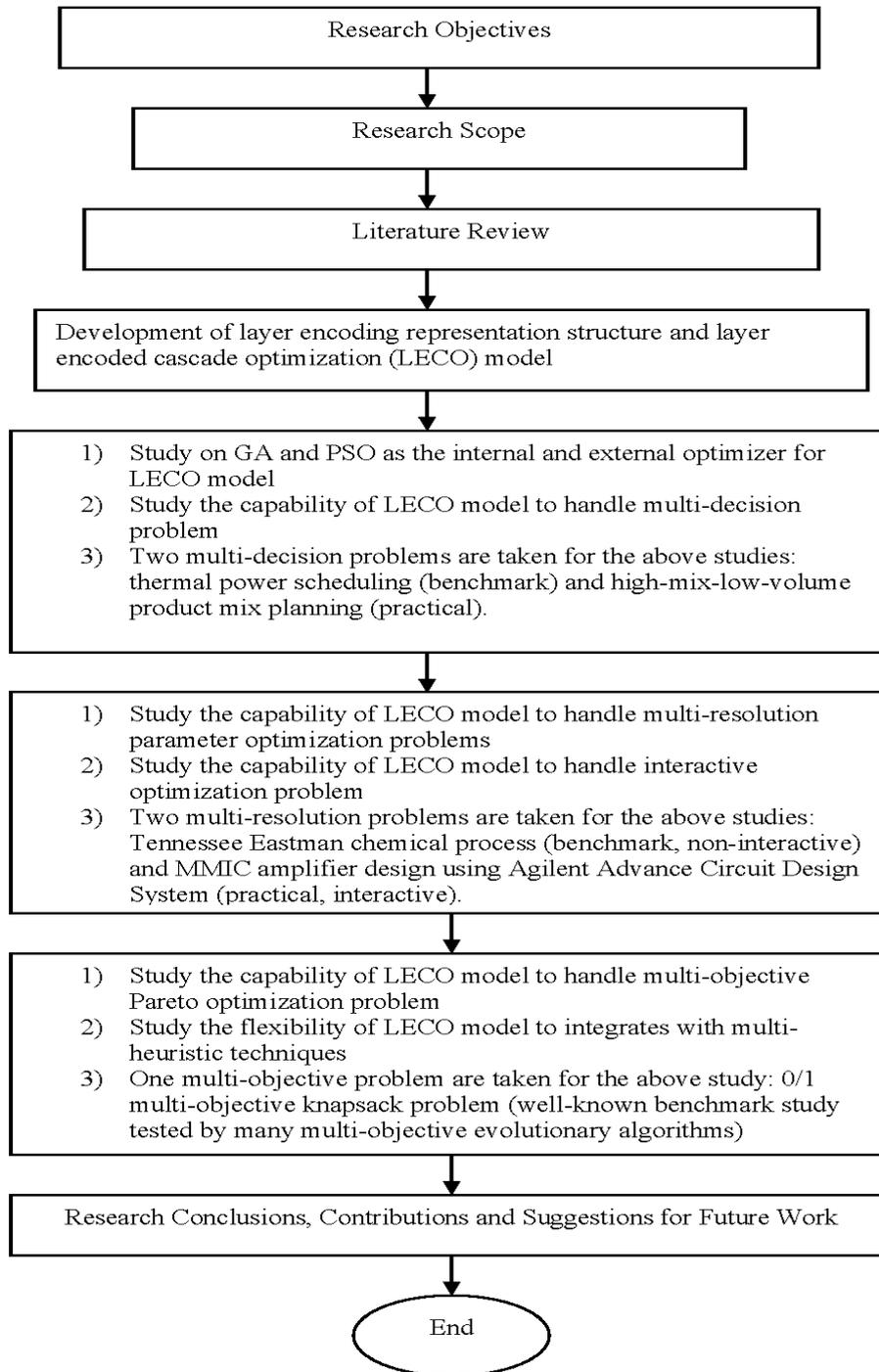


Figure 1.1 Flow chart of the overall work stages involved in the research

## 1.7 Thesis Outline

The layout of this thesis is as follows:

Following the introductory chapter, Chapter 2 begins with a literature review on GA and PSO as two widely used optimization techniques in AI. Then, the recent trends in evolutionary algorithms are discussed. Classical knowledge representation schemes, encoding structure and heuristic strategies for optimization search are reviewed.

Chapter 3 presents the development of the proposed layered encoding structure and optimization model. The underlying ideas of the formation of layered encoding structure using the concept of object-oriented knowledge representation, theory of schemata, and heuristic strategy of divide and conquer are described. The aims and advantages of the designed layered encoding structure are also discussed. Literature review on cascade strategy and multi-population evolutionary algorithms are given. An evolutionary computation-based layered encoding cascade optimization (LECO) model inspired by the cascade strategy and multi-population evolutionary algorithm is developed.

Chapter 4 describes the applicability of the proposed LECO model towards multi-decision optimization problems and presents the study of GA and PSO hybridization in different optimization layers of the proposed model. Multi-decision case studies are used to investigate the applicability and to find the best combination of GA and PSO for improving the performance of the proposed model. Initially, a side experiment is applied on a benchmark case study of thermal power generation scheduling to observe the most appropriate combination of GA and PSO for improving the layered encoding model. Then, the model is tested on a high-mix-low-volume (HMLV) real-world semiconductor production capacity planning to validate the applicability to multi-decision real-world problem. Based on the real-world case study, a set of hypothetical data is developed to evaluate the capability of the proposed model to solve larger scale problems.

Chapter 5 introduces GA-PSO layered encoding cascade optimization model to address the issues of multi-resolution and human-machine interactive problem. The difficulties to solve problem with multi-resolution and human-machine interaction are discussed. A side research is done on a simulated chemical process problem (Tennessee Eastman) to assess the ability of the layered encoding structure to cope with the representation of problem resolution. Then, a practical case study on circuit design that involves high resolution variables and requires human-machine interaction is carried out. The optimization on circuit design is implemented using the Agilent Advanced Design System (ADS). The proposed model is compared with the other built-in optimizers in ADS.

Chapter 6 reveals the ability of GA-PSO layered encoding model in handling multi-objective problem, finding Pareto trade-off and integrating other heuristic strategies. A test problem suite on knapsack problem is studied. Conventional multi-objective evolutionary approaches are reviewed. This chapter focuses on the search of Pareto optimum. The proposed model is integrated with strategies such as hill-climbing and non-dominated spread lengthens heuristic. The result is compared with other conventional multi-objective evolutionary approaches. The spread of the generated Pareto front and hypervolume evaluation are used to assess the performance of the proposed GA-PSO model.

Finally, Chapter 7 draws the conclusions and contributions of this research. Suggestions for future research are given at the end of this chapter.

Appendices contain the related data and results used in this thesis.

## **2 LITERATURE REVIEW**

### **2.1 Introduction**

Today, biological motivated optimization and search paradigm have accelerated the development of optimization techniques and problem-solving tools. Evolutionary algorithms and swarm optimizations are biological-based artificial intelligence techniques that are widely used in optimization problems. The emergence of these techniques has invigorated research on population-based solution search heuristics. Knowledge representation and solution search paradigm are two important features to be studied in the development of population-based optimization model.

This chapter introduces the basic mechanism of GAs and PSO. The applications of GA and PSO in optimization are discussed. Current trend of evolutionary algorithms and classical knowledge representation schemes are reviewed. In addition, several popular heuristic strategies for improving global and local optimization search are studied.

### **2.2 Genetic Algorithms**

The Darwinian concept of survival-of-the-fittest is the foundation of genetic-based mechanism such as evolutionary computation. GA is a genetic-based mechanism which mimics the evolutionary process of a population of individuals (Yuen and Chow, 2009). It is a numerical optimization algorithm and stochastic search method that was inspired by the Darwinian metaphor of natural biological evolution. GAs were first invented by John Holland in 1960s and were developed by him and his associates at the University of Michigan in 1960s and 1970s (Holland, 1975).

GA solves problem using the theory of natural selection and natural genetics. It is a robust technique that is able to offer advantages in solution methodologies and

optimization performance in which no auxiliary information is required. The GA searches a population of possible solutions stochastically using knowledge structure to represent the candidate solutions. The search is guided by the objective or fitness function and is evolved using genetic operators such as crossover and mutation.

An interesting survey of some GA practical works in search, optimization and machine learning has been conducted by Goldberg (1989). According to Goldberg, GAs are computationally simple yet powerful in the search for improvement. The ability to find optimal or near optimal solution without being limited by restrictive assumptions about the search space makes GAs a widespread optimization tool with applications in science, business and engineering (Fogel, 1995; Shin and Lee, 2002; Gen and Cheng, 2000; Zeng and Cheng, 2009).

### 2.2.1 GA Terminology and Basic Mechanism

As a biologically motivated intelligent technique, a number of GA's technical vocabularies are borrowed from the biological terms. For instance, GAs represent its candidate solutions as chromosomes. The chromosomes are made of a set of characters, called genes. Every gene corresponds to the inheritance of one or more characters. The location of a gene at a certain place of the chromosome is known as loci or string position and the possible values or states of gene are called allele. A combination of genes on each chromosome will dictate the structure of decision variable set within the solution.

Continuing the genetic analogy, GA works on genotype and phenotype, or in other words, coding space and solution space. Each genotype or a single chromosome represents a potential solution to a problem, which is known as its phenotype. In GA, genotype is interpreted as the coded string which is processed by the algorithm, while the decoded set of parameters represents the phenotype.

As in biological system, GA combines chromosomes or candidate solutions (parents) to produce offsprings (children) in each algorithmic generation. The chance for an individual or chromosome to survive in its present environment and to become parent that produce offsprings is very much dependent on its fitness. An evaluation function or fitness function is used to rate the fitness of chromosomes (solutions) in GA. The individuals that are selected for reproduction will undergo two important genetic operations using crossover and mutation to exchange the chromosome segments and maintain population diversity.

The basic mechanism of GA can be summarized as follows:

1. Initialize a population of chromosomes.
2. Chromosomes evaluation based on fitness function.
3. Generate new chromosomes by mating fittest chromosomes. New chromosomes are produced based on certain selection rules (e.g. roulette wheel selection and stochastic universal sampling) and genetic operators (crossover and mutation).
4. Delete less fit chromosomes of the population to make room for new members.
5. Evaluate the fitness of newly generated chromosome and insert the best individuals into the population.
6. Go to 3 until convergence or until a fixed number of generation is reached.

#### 2.2.1(a) Chromosome Encoding and Representation

Chromosome encoding or chromosome representation is a key issue in GA needed to describe each individual in the population of interest. The representation scheme determines how the problem is structured and also determines the genetic operators that are used. The most commonly used GA representation is binary string encoding in which binary code is used as the alleles of a gene (decision variable) in a string-based structure. Despite the popularity of binary encoding, it is known to have severe drawback due to the existence of Hamming cliffs (Gen and Cheng, 2000). Hamming cliffs are pairs of encoding that are having large Hamming distance in genotype space but belong to points of minimal

distance (Euclidean distance) in phenotype space (Ludvig *et al.*, 1997). For instance, the pair 0111111111 and 1000000000 are points in the phenotype space that neighboring each other but they are having maximum Hamming distance in the genotype space.

Besides binary-coded GAs, real-number and integer encoding are receiving interest in many real-world industrial applications. Investigation has been done on problem representation, and it has been shown that natural representation is more efficient and produces better solutions (Michalewicz, 1994). Michalewicz (1994) compared real-valued and binary GA, and showed that real-valued representation is more efficient in term of CPU time, and it offers higher precision with more consistent results across replication. An *et al.* (2009) mentioned that real-valued encoding can improve GA convergence rate as unnecessary decode and encode procedure can be avoided. On the other hand, Wright (1991) claimed that the use of real-valued genes offers a number of advantages in numerical function optimization in which less memory is required to convert chromosomes to phenotypes. Besides, there is also greater freedom to use different genetic operators to deal with real-valued genes.

In addition to the encoding of alleles value using binary, integer, real-number, and floating point, encoding method can also be classified as one dimensional encoding and multi-dimensional encoding. Though one dimensional encoding is widely used in many practices, multi-dimensional encoding structures are sometimes required to solve the complexity of many real-world problems. For instance, Huang *et al.* (2007) used multi-dimensional encoding in fast packet classification whereas Michalewicz (1996) and Michalewicz *et al.* (1991) used matrix representation to construct a chromosome and solved linear and nonlinear transportation problem by designing matrix-based genetic operators.

The data structure of solution representation in evolutionary algorithm by itself is a major research area. Despite the existence of linear string and matrix-based representation,

there are more complex data structures such as finite state machine and parse tree representations. Finite state machine are particularly useful in prediction or forecasting problems whereas parse tree representation are mostly used for mathematical expression in genetic programming. The details of finite state machine and parse tree representation can be obtained from Zebulum *et al.* (2001).

Besides the above mentioned representation method, specific encoding or data structure that is better suited to represent a solution may be used as the choice of encoding. The appropriate choice of encoding structure will have a major impact to the performance of evolutionary algorithms. According to Dumitrescu *et al.* (2000), the use of specific encoding permits easy transfer of problem domain expertise into algorithm and specialized representation relies on the following ideas:

- it uses specific data structures
- it considers new appropriate genetic operators if needed
- it ensures that genotypes encode feasible solutions
- it ensures that the search operators preserve feasibility

#### 2.2.1(b) Fitness Function

The fitness function is a performance measure or evaluation of chromosomes in the problem domain. According to Dumitrescu *et al.* (2000), chromosome evaluation is necessary to control the progress of GA search and the fitness function depends on the user's ability to encode the problem. Fitness function may be a cost function, loss function, penalty function or objective function.

In the case of a minimization problem, the associated fitness or objective function of the fittest individual will have the lowest numerical value. The fitness function is usually used to convert the objective function value into a measure of relative performance fitness (De Jong, 1975), as shown in equation 2.1.

$$F(x) = g(f(x)) \quad (2.1)$$

where  $f$  is the objective function,  $g$  is a function used to convert the value of the objective function to a non-negative number, and  $F$  is the relative fitness resulted from the function of  $g$  and  $f$ . However, in certain cases, the objective function and the fitness function is synonymous.

In general, fitness measure of a chromosome is independent of the fitness of the other individuals in the population. However, it is possible to have an implicit fitness function whose value depends on all individuals in the population (Dumitrescu *et al.*, 2000). For instance, an intrinsic adaptation (Packard, 1988) that ensures the co-evolution of the individuals (Kaufman and Johnsen, 1991) can be adopted.

#### 2.2.1(c) Penalty Function

Coding space and solution space are two work spaces of GA, playing important role in fitness evaluation and selection. One key issue associated with the mapping of these spaces is the issue of infeasible solutions in which the solution decoded from chromosome lies outside the feasible region of a given problem (Gen and Cheng, 2000). The issue of infeasible solution is often yielded from the manipulation of genetic operator in constrained and combinatorial optimization problems.

In GA, penalty function is the most common technique used to handle chromosome infeasibility. The penalty function transforms a constrained problem into unconstrained problem by adding a penalty term into the fitness function to penalize solution that violates the constraints. Penalty term can be integrated into evaluation function in two ways: addition form and multiplication form, as shown in equation (2.2) and (2.3) respectively.

$$\text{Addition form: } eval(x) = f(x) + p(x) \quad (2.2)$$

$$\text{Multiplication form: } eval(x) = f(x)p(x) \quad (2.3)$$

where  $x$  represents a chromosome,  $f(x)$  represents the fitness function,  $p(x)$  represents the penalty term and  $eval(x)$  represents the overall evaluation function.

Penalty method is different from the rejection method that discards all infeasible chromosomes. It is used to keep a certain number of infeasible solutions to enforce the genetic search towards the optimal solution from both sides of feasible and infeasible regions. As some infeasible solutions may provide useful information, penalty function does not simply reject all the infeasible solutions in each generation.

#### 2.2.1(d) Selection Function and Genetic Operators

In GA, the selection function is the process of deciding the reproduction chances of a particular individual (Davis, 1991). A probabilistic selection is performed based upon an individual's fitness such that the better individuals have an increased chance of being selected to be reproduced into the next generation. According to Gen and Cheng (2000), selection provides the driving force in GA, in which too much force terminate a genetic search prematurely whereas too little force will eventually slow down the evolutionary progress. There are several selection techniques, two of the most commonly used techniques are roulette wheel selection (RWS) and stochastic universal sampling selection (SUS).

Roulette wheel selection operates on the concept that the proportionate fitness of each chromosome should be reflected in that chromosome's incidence in the mating pool (Yii, 2001). Thus, each chromosome in the population has a probability of selection for crossover based on its relative fitness. This technique provides the greatest probability of selection to the most fit members of the population.

The expected occurrence ( $e$ ) of a chromosome ( $i$ ) in the mating pool is measured by dividing the fitness of a chromosome ( $f$ ) with the sum of the fitness of all chromosomes in a

population ( $n$ ). Equation (2.4) and (2.5), respectively show the calculation of probability of selection,  $pselect_i$ , and the formulation of expected occurrence in mating pool,  $e_i$ . The expected occurrence of a chromosome can affect the performance and diversity of a genetic search.

$$pselect_i = \frac{f_i}{\sum_{k=1}^n f_k} \quad (2.4)$$

$$e_i = pselect_i \times n \quad (2.5)$$

As the process of selection can be analogous to the spinning of a roulette wheel, this selection technique is called the roulette wheel selection. Based on the relative fitness of each chromosome, the amount of space on the wheel allocated by each chromosome reflects its probability to be selected for mating. The fitter the chromosome, the more the area on the roulette wheel that is occupied by that chromosome. The pointer in a spin of the wheel decides the selection of an individual to undergo reproduction.

Stochastic universal sampling (SUS) is a single-phase sampling algorithm with minimal spread as there is a limit for the number of times selecting a particular chromosome. Different from RWS, SUS uses  $N$  equally spaced pointers instead of single selection pointer. In SUS,  $N$  is the number of selection required. The underlying idea of this approach is to maintain the expected number of copies of each chromosome so that these chromosomes are available for mating in the next generation.

Besides selection function, the genetic operators in GA plays important role in chromosome reproduction. They provide a basic search mechanism to create new solutions based on existing solutions in the population. Two basic types of operators are crossover and mutation.

Crossover involves recombination of two preferentially chosen strings by exchanging the segments of the strings (Padhy, 2005). Du *et al.* (2009) mentioned that crossover is a key operator to ameliorate a population. Crossover produces new individuals that have some feature of both parents in term of genetic material. It is a method whereby information for differing solutions can be combined to enhance exploration of new areas of the search space. According to Morad (1997), the generated off-spring strings from the crossover process might have either higher or lower fitness values than the parent strings, but the selection process will ensure that higher fitness strings have higher probability of reproduction in the future generation.

In general, crossovers occur on the pairs of individuals in a population under a certain probability called crossover probability. The most commonly used crossover methods are single-point crossover and multiple-point crossover. Table 2.1 shows the operation for single-point crossover and multiple-point crossover for binary string chromosomes.

Table 2.1 Crossover operation in binary string chromosomes

Type of Crossover	Single-point	Multiple-point
Before Crossover	0011   011010 (parent 1)	001   101   1010 (parent 1)
	1100   010101 (parent 2)	111   011   0101 (parent 2)
After Crossover	0011   010101 (child 1)	001   011   1010 (child 1)
	1100   011010 (child 2)	111   101   0101 (child 2)

In addition to crossover operation, mutation is another genetic operator that is mainly used to maintain genetic diversity in the GA search. Du *et al.* (2009) claimed that mutation is related to the efficiency of GA. Mutation is a random process that alters one individual to produce a single new solution. The process can be implemented by

incrementing or decrementing a locus, or perhaps replacing it with a randomly generated number. By randomly altering values on genomes, this operation helps to ensure a more complete coverage of the search-space. As mentioned in Goldberg (1989), mutation acts as a safety net to recuperate good genetic material that may be lost through the evolutionary process of selection and crossover. In GA, mutation is normally applied to the population with low probability. Table 2.2 shows the mutation operation for binary string chromosomes.

Table 2.2 Mutation operation in binary string chromosomes

Before Mutation	1100101   1   001110
After Mutation	1100101   0   001110

### 2.2.2 Theory of Schemata

GAs are highly nonlinear search algorithms that is difficult to predict its behavior when varying its parameters (Zebulum *et al.*, 2001). Therefore, the attempt to draw a precise mathematical model of GA has motivated the introduction of the schema theory by John Holland (Holland, 1975). The schema theory provides information to guide a directed search for improvement.

A schema can be viewed as a template that provides particular pattern at certain string position that matches a number of chromosomes with similarities. The template consists of similar pattern of alleles at certain loci by neglecting some allele values. Considering general binary strings with binary alphabet ( $K=2$ ), schema appends a “don’t care” symbol besides binary alphabets of 0 and 1. The “don’t care” symbol is usually represented by an asterisk, “\*”, as shown in Figure 2.1.

In Figure 2.1, an example of schema for chromosomes with eight loci ( $L=8$ ) is given. The schema samples all chromosomes having ‘1’ at the fourth locus and ‘0’ at the sixth locus.