ENHANCED PARTICLE SWARM OPTIMIZATION ALGORITHMS WITH ROBUST LEARNING STRATEGY FOR GLOBAL OPTIMIZATION

LIM WEI HONG

UNIVERSITI SAINS MALAYSIA
2014

ENHANCED PARTICLE SWARM OPTIMIZATION ALGORITHMS WITH ROBUST LEARNING STRATEGY FOR GLOBAL OPTIMIZATION

by

LIM WEI HONG

Thesis submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy

September 2014

ACKNOWLEDGEMENT

This thesis is dedicated to everyone in the field of computational intelligence who embarks the journey of expanding the collection of knowledge and transcendent passion for computational intelligence.

It is an honor for me to thank those who made this thesis possible. First, I am grateful to Associate Professor Dr. Nor Ashidi Mat Isa, my thesis advisor and project supervisor, for seeing the promise of this thesis and achieving research conducted under his watchful eyes. His guidance and patience throughout the tumultuous time of conducting scientific investigations related to this project are much appreciated. Besides, his invaluable support and insightful suggestions, not to mention all the hardwork and extra time poured in has resulted in the completion of this project.

Further thanks go to the Universiti Sains Malaysia (USM) Postgraduate Fellowship Scheme and Postgraduate Research Grant Scheme (PRGS) entitled “Development of PSO Algorithm with Multi-Learning Frameworks for Application in Image Segmentation”, which provided financial support for my study.

I would like to take the opportunity to thank those people who spent their time and share their knowledge for helping me to improve my research work with the best results, especially the current and former members of Imaging and Intelligent System Research Team (ISRT): Eisha, Jing Rui, Naim, Sazali, Suheir, Jason, Manjur, Rawia, Helmi, Kenny, Tatt Hee, Khang Siang, Shue Siong, Chen Hee, and Sheng Hong. My special thank reached out to Abdul Latiff Abdul Tawab, Ahmand Ahzam Latib, Nor Azhar Zabidin, and Amir Hamid for their technical supports.

In additions, applauds and appreciations are dedicated to my friends: Li Lian, Eleanor, Qaspiel, Earn Tzeh, Yeong Chin, Wee Chuen, Pei Nian, Nick, Susan, Jing Huey, Wei Zeng, Wooi Keong, Mae Yin, Man Jia, and Joyc for their supports and friendships. Their kindness, generousness, and help made an easy life for me in the journey of pursuing my PhD study.

Specially, I want to appreciate the unconditional support from my family, my parents, my brothers and my sister. Their appreciation cannot be expressed in words. Without their unconditional love, encouragement, understanding, and faith, all the things I have are impossible.

Lastly, I also place on record, my sense of gratitude to some oversea researches: Dr. Changhe Li, Dr. Mengqi Hu, Dr. Marco A. Montes de Oca, Dr. Joaquin Derrac Rus, Prof. P. N. Suganthan, and Dr. Abdul Ghani Abro, who directly and indirectly have lent their helping hand in this research.

TABLE OF CONTENTS

CHAPTER 5 – PARTICLE SWARM OPTIMIZATION WITH ADAPTIVE TIME-VARYING TOPOLOGY CONNECTIVITY

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACL-PSO	PSO with Aging Leader and Challengers
ACO	Ant Colony Optimization
AFPSO	Adaptive Fuzzy PSO
AI	Artificial Intelligence
ANN	Artificial Neural Network
APSO	Adaptive PSO
APTS	Adaptive Population Tuning Scheme
APV	Adjusted *p*-value
ATA	Adaptive Task Allocation
ATLPSO-ELS	Adaptive Two-Layer Particle Swarm Optimization with Elitist Learning Strategy
ATVTC	Adaptive Time-Varying Topology Connectivity
BBO	Biogeography-based Optimization
BPSO	Basic PSO
CES	Classical Evolution Strategy
CI	Computational Intelligence
CLPSO	Comprehensive Learning PSO
CMAES	Covariance Matrix Adaptation Evolution Strategy
CRO	Chemical Reaction Optimization
DE	Differential Evolution
DMS-PSO	Dynamic Multi-Swarm PSO
DNLPSO	Dynamic Neighborhood Learning-based PSO
DNSCLPSO	Diversity Enhanced CLPSO with Neighborhood Search
DNSPSO	Diversity Enhanced PSO with Neighborhood Search
DTA	Dimension-level Task Allocation
EA	Evolutionary Algorithm
EC	Evolutionary Computation
EKE	Elitist-based Knowledge Extraction
ELPSO	Example-based PSO
ELS	Elitist-based Learning Strategy
ENT	Expanding Neighborhood Topology
EO	Extremal Optimization
EP	Evolutionary Programming
EPUS	Efficient Population Utilization Strategy
ES	Evolution Strategy

ESE	Evolutionary State Estimation
FA	Factor Analysis
FAPSO	Fuzzy Adaptive PSO
FE	Fitness Evaluation
FIPS	Fully-Informed PSO
FlexiPSO	Flexible PSO
FLPSO-QIW	Feedback Learning PSO with Quadratic Inertia Weight
FM	Frequency-Modulated
FPSO	Frankenstein PSO
FWER	Family Wise Error Rate
G3PCX	Generalized Generation Gap Module with Generic Parent-Centric Crossover Operator
GA	Genetic Algorithm
GEP	Gene Expression Programming
GP	Genetic Programming
GSO	Group Search Optimization
HPSO-TVAC	Hierarchical PSO with Time-Varying Acceleration Coefficient
IMM	Intelligent Move Mechanism
IPSO	Improved PSO
ITA	Individual-level Task Allocation
MC	Memetic Computing
MO	Multiple Objectives
MoPSO	Median-Oriented PSO
MPSO-TVAC	Mutation PSO with Time-Varying Acceleration Coefficient
MS	Metaheuristic Search
NS	Neighborhood Search
OA	Orthogonal Array
OCABC	Orthogonal Learning-based Artificial Bee Colony
ODEPSO	Extrapolated PSO based on tOrthogonal Design
ODPSO	PSO based on Orthogonal Design
OED	Orthogonal Experiment Design
OEDLS	OED-based Learning Strategy
OLPSO	Orthogonal Learning PSO
OPSO	Orthogonal PSO
OTLBO	Orthogonal Teaching Learning based Optimization
OT-PSO	Orthogonal Test-based PSO
OXBBO	Biogeography-based Optimization with Orthogonal Crossover Operator

OXDE	Differential Evolution with Orthogonal Crossover Operator
PAE-QPSO	Phase Angle-Encoded Quantum PSO
PFSD	Population’s Fitness Spatial Diversity
PS	Producer-Scrounger
PSD	Population Spatial Diversity
PSO	Particle Swarm Optimization
PSO-ATVTC	Adaptive Time-Varying Topology Connectivity
PSODDS	PSO with Distance-based Dimension Selection
PSO-DLTA	Particle Swarm Optimization with Dual Level Task Allocation
PSO-MAM	PSO with Multiple Adaptive Method
RCBBO	Real-Coded Biogeography-based Optimization
RCCRO	Real-Coded Chemical Reaction Optimization
RPPSO	Random Position PSO
SA	Simulation Annealing
SALPSO	Self-Adaptive Learning-based PSO
SI	Swarm Intelligence
SLPSO	Self-Learning PSO
SO	Single Objective
SPLS	Stochastic Perturbation-based Learning Strategy
TCM	Topology Connectivity Modification
TLBO	Teaching and Learning Based Optimization
TPLPSO	Teaching and Peer-Learning Particle Swarm Optimization
TS	Tabu Search
TVAC	Time-Varying Acceleration Coefficient
UPSO	Unified PSO

ALGORITMA PENGOPTIMUMAN KAWANAN ZARAH DIPERTINGKATKAN DENGAN STRATEGI PEMBELAJARAN TEGUH UNTUK PENGOPTIMUMAN GLOBAL

ABSTRAK

Pengoptimuman Kawanan Zarah (PSO) merupakan satu algoritma pencarian metaheuristik (MS) yang diinspirasi oleh interaksi sosial kumpulan burung atau kawanan ikan semasa pencarian sumber makanan. Walaupun PSO asal adalah satu teknik pengoptimuman yang berkesan bagi menyelesai masalah pengoptimuman global, algoritma ini mengalami beberapa kelemahan dalam penyelesaian masalah yang berdimensi tinggi dan kompleks, seperti kadar penumpuan yang lambat, kecenderungan yang tinggi untuk terperangkap dalam optima setempat dan kesulitan dalam penyeimbangan penjelajahan/penyusupan. Untuk mengatasi kelemahan-kelemahan tersebut, penyelidikan ini telah mencadangkan empat variasi PSO yang dipertingkatkan, iaitu, PSO dengan Pengajaran and Pembelajaran Sebaya (TPLPSO), PSO Dua Lapis Adaptif dengan Strategi Pembelajaran Elit (ATLPSO-ELS), PSO dengan Sambungan Topologi Melalui Perubahan Masa Adaptif (PSO-ATVTC) dan PSO dengan Peruntukan Tugas Secara Dua Peringkat (PSO-DLTA). Satu fasa pembelajaran alternatif telah dicadangkan dalam TPLPSO dengan menawarkan arah pencarian baharu kepada zarah-zarah yang gagal untuk meningkatkan kecergasannya dalam fasa pembelajaran sebelumnya. Dua mekanisme penyesuaian untuk peruntukan tugas pula telah dicadangkan dalam ATLPSO-ELS bagi meningkatkan keupayaan algoritma dalam penyeimbangan penjelajahan/penyusupan semasa proses pengoptimuman. Sebagai satu variasi PSO yang dilengkapi dengan pelbagai strategi pembelajaran, PSO-ATVTC mempunyai satu mekanisme yang berkesan dan cekap bagi menyesuaikan kekuatan penjelajahan/penyusupan bagi zarah-zarah yang berbeza dengan memanipulasikan struktur kejiranan mereka secara sistematik. Berbeza dengan kebanyakan variasi-variasi PSO yang sedia ada, PSO-DLTA mempunyai kemampuan untuk melaksanakan peruntukan tugas secara peringkat dimensi.

Secara khususnya, modul peruntukan tugas peringkat dimensi (DTA) yang dicadangkan dalam PSO-DLTA berkeupayaan untuk memperuntukkan tugas-tugas pencarian yang berbeza kepada komponen dimensi zarah yang berlainan berdasarkan ciri-ciri jarak yang unik di antara sesuatu zarah dan zarah global terbaik dalam setiap dimensi. Prestasi keseluruhan bagi keempat-empat variansi PSO yang dicadangkan telah dibandingkan dengan variasi-variasi PSO dan algoritma-algoritma MS yang sedia ada. 30 fungsi penanda aras yang mempunyai ciri-ciri berbeza dan tiga masalah reka bentuk kejuruteraan dalam dunia sebenar telah digunakan. Keputusan eksperimen yang dicapai oleh setiap variasi PSO yang dicadangkan juga dinilai secara menyeluruh dan disahkan melalui analisis statistik bukan parametrik. Berdasarkan keputusan eksperimen, TPLPSO mempunyai kerumitan pengiraan yang paling rendah dan algoritma ini menunjukkan kejituan pencarian, kepercayaan pencarian dan kecekapan pencarian yang baik dalam penyelesaian fungsi penanda aras yang mudah. ATLPSO-ELS mencapai kemajuan prestasi yang ketara, dari segi kejituan pencarian, kepercayaan pencarian dan kecekapan pencarian, dalam penyelesaian fungsi penanda aras yang lebih mencabar, namun dengan peningkatan kerumitan pengiraan. Sementara itu, PSO-ATVTC dan PSO-DLTA berjaya menyelesai fungsi-fungsi penanda aras yang mempunyai ciri-ciri berbeza dengan kejituan pencarian, kepercayaan pencarian dan kecekapan pencarian yang memuaskan, tanpa menjejaskan kerumitan rangka kerja algoritma. Antara kempat-empat variansi PSO yang dicadangkan, PSO-ATVTC telah dibuktikan sebagai variasi yang berprestasi terbaik, memandangkan algoritma ini menghasilkan kemajuan prestasi yang paling baik, dengan kerumitan pengiraan yang kedua terendah.

ENHANCED PARTICLE SWARM OPTIMIZATION ALGORITHMS WITH ROBUST LEARNING STRATEGY FOR GLOBAL OPTIMIZATION

ABSTRACT

Particle Swarm Optimization (PSO) is a metaheuristic search (MS) algorithm inspired by the social interactions of bird flocking or fish schooling in searching for food sources. Although the original PSO is an effective optimization technique to solve the global optimization problem, this algorithm suffers with several drawbacks in solving the high dimensional and complex problems, such as slow convergence rate, high tendency to be trapped into the local optima, and difficulty in balancing the exploration/exploitation. To mitigate these drawbacks, this research has proposed four enhanced PSO variants, namely, Teaching and Peer-Learning PSO (TPLPSO), Adaptive Two-Layer PSO with Elitist Learning Strategy (ATLPSO-ELS), PSO with Adaptive Time-Varying Topology Connectivity (PSO-ATVTC), and PSO with Dual-Level Task Allocation (PSO-DLTA). An alternative learning phase is proposed into the TPLPSO to offer the new search direction to the particles which fail to improve its fitness during the previous learning phase. Two adaptive mechanisms of task allocation are proposed into the ATLPSO-ELS to enhance the algorithm's capability in balancing the exploration/exploitation during the optimization process. Being a PSO variant equipped with multiple learning strategies, PSO-ATVTC has an effective and efficient mechanism to adaptively adjust the exploration and exploitation strengths of different particles, by systematically manipulating their respective neighborhood structures. Unlike most existing PSO variants, PSO-DLTA has the capability of performing the dimension-level task allocation. Specifically, the dimension-level task allocation (DTA) module proposed into the PSO-DLTA is able to assign different search tasks to different dimensional components of a particle, based on the unique distance characteristics between the particle and the global best particle in each dimension. The overall performances of the four proposed PSO variants have been compared with a number of existing PSO variants and other MS algorithms on 30

benchmark functions with different characteristics and three real-world engineering design problems. The experimental results obtained by each proposed PSO variant are also thoroughly evaluated and verified via the non-parametric statistical analyses. Based on the experiment results, TPLPSO is observed to have the lowest computational complexity and this algorithm exhibits excellent search accuracy, search reliability, and search efficiency in solving simpler benchmark functions. ATPLSO-ELS achieves significant performance improvement, in terms of search accuracy, search reliability, and search efficiency, in solving more challenging benchmark functions, with the cost of increasing computational complexity. Meanwhile, PSO-ATVTC and PSO-DLTA successfully solve the benchmark functions with different characteristics with promising search accuracy, search reliability, and search efficiency, without severely compromising the complexities of algorithmic frameworks. Among the four proposed PSO variants, PSO-ATVTC is concluded as the best performing variant, considering that this algorithm yields the most significant performance improvement, by incurring the second lowest computational complexity.

CHAPTER 1

INTRODUCTION

1.1 Concept of Global Optimization

Global optimization is a branch of applied mathematics and numerical analysis that deals with the optimization of a function or a set of functions (Li, 2010, Liberti, 2008, van den Bergh, 2002). It is a process of finding the best solution of a given problem that would have either maximized or minimized the problem objective and to satisfy all criteria associated with the problem objective (Lam et al., 2012, Chetty and Adewumi, 2013, van den Bergh, 2002). This concept is widely used by humankind in solving various problems, ranging from the development of cutting-edge technologies to human's daily life. For instance, geneticists are interested in designing the optimal sequences of deoxyribonucleic acid (DNA) to achieve the maximum reliability of molecular computation (Shin et al., 2005, Zhang et al., 2007, Blum et al., 2008). Meanwhile, economists desire to minimize their prediction error for more accurate prediction of the stock market trends (Yu et al., 2009, Majhi et al., 2009, Singh and Borah, 2014).

From the mathematical perspective, the aim of global optimization is to determine the optimal (or best) solution *x* out of a set of solutions $\Re^D$, where $x = [x_1, x_2, ..., x_D]$ and $\Re^D$ denote a *D*-dimensional vector and the *D*-dimensional problem search space, respectively (Lam et al., 2012, Chetty and Adewumi, 2013, van den Bergh, 2002). The optimality of the solution vector *x* is assessed through the objective function *ObjV* of a given problem, where *ObjV* is used to characterize the landscape of search space $\Re^D$. The outcome of this assessment is scalar and it is represented by an objective function value *ObjV* (*x*). A global optimization can be subjected to *M* constrains, i.e. $C_1(x)$, $C_2(x)$, … , $C_M(x)$ to determine if the solution vector *x* is a feasible solution to the search space $\Re^D$. For an unconstrained minimization problem, the global optimum solution x^* is defined as (Chetty and Adewumi, 2013):

$$ObjV(x) = \{x^* \in \Re^D : ObjV(x^*) \leq ObjV(x) \text{ for all } x \in \Re^D\} \tag{1.1}$$

As shown in Equation (1.1), the global optimum solution x^* of a given minimization problem yields the lowest objective function value in the entire search space $\Re^D$. On the other hand, the global optimum solution x^* of an unconstrained maximization problem produces the highest objective function value in the entire search space and it is stated as (Chetty and Adewumi, 2013):

$$ObjV(x) = \{x^* \in \Re^D : ObjV(x^*) \geq ObjV(x) \text{ for all } x \in \Re^D\} \tag{1.2}$$

Global optimization is a fast growing and significant research field, considering that it plays important role in various practical application fields such as business, science, engineering, finance, and many other fields. Nevertheless, it has become a more challenging task, attributed to the escalating complexities of the problem search spaces. A wide variety of optimization techniques are developed to find the global optima of these challenging problems. In the following section, the global optimization algorithms that are used to solve the global optimization problems are presented. Without loss of generality, this thesis will focus on the global minimization problems in the following chapters. Specifically, these global minimization problems have single objective and without constraints in the search space $\Re^D$, except the constraint of the search domain.

1.2 Global Optimization Algorithm

Global optimization involves the searching of the best possible solution to a given problem within a reasonable time limit. There are numerous global optimization algorithms developed to deal with this task. One of the factors that determine the ability of a global optimization algorithm in finding the global optimum of a given problem is the complexity of the search space. For example, it is more likely for a global optimization algorithm to find the global optimum of a simple unimodal function than a hybrid composition benchmarks. In general, the global optimization algorithms which are used to tackle the global optimization

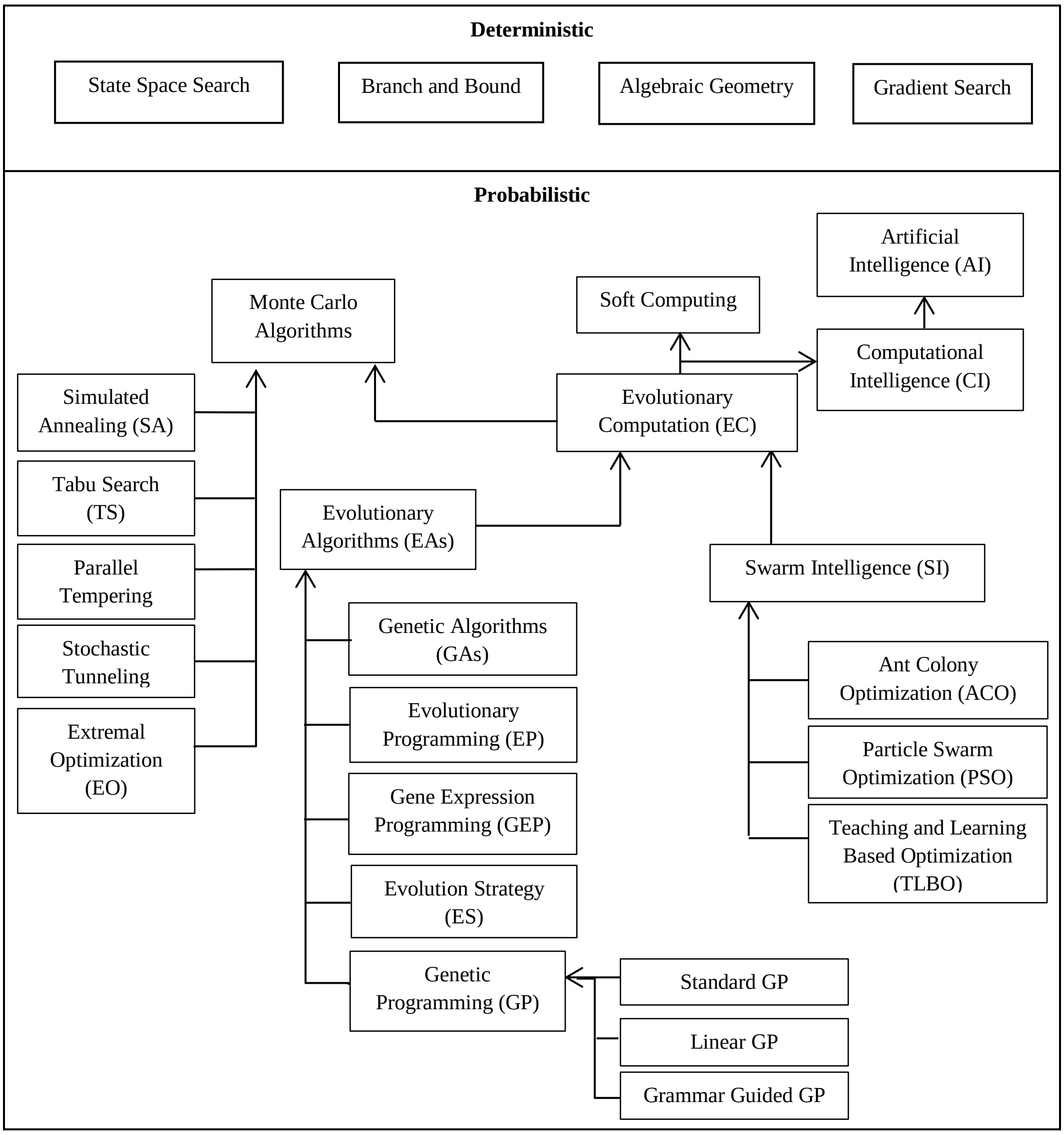

Figure 1.1: Categorization of global optimization algorithms (Li, 2010, Weise, 2008).

problems can be categorized into two basic classes, namely deterministic and probabilistic algorithms (Li, 2010, Chetty and Adewumi, 2013, Weise, 2008), as illustrated in Figure 1.1.

1.2.1 Deterministic Algorithm

As illustrated in Figure 1.1, deterministic algorithms include the state space search, branch and bound, algebraic geometry, gradient search, and others (Weise, 2008). These algorithms share a common characteristic, i.e. they employ the exact methods to solve the global optimization problems (Chetty and Adewumi, 2013, Li, 2010). These exact methods perform

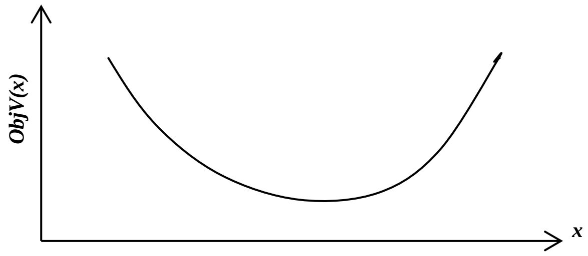

Figure 1.2: Fitness landscape with sufficient gradient information (Liberti, 2008, Li, 2010).

the exhaustive search of solution space to obtain the global optimum of a given problem. These exhaustive searches, however, are only feasible when there is sufficient gradient information of the objective function (Li, 2010, del Valle et al., 2008). For example, the fitness landscape of a unimodal function, as illustrated in Figure 1.2, consists of clear relation between the possible solutions and the objective function. This characteristic enables the deterministic algorithms to exhaustively explore and evaluate every possible solution in the search space of unimodal function, and therefore obtain the global optimum.

On the other hand, it is impractical to use the deterministic algorithms to find the global optimum when the objective function of a given problem is too difficult, or has insufficient or no gradient information for the exhaustive search. Generally, an objective function is considered difficult to solve if it is not differentiable, not continuous, or has excessive amount of local optima in the fitness landscape (Li, 2010). The fitness landscapes of some difficult objective functions are presented in Figure 1.3. For example, the fitness landscape in Figure 1.3(a) has too many local optima and the deterministic algorithms do not know the right direction during the search process. Meanwhile, the fitness landscape as shown in Figure 1.3(b) exhibits deceptiveness and it tends to mislead the deterministic algorithms away from the global optimum. Figures 1.3(c) and 1.3(d) show that the global optima of objective functions are located on the plateaus and the fitness functions do not provide any meaningful gradient information to the deterministic algorithms to guide the search.

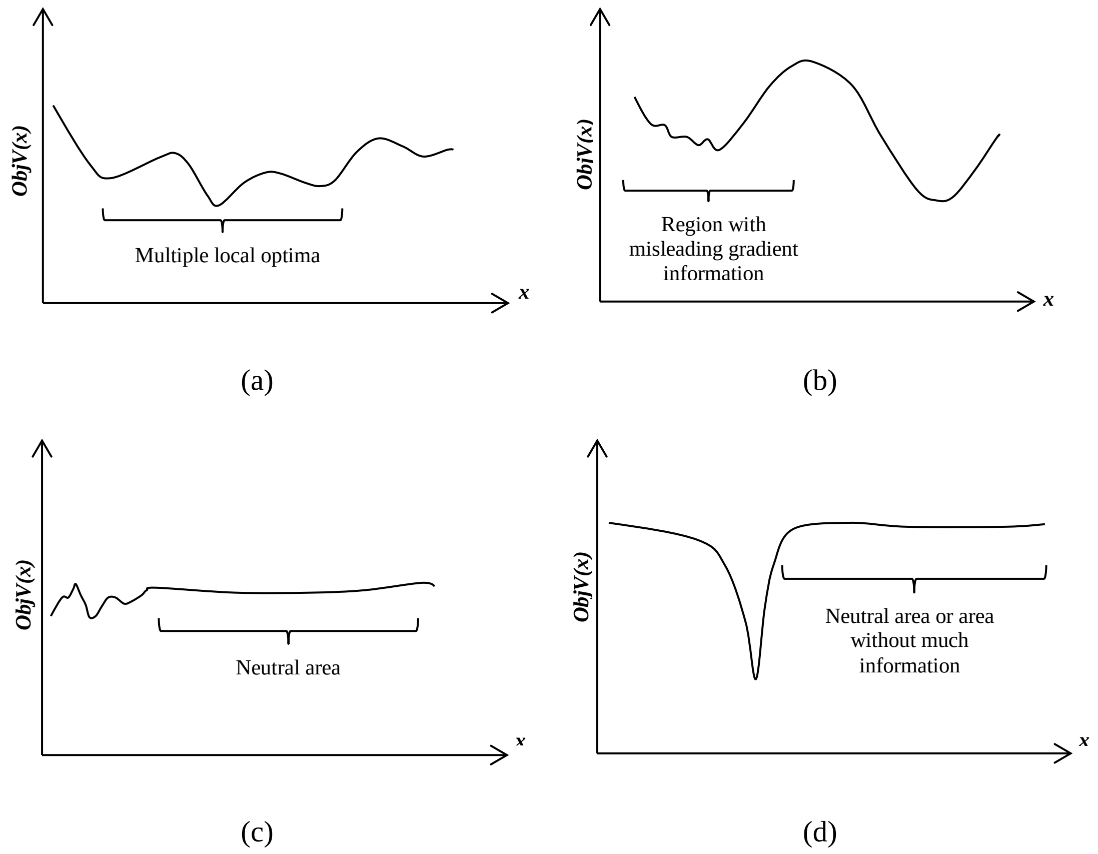

Figure 1.3: Different properties of difficult fitness landscapes: (a) multimodal, (b) deceptive, (c) neutral, and (d) needle-in-a-haystack (Blum et al., 2008, Weise, 2008).

1.2.2 Probabilistic Algorithm

As depicted in Figure 1.3, it can be observed that the objective functions with difficult fitness landscapes impose severe challenges to the deterministic algorithms and this inevitably leads to the poor optimization results. The inferior performance of these exhaustive approaches eventually bring the era of the stochastic-based probabilistic algorithms. Unlike the deterministic approach, the probabilistic algorithms are derivative-free and they tend to exhibit relatively resilient search performance in various types of optimization problems, including those with multimodal, deceptive, or noncontinuous fitness landscapes. Most of the probabilistic algorithms are Monte Carlo-based (Krauth, 1996), considering that these algorithms employ the randomization in determining the solutions of global optimization problems (Chetty and Adewumi, 2013).

Metaheuristic (Bianchi et al., 2009, Blum and Roli, 2003) is another important element that could be found in the probabilistic algorithms. Specifically, metaheuristic helps the probabilistic algorithms to decide which candidate solutions to be processed and how to generate the new candidate solutions based on the currently gathered information. This process is performed stochastically by employing the statistical information obtained from the samples in the search space or based on an abstract model inspired from a natural phenomenon or a physical process (Li, 2010, Weise, 2008). For instance, simulated annealing (SA) (Kirkpatrick et al., 1983) utilizes the Boltzmann probability factor of atom configurations of solidifying metal melts to determine which candidate solutions to be processed next. On the other hand, the extremal optimization (EO) (Boettcher and Percus, 1999) takes the inspiration from the metaphor of thermal equilibria in physics.

An important class of probabilistic Monte Carlo metaheuristic is the evolutionary computation (EC) (De Jong, 2006), which is also a class of soft computing as well as a part of the artificial intelligence, as illustrated in Figure 1.1. EC-based probabilistic algorithms rely on the concept of a population of individuals to solve a given problem, where each individual represents a candidate solution in the problem search space. The probabilistic search operators of EC algorithms are used to iteratively refine the multiple candidate solutions, in order to ensure these individuals evolve towards the increasingly promising solutions. Two of the most important members in EC class are evolutionary algorithm (EA) (Back, 1996) and swarm intelligence (SI) (Bonabeau et al., 1999).

The developments of the EA-based probabilistic algorithms are inspired by the natural evolution in the biology world (Back, 1996). The probabilistic search operators of EAs that are used to generate the new candidate solutions are mimicked from the nature evolution processes such as natural selection and survival of the fittest. Genetic algorithm (GA) (Goldberg and Holland, 1988, Weise, 2008) is a subclass of EA and this algorithm mimics the metaphor of natural biological evolution via the mechanisms of mutation, crossover, and selection. Evolutionary programming (EP) and evolutionary search (ES) are another two important members of EA (De Jong, 2006, Back, 1996). Both of these algorithms share many similarities in term of search mechanism, except that EP is not equipped with the recombination operator. Another difference that distinguishes these two EAs is the characteristic of their respective selection operators (Li, 2010). Specifically, EP employs a soft selection mechanism, known as the stochastic-based tournament selection, to offer the individuals with inferior solutions a probabilistic opportunity to survive in the next generation. On the other hand, ES uses the deterministic selection (Weise, 2008), i.e. a hard selection mechanism that inhibits the survival of worst individual in the next generation. Meanwhile, both of the genetic programming (GP) (Koza, 1992) and gene expression programming (GEP) (Ferreira, 2001, Ferreira, 2004) are used to evolve the computer programs. Unlike GP where each individuals are encoded as nonlinear entities of different sizes and shapes (parse trees), the individuals in GEP are first expressed as linear strings of fixed length (genome), and then translated as nonlinear entities of different sizes and shapes (simple diagram representation of expression trees) (Ferreira, 2001, Ferreira, 2004). GEP is more versatile than GP, considering that the former successfully creates the separate entities of genome (genetype) and expression tree (phenotype) (Ferreira, 2001, Ferreira, 2004).

SI is another important class of probabilistic Monte Carlo metaheuristic in EC. Generally, SI takes inspiration from the natural and artificial systems composed of population of simple agents that coordinate using decentralized control and self-organization (Bonabeau et al., 1999). Compared to the EA that primarily focuses on the competitive behavior in biological evolution, SI studies on the collective behaviors exhibited by the local interactions of the individuals with each other and with the environments, which could eventually lead to the emergence of intelligent global behavior (Bonabeau et al., 1999). One example of SI-based global optimization algorithm is the ant colony optimization (ACO) (Dorigo and Blum, 2005, Dorigo et al., 1996) that is inspired by the foraging behavior of ants. This algorithm is initially proposed to search for an optimal path in graph with a set of software agents called “artificial ant”. Particle swarm optimization (PSO) (Kennedy and Eberhart, 1995, Banks et al., 2007, del Valle et al., 2008) is another well-known member of SI and it is inspired by the collaborative behavior of a swarm of fishes or birds in searching for foods. Recently, a new form of SI, namely the Teaching and Learning Based Optimization (TLBO) (Rao et al., 2011, Rao et al., 2012) is proposed. Unlike the ACO and PSO that emulate the collective behaviors of insects and animals, the development of TLBO is motivated by the human teaching and learning paradigm in school. Besides these three algorithms, more inspiring SI-based algorithms have been proposed in the past decade to capitalize the benefits of decentralized and self-organizing behaviors of the SI systems in tackling various types of challenging optimization problems. Considering that this thesis focuses on developing the new PSO algorithms, the following section in this chapter will discuss the basic concept of PSO in detail.

1.3 Particle Swarm Optimization

As explained in the previous subsection, the development of PSO is motivated by the collective and collaborative behaviors of bird flocking and fish schooling in searching for foods (Kennedy and Eberhart, 1995, Eberhart and Shi, 2001, Banks et al., 2007, del Valle et al., 2008), as illustrated in Figure 1.4. This SI-based algorithm was first proposed by Kennedy and Eberhart in 1995. As a population-based probabilistic Monte Carlo metaheuristic, PSO employs a set of software agents called particles that fly through the multidimensional problem hyperspace with given velocity to simultaneously evaluate many points in the search space. Specifically, the position of each particle in the search space represents a potential solution of a given optimization problem. Meanwhile, the location of the food source where these particles are searching for is regarded as the global optimum of problem. Compared to most of the EC-based algorithm, the PSO particles have more effective memory capability, considering that these particles are able to remember their previous best positions (self-cognitive experience) as well as the neighborhood best position (social experience). These two experiences are the vital components in PSO learning strategy and they are used to adjust the flying direction of each PSO particle in the search space (Kennedy and Eberhart, 1995, Eberhart and Shi, 2001).

(a) (b)

Figure 1.4: The collective and collaborative behaviors of (a) bird flocking and (b) fish schooling in searching for foods.

During the search process, all particles have a degree of freedom or randomness in their movements. This characteristic allows each individual in the particle swarm to scatter around and move independently in the problem search space. Besides navigating through the problem search space independently and stochastically, these particles also interact with its neighborhood members via the information sharing mechanism. Specifically, the best performing particle in a particular neighborhood structure will announce its location in the search space to its neighborhood members via some simple rules. The social interaction exist between the particles in the problem search space enables the PSO population gradually moves towards the promising regions from different directions, thereby leads to the swarm convergence (Kennedy and Eberhart, 1995, Eberhart and Shi, 2001). Commonly, swarm convergence is attained when the PSO swarm is no longer able to find new solutions or the algorithm keeps searching on a small subset region of the search space (Li, 2010).

PSO has captured much attention in the research arena of computational intelligence since its inception, due to its effectiveness and simple implementation in solving optimization problems. For example, a quick browse on IEEE Xplore with a simple query "particle swarm optimization" returns more than 12,000 hits for papers published after year 2000. The current relevance of PSO can also be shown through the visibility of this topic at the Science Direct database. Figure 1.5 illustrates an important number of PSO-related research publications per year in Science Direct database. It demonstrates a growing trend despite the item related to PSO first appeared in 1995, implying that PSO is still a research subject of great interest. To further emphasize the importance of PSO in the research community, many scientists and engineers have capitalized this algorithm to solve many real-world engineering design problems because PSO has fast convergence speed and requires less parameter tunings (del Valle et al., 2008, Banks et al., 2007). Some of these engineering design problems involve power system design (del Valle et al., 2008, AlRashidi and El-Hawary, 2009, Neyestani et al., 2010, Wang et al., 2011, Wang et al., 2013a, Zhang et al., 2012), artificial neural network (ANN) training (Mirjalili et al., 2012, Yaghini et al., 2013), data clustering (Shih, 2006, Yang et al., 2009, Kiranyaz et al., 2010, Sun et al., 2012), data mining (Wang et al., 2007, Özbakır and Delice, 2011, Sarath and Ravi, 2013), parameter estimation and identification of systems (Liu et al., 2008, Modares et al., 2010, Sakthivel et al., 2010), and many other engineering problems (Huang et al., 2009, Lin et al., 2009, Sharma et al., 2009, Yan et al., 2013, Sun et al., 2011).

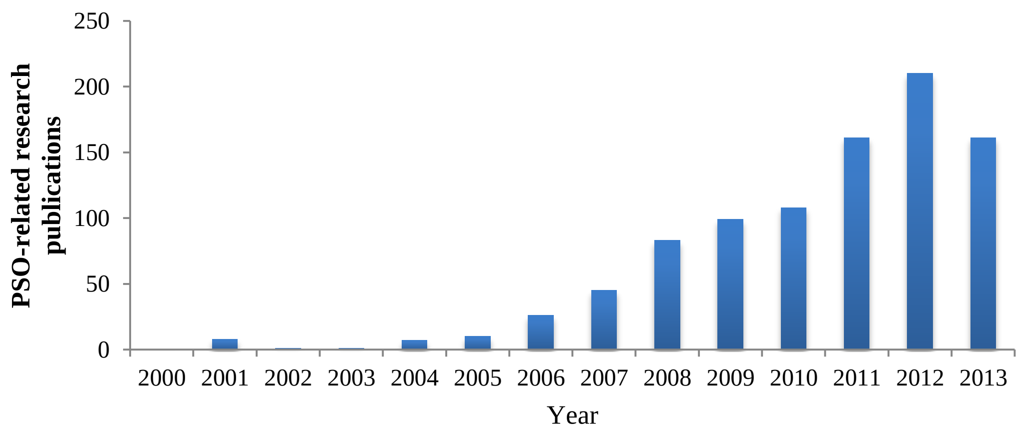

Figure 1.5: Number of research publications per year for PSO in Science Direct database.

1.4 Challenges of PSO in Global Optimization

Although PSO is a popular choice of optimization technique in solving the global optimization problems, earlier research reveals that this SI-based algorithm suffers with some drawbacks. These drawbacks could jeopardize the search performance of PSO and thus restrict the application of this algorithm in solving the real-world problems. This section aims to cover the challenges faced by PSO in global optimization and how these challenges affect the algorithm's optimization capability.

One of the main concerns on PSO is that this algorithm and most of its existing variants do not offer the alternative learning phase to the particles when the latter fails to improve the quality of their solution (i.e., fitness) during the optimization process. Each PSO particle updates its new solutions by referring to its self-cognitive experience and the social experience. Considering that some random movements are involved during the search process, there is a probabilistic opportunity for the particle to produce a new solution with less promising quality (i.e., fitness) as compared to its previous one. Under this scenario, the particle's convergence speed towards the global optimum solution might slow down, considering this particle is not on the right trajectory to locate the global optimum.

Another challenging issue of PSO is that, although the neighborhood best particle (social experience) is crucial in guiding the PSO swarm during the search process, the neighborhood best particle has the poorest learning strategy to update itself (Kiranyaz et al., 2010). For the neighborhood best particle, its personal and neighborhood best positions are same and this similarity inevitably nullifies the self-cognitive and social components of particle during the velocity update (Kiranyaz et al., 2010). As compared to other population members, the neighborhood best particle suffers with higher risk to stagnate at the local optimum or any arbitrary point in the search space because of the zero velocity produced by the nullified effect. Consequently, the poor optimization results are delivered (Clerc and Kennedy, 2002, van den Bergh, 2002, Ozcan and Mohan, 1999).

PSO also suffers with the intense conflict between exploration and exploitation searches (Shi and Eberhart, 1998). Specifically, exploration encourages the algorithm to wander around the entire search space to cover the unvisited regions, whereas exploitation emphasizes on the local refinement of the already found near-optimal solutions. Neither of these two contradict strategies should be overemphasized because excessive exploration tends to consume more computation resources, whereas excessive exploitation could lead the PSO towards the premature convergence. In general, the premature convergence is undesirable and it is identified when the PSO converges to a local optimum while there are other better locations existing in the fitness landscape than the currently searched area (Ozcan and Mohan, 1999, Clerc and Kennedy, 2002, van den Bergh and Engelbrecht, 2004, Liang et al., 2006, van den Bergh and Engelbrecht, 2006).

The universality and robustness of the PSO and most of its variants in tackling the diverse set of global optimization problems with different properties are also questionable. The inability of the PSO to best cope with all problems is attributed to the fact that different problems have differently shaped fitness landscape. The problem's difficulty is further compounded by the fact that in a certain benchmark, such as the composition function (Suganthan et al., 2005), the shape of the local fitness landscape in different subregions may be significantly different (Li, 2010, Li et al., 2012). To effectively solve these complex problems, different PSO particles should play different roles (i.e., perform different learning strategies) in different locations of fitness landscape and different search stages. However, most of the PSO variants that have been proposed so far use only one type of learning strategy and thus have limited choices of exploration/exploitation strengths to perform the search in different subregions of the search space (Wang et al., 2011).

Finally, it can also be observed that the original PSO and most of its variants tend to restrict the PSO particle to perform one type of learning strategy at the population level or the individual level. For population level, all particles in the population need to perform one type of learning strategy. Meanwhile, for the individual level, each particle can choose the desired learning strategy based on some decision making mechanisms. In these two approaches, the particle performs the same learning strategy in all dimensional components, without considering the particle's characteristics in each dimension of the search space.

According to the "*two step forward, one step back*" phenomenon as explained by van den Bergh and Engelbrecht (2004), different particles in PSO could have different characteristics in different dimension of the search space. These unique characteristics should be capitalized to assign the appropriate learning strategy to each dimensional component of particle.

1.5 Research Objectives

As discussed in the previous subsection, there are several main issues encountered by the original PSO and some of its existing variants, which tend to restrict their optimization capabilities. This thesis aims to alleviate the aforementioned issues by developing few enhanced PSO algorithms with robust learning strategies for global optimization problems. The objectives of this research work are presented as follows:

1. To devise an alternative learning phase to the particle as well as to introduce a unique learning strategy to the neighborhood best particle.
2. To develop two adaptive task allocation mechanisms to the PSO population for achieving better regulations of the exploration/exploitation searches of particle without significantly compromising the algorithm's convergence speed.
3. To propose an innovative mechanism that enables the particles to adaptively choose the appropriate learning strategies for the robust searching in various types of optimization problems.
4. To develop a dimension-level task allocation mechanism to the PSO for enabling each dimensional component of the PSO particle to select an appropriate learning strategy based on its characteristics in each dimension of the search space.

1.6 Research Scopes

The scope of this research focuses on the design and development of the enhanced PSO algorithms with robust learning strategies. Specifically, these enhanced PSO variants are confined to solve the single objective and unconstrained global minimization problems with static and non-noisy fitness landscapes.

In this thesis, all proposed PSOs are tested on a total of 30 benchmarks with different types of fitness landscapes to conclusively evaluate the algorithm's performance. Considering that each benchmark problem is specifically designed to evaluate certain properties of an algorithm, they are useful to verify the viability of fundamental concepts introduced into proposed PSO variants. To investigate the feasibility of the proposed algorithms in real-world applications, three engineering design problems are also employed for the performance evaluations.

Finally, the proposed algorithms, alongside with numerous published PSO variants, are coded and tested in the MATLAB® R2012b with Intel ® Core ™ i7-2600 CPU @ 3.40GHz and 4GB RAM environment.

1.7 Thesis Outline

This chapter briefly introduced the research background and some preliminary knowledge regarding the global optimization and the algorithms used to solve this task, particularly on the PSO. The problem statements, research objectives, and research scope of this research are included in this chapter. The rest of this thesis is structured as follows.

In Chapter 2, a comprehensive review of the existing PSO variants is presented. The mechanism of the recently proposed TLBO is also described, considering that this algorithm plays an important role in the next chapter. The advantages and limitations of these PSO variants and TLBO are reviewed to gain a deeper understanding on their conceptual successes and shortcomings. The 30 benchmarks problems and three engineering design problems used in the performance evaluations are also discussed in this section. Finally, the details of the performance metrics and the statistical analyses used in the performance comparisons are provided.

In Chapter 3, the first proposed PSO algorithm, namely the Teaching and Peer-Learning Particle Swarm Optimization (TPLPSO) is introduced. This chapter starts with the research ideas that lead to the development of TPLPSO, followed by the main modifications introduced. Simulation results of TPLPSO in solving the benchmark and engineering design problems are obtained and compared with those from the state-of-art algorithms.

Chapter 4 proposes the second enhanced PSO algorithms, namely the Adaptive Two-Layer Particle Swarm Optimization with Elitist Learning Strategy (ATLPSO-ELS). The adaptive task allocation mechanisms of ATLPSO-ELS are described in sufficient detail. At the end of Chapter 4, the comparative studies on the performances of ATLPSO-ELS and its peer algorithms are conducted based on their simulation results.

Chapter 5 presents the technical details of the third enhanced PSO algorithms, namely the Particle Swarm Optimization with Adaptive Time-Varying Topology Connectivity (PSO-ATVTC). Several design issues of PSO-ATVTC are carefully addressed. Finally, the experimental results are presented, analyzed, compared, and discussed.

The fourth proposed PSO algorithm, namely the Particle Swarm Optimization with Dual-Level Task Allocation (PSO-DLTA), is described in Chapter 6. The research idea that inspires the development of PSO-DLTA is first explained, followed by the methodology of this algorithm. At the end of this chapter, the effectiveness of the proposed PSO-DLTA is investigated through an extensive amount of simulations. The overall performances of the four PSO algorithms proposed in this research, i.e., TPLPSO, ATLPSO-ELS, PSO-ATVTC, and PSO-DLTA, are also compared and discussed.

Finally, Chapter 7 draws the conclusions and highlights the contributions of this research. A number of interesting directions to be pursued are detailed as future works.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Particle swarm optimization (PSO) has emerged as a promising optimization tool for solving various types of global optimization problems and real-world engineering design problems since its inception. A demanding yet stimulating undertaking of PSO-based optimization technique is to solve a given optimization problem with the best search accuracy and the fastest convergence speed, while incurring the least computational complexity. These contradictory goals have led to the advancement of PSO-based optimization techniques because various innovative approaches have been proposed in the past decades to improve the algorithm's performance.

This chapter starts with a comprehensive review on the basic PSO algorithm and the recent proposed PSO variants. Specifically, Section 2.2 offers an in-depth treatment of prevalent subject matters popularly discussed in PSO literature. The recently proposed Teaching and Learning Optimization (TLBO) is briefly reviewed in Section 2.3 for an insight into its theoretical and methodological fundamentals. The 30 benchmark problems and the three engineering design problems used for the PSO performance evaluation are introduced in Section 2.4. In Section 2.5, the performance metrics and the statistical analyses employed in the performance comparison of algorithms are presented. Finally, Section 2.6 concludes this chapter.

2.2 Particle Swarm Optimization and Its Variants

This section discusses the mechanism of the basic PSO (BPSO). In what follows, the comprehensive reviews of several state-of-arts PSO variants will be provided. The advantages and limitations of these proposed PSO variants are also summarized in this section to gain a deeper understanding on their conceptual successes and shortcomings.

2.2.1 Basic Particle Swarm Optimization

In PSO, the PSO swarm is modeled as a group of particles with negligible mass and volume that navigate through the D dimensional hyperspace. D denotes the dimension of search space and it is interpreted as the number of variables being optimized in a given problem.

In the context of optimization, the location of each particle in the hyperspace represents a potential solution of a given problem. Meanwhile, the fitness value of each particle determines the solution quality. It must not be confused with the concept of fitness value and the objective function value ($ObjV$) as mentioned in Section 1.1. In the framework of evolutionary optimization terminology, "fitness" is a measure of the goodness of each solution. All optimization techniques are based on the fitness optimization, which leads to problem-dependent objective function minimization/maximization at the end of the optimization problems. As one of the scopes of this research work, this thesis considers the minimization problems because the benchmark and real-world engineering problems used to evaluate the algorithm's search performance have the global optima at the valleys of the fitness landscapes instead of the peaks. Thus, to state that solution A is better or fitter than solution B, the $ObjV$ of the former must always be lower than the that of the latter, i.e., $ObjV(\text{A}) < ObjV(\text{B})$.

In general, the current state of each particle is associated with two vectors, namely the position vector $X_i = [X_{i1}, X_{i2}, \ldots, X_{iD}]$ and the velocity vector $V_i = [V_{i1}, V_{i2}, \ldots, V_{iD}]$, where i denotes the particle's index in the search space. Unlike the other CI-based algorithms, each PSO particle i has the capability of memorizing the personal best experience (i.e., self-cognitive experience) that it ever attained and this experience is represented by the personal best position vector of $P_i = [P_{i1}, P_{i2}, \ldots, P_{iD}]$. Another notable best experience that is tracked by particle i is the group best experience (i.e., social experience) obtained by any particle in the neighbors of the particle i. This experience is expressed as the neighborhood best position vector of $P_n = [P_{n1}, P_{n2}, \ldots, P_{nD}]$ and its value depends on the particle's topological structure. For instance, the topology of global version PSO as illustrated in Figure 2.1(a) is fully-connected, given that each particle takes all the population as its topological neighbors. In

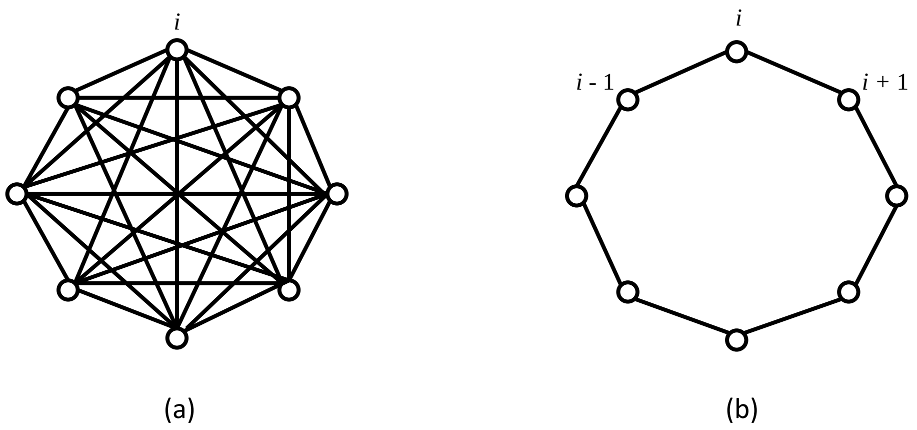

Figure 2.1: Topological structures of PSO with (a) the fully-connected topology and (b) the local ring topology. Each circle represents one particle, while each line represents the connection of one particle to others in the population (del Valle et al., 2008).

the fully-connected topology, a particle *i* uses the best experience of the entire swarm as its neighborhood best experience. This best value is known as the global best position and it is denoted as $P_g = [P_{g1}, P_{g2}, \ldots, P_{gD}]$. On the other hand, Figure 2.1(b) demonstrates a local version PSO with the ring topology, where each particle *i* only considers two of its most adjacent particles (i.e., the particles with indexes of i - 1 and i + 1) as its neighborhood members. For ring topology, the best neighborhood experience of particle *i* is selected from the personal best experiences among the particles $i - 1$, i, and $i + 1$. The selected best value is known as the local best position and it is expressed as $P_l = [P_{l1}, P_{l2}, \ldots, P_{lD}]$.

During the search process, the velocity vector of each particle *i* is stochastically adjusted according to its self-cognitive experience P_i and the social experience P_n (Kennedy and Eberhart, 1995, Eberhart and Shi, 2001). The inclusion of social experience during the velocity updating process implies the collective and collaboration behaviors in PSO swarm, given that the most successful particle shares its useful information to its neighborhood members to guide the search. The new position of particle *i* in the search space is subsequently computed based on the updated velocity vector. Mathematically, at the $(t + 1)$-th iteration of the search process, the d-th dimension of particle *i*'s velocity, $V_{i,d}(t + 1)$, and position $X_{i,d}(t + 1)$ are updated as follows:

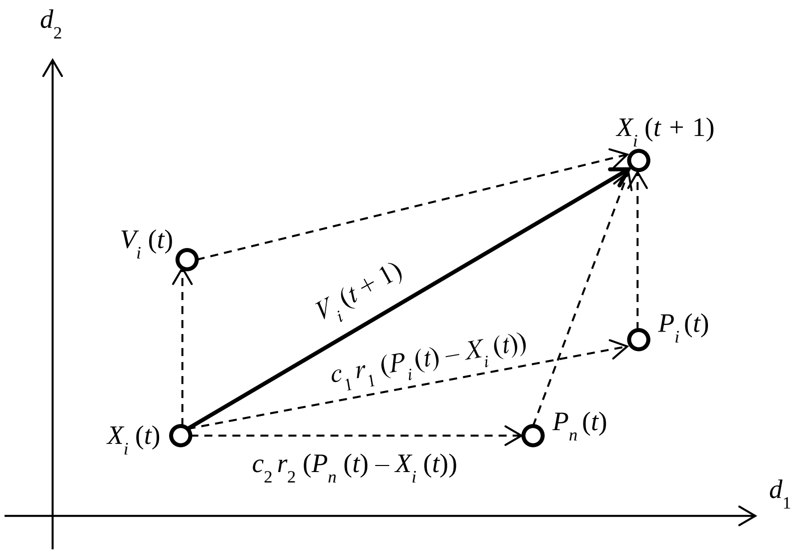

Figure 2.2: Particle *i*'s trajectory in the two-dimensional fitness landscape (Li, 2010).

$$V_{i,d}(t+1) = \omega V_{i,d}(t) + c_1 r_{1,d}(P_{i,d}(t) - X_{i,d}(t)) + c_2 r_{2,d}(P_{n,d}(t) - X_{i,d}(t)) \qquad (2.1)$$

$$X_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t+1) \qquad (2.2)$$

where $i = 1, 2, \ldots, S$; S is the population size of particle swarm; c_1 and c_2 are the acceleration coefficients that control the influences of self-cognitive (i.e., P_i) and social (i.e., P_n) components of particle, respectively; $r_{1,d}$ and $r_{2,d}$ are two random numbers in the range of [0, 1] with uniform distribution; and ω is the inertia weight used to determine how much the previous velocity of a particle is preserved (Shi and Eberhart, 1998). Figure 2.2 demonstrates the trajectory of particle *i* in the two-dimensional fitness landscape.

As shown in the right of Equation (2.1) and Figure 2.2, the velocity component of each particle is decomposed into three components (del Valle et al., 2008, Li, 2010). The first component is called the inertia component, given that this component models the tendency of the particle *i* to persist its previous search direction and to enable it searches for more unexplored regions. Meanwhile, the second and third components are known as the self-cognitive and social components of particle *i*, respectively. The self-cognitive component treats the particle *i* as an isolated being and adjust the particle’s behavior according to its own experience. In contrary, the social component suggests each particle *i* to ignore its own experience and adjust its trajectory according to the best particle in the neighborhood. Both of the inertia and self-cognitive components governs the exploration search capability of particle *i*, while the particle *i*'s exploitation behavior is influenced by the social components.

Once the updated position of particle *i* is obtained, the fitness of this new solution is evaluated. Specifically, the objective function value of the updated position is computed as $ObjV\ [X_i(t+1)]$ and then compared with $ObjV\ [P_i(t)]$, i.e., the objective function value of the personal best position of particle *i*. For minimization problem, the updated $X_i(t+1)$ is considered to have better fitness than $P_i(t)$ if the former has lower objective function value than the latter, i.e., $ObjV\ [X_i(t+1)] < ObjV\ [P_i(t)]$. In this scenario, the personal best position of particle *i* is replaced by the updated $X_i(t+1)$ at iteration $(t + 1)$, as illustrated in Equation (2.3). On the other hand, if $ObjV\ [X_i(t+1)] > ObjV\ [P_i(t)]$, it implies that the updated $X_i(t+1)$ of particle *i* has worse fitness than its $P_i(t)$. Thus, the personal best position of particle *i* is not replaced at iteration $(t + 1)$, as shown in Equation (2.3).

$$P_i(t+1) = \begin{cases} X_i(t+1), & \text{if } ObjV[X_i(t+1)] < ObjV[P_i(t)] \\ P_i(t), & \text{otherwise} \end{cases} \tag{2.3}$$

The neighborhood best position (i.e., P_n) of each particle P, on the other hand, is identified from the personal best position vectors of all particles that are located in the same neighborhood. At each iteration *t*, the neighborhood best position of a particle swarm is identified as follows:

$$P_n(t) = \underset{\forall i \in [1, S_{ns}]}{\arg} \min[ObjV(P_i(t))] \tag{2.4}$$

where S_{ns} denotes the neighborhood size of the particle swarm and this value depends on the topological structure of the PSO. For example, the value of S_{ns} in the fully-connected topology [as illustrated in Figure 2.1(a)] is equal to the population size, i.e., $S_{ns} = S$, considering that each particle in this topology takes all the population as its topological neighbors. For ring topology [as depicted in Figure 2.1(b)], each particle *i* only considers two of its most adjacent particles as its neighborhood members and therefore $S_{ns} = 3$.

BPSO

Input: Population size (S), dimensionality of problem space (D), objective function (F), the initialization domain (RG), problem's accuracy level (ε)

Generate initial swarm and set up parameters for each particle;
while the termination criterion is not satisfied **do**
    for each particle i **do**
        Update the velocity V_i and position X_i using Equations (2.1) and (2.2), respectively;
        Perform fitness evaluation on the updated X_i;
        if $ObjV(X_i) \le ObjV(P_i)$ **then**
            $P_i = X_i$, $ObjV(P_i) = ObjV(X_i)$;
            if $ObjV(X_i) \le ObjV(P_g)$ **then**
                $P_g = X_i$, $ObjV(P_g) = ObjV(X_i)$;
            end if
        end if
    end for
end while

Output: The best found solution, i.e. the global best particle's position (P_g)

Figure 2.3: Basic PSO algorithm.

Without loss of generality, the remaining section of this thesis refers the BPSO as the global version of PSO, as illustrated in Figure 2.1(a). In other words, the neighborhood best position P_n of BPSO refers to the global best position P_g. The implementation of the BPSO is illustrated in Figure 2.3. Several stopping criteria had been proposed in literature to terminate the BPSO. Specifically, the BPSO can be terminated when (1) the predefined maximum number of iteration or the function evaluation is reached, (2) the predefined accuracy of the solution has been achieved, (3) the fitness improvement of the swarm becomes insignificant, and (4) the normalized radius of swarm is close to zero, implying the sufficient convergence of swarm. In this thesis, the maximum number of fitness evaluation (FE_{max}) is selected as the termination criterion because the fitness evaluation process consumes more computational resources than other PSO mechanisms during the optimization process (Feng et al., 2013, Mezura-Montes and Coello, 2005). It must not be confused with the concept of number of iteration and number of fitness evaluation (FE). The former is updated when all particles in the population have updated and evaluated the fitness of their respective new positions. Meanwhile, FE is updated when a particular particle has updated and evaluated its new position in the search space. Intuitively, the FE number consumed in an optimization problem is higher than the iteration number and thus it serves as a better indicator to measure the computation cost required by an algorithm to solve a given optimization problem.

2.2.2 Variants of Particle Swarm Optimization

As mentioned in the previous chapter (Section 1.5), the original PSO suffers with some demerits which could degrade its optimization capability and restrict its wider application in real-world problems (Eberhart and Shi, 2001, Banks et al., 2007, del Valle et al., 2008). Numerous research works have been proposed in the past decades to address the aforementioned drawbacks and to improve the performance of PSO.

In order to provide a comprehensive review and broader sight on the state-of-art PSO variants in global optimization, a classification scheme as depicted in Figure 2.4 is used to pool the PSO variants that are modified by similar approach into the same category. Specifically, the modification and improvement performed on the PSO are categorized into four major approaches, namely the (1) parameter adaptation, (2) modified population topology, (3) modified learning strategy, and (4) hybridization with orthogonal experiment design (OED) technique (Montgomery, 1991, Hedayat, 1999). The diverse ideas of scholar who contributed to the improvement of PSO in each major approach are reviewed comprehensively in the following subsections.

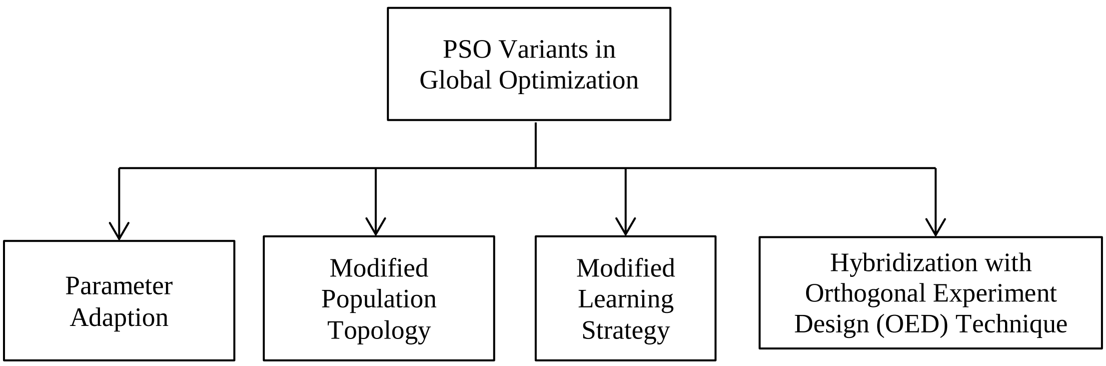

Figure 2.4: The classification scheme of state-of-art PSO variants in global optimization.

2.2.2(a) Parameter Adaptation

Parameter adaptation is the one of the earliest research directions attempted by the researchers to improve the PSO. This approach studies the effects of PSO parameters on the dynamical behaviors of swarm, followed by the tuning of these parameters to alter the particle's movement behavior. The thorough convergence analysis and stability studies of PSO also lead to the introduction of new parameter that is useful to achieve the better optimization outcomes.

Shi and Eberhart (1998) proposed a parameter called inertia weight ω to balance the exploration and exploitation capabilities of PSO swarm. Various strategies have been developed to tune the parameter ω since then. In their earlier work, Shi and Eberhart (1998) suggested that the parameter ω with a fixed value lying between 0.8 and 1.2 is able to achieve a good convergence behavior of swarm. Later, a time-varying scheme that linearly decreases ω with the iteration number was introduced by Shi and Eberhart (1999). Accordingly, the value of ω is initially set to a larger value (i.e., ω = 0.9) to allow the particles explore the search spaces in the early stage of optimization. Once the optimal region is located, ω is gradually decreased to 0.4 to refine the optimal search area in the latter stage of optimization. Chatterjee and Siarry (2006), and Cai et al. (2008), on the other hand, proposed to vary the ω in nonlinear manner. As compared to the linear variation approach, the nonlinear variation of ω enables the particle swarm to explore the search space in more aggressively manner during the early stage of optimization, in order to locate the optimal region with faster rates. Clerc and Kennedy (2002) performed thorough theoretical studies on the PSO convergence properties and subsequently proposed a similar parameter known as the constriction factor χ. Accordingly, the parameter χ could prevent the swarm explosion by providing the damping effect on the particle's trajectory. Experimental study revealed that the parameters ω and χ are algebraically equivalent when the condition of $\omega = \chi = 0.729$ is fulfilled (Eberhart and Shi 2000).

Acceleration coefficient (c_1 and c_2) is another subject of great interest in the parameter adaptation approach, considering that c_1 and c_2 govern the exploitation and exploration capabilities of PSO swarm, respectively. According to the studies performed by Ozcan and Mohan (1999), the PSO particle is observed to oscillate around a sinusoidal path when the value of $c = c_1 + c_2$ is set between 0 and 4.0. The oscillation frequency and complexity of the sinusoidal path increase with the value of c. When c is set larger than 4.0, the particle's trajectory starts to diverge and swarm explosion occurs. Based on their experimental studies, Ozcan and Mohan concluded that the maximum value for c should be 4.0 (Ozcan and Mohan, 1999). However, the values of c_1 and c_2 need not to be always equal to each other, given that the influences of self-cognitive and social components of PSO swarm should be different based on the nature of problem. Suganthan (1999) attempted to improve the PSO performance by linearly decreasing both c_1 and c_2 with time. However, they observed that the PSO with fixed c_1 and c_2 (i.e., 2.0) yields better solutions than the linearly decreasing scheme. Ratnaweera et al. (2004) continued to investigate the feasibility of dynamic c_1 and c_2 in improving the PSO performance. They revised the swarm behavior and found out that the self-cognitive component is more important during the early stage of optimization, considering that particles need to wander through the unexplored search region. In the latter stage, the influence of social component becomes more significant to encourage the PSO swarm converges towards the already found optimal regions. Based on these observations, Ratnaweera et al. (2004) proposed a time-varying acceleration coefficient (TVAC) strategy to linearly decrease and increase the values of c_1 and c_2 with time, respectively. Two PSO-TVAC variants, namely the PSO-TAVC with mutation (MPSO-TVAC) and self-organizing hierarchical PSO-TVAC (HPSO-TVAC), are developed in their work. Both of these variants employ the mutation and velocity re-initialization strategies, respectively, to alleviate the premature convergence issue.

Although the aforementioned works improve the PSO, some notable issues could be identified from these earlier reported simple rule-based parameter tuning strategies. First, most of the simple rule-based parameter tuning strategies are time-varying and they tend to