TKK Dissertations 161
Espoo 2009

HYBRID NATURE-INSPIRED COMPUTATION METHODS FOR OPTIMIZATION

Doctoral Dissertation

Xiaolei Wang

Helsinki University of Technology
Faculty of Electronics, Communications and Automation
Department of Electrical Engineering

TKK Dissertations 161
Espoo 2009

HYBRID NATURE-INSPIRED COMPUTATION METHODS FOR OPTIMIZATION

Doctoral Dissertation

Xiaolei Wang

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Electronics, Communications and Automation for public examination and debate in Auditorium S4 at Helsinki University of Technology (Espoo, Finland) on the 29th of May, 2009, at 12 noon.

Helsinki University of Technology
Faculty of Electronics, Communications and Automation
Department of Electrical Engineering

Teknillinen korkeakoulu
Elektroniikan, tietoliikenteen ja automaation tiedekunta
Sähkötekniikan laitos

Distribution:
Helsinki University of Technology
Faculty of Electronics, Communications and Automation
Department of Electrical Engineering
P.O. Box 3000 (Otakaari 5)
FI - 02015 TKK
FINLAND
URL: http://sahkotekniikka.tkk.fi/en/
Tel. +358-9-451 4965
Fax +358-9-451 2432
E-mail: xiaolei@cc.hut.fi

ISBN 978-951-22-9858-7
ISBN 978-951-22-9859-4 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: http://lib.tkk.fi/Diss/2009/isbn9789512298594/

TKK-DISS-2593

Multiprint Oy
Espoo 2009

ABSTRACT OF DOCTORAL DISSERTATION	HELSINKI UNIVERSITY OF TECHNLOGY P.O. BOX 1000, FI-02015 TKK http://www.tkk.fi
Author Xiaolei Wang	
Name of the dissertation Hybrid Nature-Inspired Computation Methods for Optimization	
Manuscript submitted 23/2/2009	Manuscript revised 28/4/2009
Date of the defence 29/5/2009	
☐ Monograph	☒ Article dissertation (summary + original articles)
Faculty: Faculty of Electronics, Communications and Automation Department: Department of Electrical Engineering Field of research: Industrial Electronics Opponent(s): Prof. Mark J. Embrechts Supervisor: Prof. Seppo J. Ovaska Instructor: Docent Xiao-Zhi Gao	

Abstract

The focus of this work is on the exploration of the hybrid Nature-Inspired Computation (NIC) methods with application in optimization. In the dissertation, we first study various types of the NIC algorithms including the Clonal Selection Algorithm (CSA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing (SA), Harmony Search (HS), Differential Evolution (DE), and Mind Evolution Computing (MEC), and propose several new fusions of the NIC techniques, such as CSA-DE, HS-DE, and CSA-SA. Their working principles, structures, and algorithms are analyzed and discussed in details. We next investigate the performances of our hybrid NIC methods in handling nonlinear, multi-modal, and dynamical optimization problems, e.g., nonlinear function optimization, optimal LC passive power filter design, and optimization of neural networks and fuzzy classification systems. The hybridization of these NIC methods can overcome the shortcomings of standalone algorithms while still retaining all the advantages. It has been demonstrated using computer simulations that the proposed hybrid NIC approaches are capable of yielding superior optimization performances over the individual NIC methods as well as conventional methodologies with regard to the search efficiency, convergence speed, and quantity and quality of the optimal solutions achieved.

Keywords Nature-Inspired Computation (NIC), hybrid algorithms, optimization	
ISBN (printed) 978-951-22-9858-7	ISSN (printed) 1759-2239
ISBN (pdf) 978-951-22-9859-4	ISSN (pdf) 1759-4584
Language English	Number of pages 80 p. + app. 97 p.
Publisher Multiprint Oy	
Print distribution Department of Electrical Engineering	
☒ The dissertation can be read at http://lib.tkk.fi/Diss/2009/isbn9789512298594/	

Preface

The process of learning and research is rich of mystery and joy during my past four years, and it is a great pleasure for me to work on this thesis. I hope, too, readers will find it worthy and comfortable to read.

First and foremost, I gratefully appreciate the assiduous guidance and encouragement throughout the whole research work from my supervisor, Prof. Seppo J. Ovaska. Without his invaluable advice, I would not have completed this work. I am also thankful to Docent Xiao-Zhi Gao, my instructor, for sharing his profound experience in the research of hybrid soft computing algorithms as well as proposing insightful comments and suggestions on my daily work.

I am much obliged to all the personnel at the Department of Electrical Engineering for providing such a pleasant research atmosphere. Prof. Jorma Kyyrä is especially thanked for his warm-hearted help. Secretaries Anne Hovi, Anja Meuronen, and laboratory technician Ilkka Hanhivaara always help me in handling practical matters. I am very grateful to my colleagues, Lic. Konstantin Kostov and Dr. Jarno Martikainen, with whom I had so many helpful discussions. I can only wish them the best of luck with their future life and research career.

I am thankful to Ms. Marja Leppäharju, the coordinator of Graduate School in Electronics, Telecommunications and Automation (GETA), for her kind assistance in my study.

My sincere acknowledgment is rendered to Prof. Jon Timmis from the University of York, UK for his instructions during his academic visit in Finland.

I would like to express my deep appreciation to Profs. Alden H. Wright and David A. Pelta due to their highly professional comments and constructive criticism for the improvement of my thesis.

I gratefully acknowledge the financial aid granted by the GETA, Emil Aaltosen säätiö and Tekniikan edistämissäätiö.

Last but not the least, I want to thank my parents and husband, who are indispensable in my life, study and work.

Otaniemi, April 2009

Xiaolei Wang

Table of Contents

List of Abbreviations

Ab	Antibody
ACO	Ant Colony Optimization
Ag	Antigen
AIS	Artificial Immune Systems
ANN	Artificial Neural Networks
AS	Ant System
C-C	Cascade-Correlation
CI	Computational Intelligence
CSA	Clonal Selection Algorithm
CSP	Clonal Selection Principle
DE	Differential Evolution
DSPSO	Dynamical Species-based PSO
EP	Evolutionary Programming
ES	Evolutionary Strategies
FHSClust	Fuzzy Harmony Search Clustering
GA	Genetic Algorithms
GPEA	Geometrical Place Evolutionary Algorithms
HM	HS Memory
HMCR	Harmony Memory Considering Rate
HS	Harmony Search
INPSO	Independent Neighborhoods Particle Swarm Optimization
MEC	Mind Evolutionary Computing
MGA	Multi-national GA
MSA	Modified SA
NHRVGA	Novel Hybrid Real-Value Genetic Algorithms
NIA	Nature-Inspired Algorithms
NIC	Nature-Inspired Computation
PAR	Pitching Adjust Rate

PSO	Particle Swarm Optimization
SA	Simulated Annealing
SC	Soft Computing
SI	Swarm Intelligence

List of Symbols

c_1	Cognitive parameter
c_2	Social parameter
C	Temporary pool
C_1	Mutated antibody pool
E	Fitness of candidate
$E^{'}$	Fitness of new candidate
f(x)	Objective function
F	Search domain
g	Number of global ants
$g(\hat{\theta}_k)$	Gradient at $\hat{\theta}_k$
G	Group index
i	Particle index *or*
	Region index *or*
	Index
j	Index
k	Number of generations *or*
	Index
$L(\hat{\theta})$	Objective function
M	Memory cells
n	Iteration *or*
	Problem dimension
n_r	Global ants
n_x	Number of input variables
n_{ϖ}	Dimension of $\varpi(t)$
N	Size of harmony memory *or*
	A set of feasible points

N_G	Number of groups in the MEC
p_{gd}^n	Position of the best particle in the swarm
p_{id}^n	Particle's best previous position
$P_i(t)$	Transition probability of region i at time t
P	Acceptance probability
P_{init}	Initial antibody pool
P_r	Best candidates
$r\ (k)$	Randomly selected chromosome at generation k
$r'(k+1)$	Trial update of $r(k)$
$r''(k+1)$	Trial update of $r'(k+1)$
r_1	Random value in the range of $[0,1]$
r_2	Random value in the range of $[0,1]$
S	Search domain *or*
	Population size
S_G	Group size
t	Time
T	Temperature
T_0	Initial temperature
v_{id}^n	Velocity of particle i at iteration n
w	Inertia weight
x	Solution
$\mathbf{x}$	Solution vector
$x^{'}$	New solution
x^*	Global optimum
x_N^*	x^* in domain N
$x^*(t)$	Optimum at time t
x_{id}^n	Position of particle i at iteration n

α_k	Step size
λ	Learning coefficient *or* Pre-determined weight
$\hat{\theta}$	Possible solution
$\hat{\theta}_0$	Arbitrary initial point
ρ	Pheromone evaporation rate
$\tau_i(t)$	Total pheromone at region i at time t
$\Delta\tau_i$	Pheromone increment
$\varpi(t)$	Time-dependent objective function control parameters at time t

List of Publications

This dissertation consists of an overview and the following publications:

[P1] X. Wang, “Clonal selection algorithm in power filter optimization,” in *Proceedings of the IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications*, Espoo, Finland, June 2005, pp. 122-127.

[P2] X. Wang, X. Z. Gao, and S. J. Ovaska, “A hybrid optimization algorithm in power filter design,” in *Proceedings of the 31st Annual Conference of the IEEE Industrial Electronics Society*, Raleigh, NC, November 2005, pp. 1335-1340.

[P3] X. Wang, X. Z. Gao, and S. J. Ovaska, “A novel particle swarm-based method for nonlinear function optimization,” *International Journal of Computational Intelligence Research*, vol. 4, no. 3, pp. 281-289, 2008.

[P4] X. Z. Gao, X. Wang, and S. J. Ovaska, “Uni-modal and multi-modal optimization using modified harmony search methods,” *International Journal of Innovative Computing, Information and Control*, in press.

[P5] X. Wang, X. Z. Gao, and S. J. Ovaska, “Fusion of clonal selection algorithm and harmony search method in optimization of fuzzy classification systems,” *International Journal of Bio-Inspired Computation*, vol. 1, no. 1-2, pp. 80-88, 2009.

[P6] X. Z. Gao, X. Wang, and S. J. Ovaska, “Fusion of clonal selection algorithm and differential evolution method in training cascade-correlation neural network,” *Neurocomputing*, in press.

[P7] X. Wang, X. Z. Gao, and S. J. Ovaska, “A simulated annealing-based immune optimization method,” in *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, Porvoo, Finland, September 2008, pp. 41-47.

[P8] X. Wang, X. Z. Gao, and S. J. Ovaska, “A hybrid optimization algorithm based on ant colony and immune principles,” *International Journal of Computer Science and Applications*, vol. 4, no. 3, pp. 30-44, 2007.

1. Introduction

1.1 Background of optimization

Optimization is a computational science that studies techniques for finding the 'best' solutions. It has been widely employed in a large variety of fields, including transportation, manufacturing, physics, and medicine. Real-world optimization frequently suffers from the following problems [Spa03]:

- Difficulties in distinguishing global optimal solutions from local optimal ones.
- The presence of noise in evaluating the solutions.
- The 'curse of dimensionality' causes the size of the search space to grow exponentially with the problem dimension.
- Difficulties associated with given constraints and the need for problem-specific optimization methods.

Numerous conventional optimization schemes have been proposed and developed. In fact, we have observed their successful applications and implementations, ranging from new drug design and protein structure prediction to power system scheduling. Among these schemes, the steepest descent method is a typical one in that it is based on the derivative of the objective functions to be optimized. Unfortunately, they also face difficulties in meeting the growing needs of modern industry, in which the existing optimization problems tend to be dynamic, constrained, multi-variable, multi-modal, and multi-objective. Conventional optimization methods have been limited by a weak global search ability, instability, and inefficiency, especially when attempting highly nonlinear optimization tasks. For example, the gradient information needed in the steepest descent method can be expensive, time-consuming, or even impossible to obtain under certain circumstances. Moreover, most of the conventional optimization approaches are not efficient enough in dealing with practical large-scale systems.

1.2 Nature-inspired computation methods

The Nature-Inspired Computation (NIC) methods refer to as those algorithms derived by mimicking natural phenomena and biological models [Cas07] [Ova06]. For instance, the Clonal Selection Algorithm (CSA) draws its inspiration from natural immune systems, which can prevent the human body from invasion by disease-causing pathogens. The collective behaviors of the foraging of ants, mound construction of termites, nest-building of wasps, and web-weaving of spiders have been studied and have inspired the so-called Swarm Intelligence (SI) algorithms. The NIC methodologies, such as the CSA, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing (SA), Harmony Search (HS), Differential Evolution (DE), and Mind Evolution Computing (MEC), are emerging types of Soft Computing (SC) or Computational Intelligence (CI) methods [Zad96].

Many engineering problems are indeed complex and have built-in uncertainties as well. It is well known that conventional optimization approaches usually require some prior information beforehand, which is often either difficult to acquire or even unavailable. In recent years, NIC methods have gained considerable attention from different communities [Emb97]. Compared with conventional optimization schemes, NIC methods offer more suitable candidates for dealing with the demanding problems faced by industry, and can thus offer us robust, competitive solutions. As a matter of fact, for optimization applications in fields such as pattern recognition, self identity, data analysis and machine learning, NIC methods are capable of outperforming the classical techniques by providing better and more flexible solution choices.

1.3 Hybridization of nature-inspired computation optimization methods

The NIC algorithms have their own advantages and disadvantages, since they have received their inspiration from individual natural phenomena; for example, the PSO is inspired by the flocking of birds, fish schooling, and the ACO foraging behaviors of ants. If used separately, the weaknesses of these algorithms may hinder their wider employment in attacking challenging optimization problems. That is to say, these NIC methods are complementary rather than competitive. It is generally advantageous to apply them in combination instead of individually. For example, the harmful particle clustering of the original PSO method can be overcome by introducing operations from other NIC algorithms so as to randomly disturb the trend of solution centralization. The performance of the regular HS method can be significantly enhanced if the harmony memory members are fine-tuned on the basis of the CSA. Thus, utilizing the advantages of various NIC algorithms while avoiding their drawbacks promotes advances in hybrid NIC techniques.

The past decade has witnessed the overwhelming success of hybrid NIC optimization methods that can effectively combat with their individual drawbacks while benefiting from each other's strengths. The hybridization strategies have been developed with the aim of coping with specific types of optimization problems. For instance, the ACO, a promising NIC technique, has been merged with the CSA for solving those optimization problems under dynamic environments [P8]. Fusion of the SA and PSO can result in a hybrid PSO with an improved global search capability for nonlinear problems. To summarize, the motivation for hybridization is to improve convergence acceleration, robustness, and reliability. In general, hybrid NIC methods can be classified into different categories according to the measures used, e.g., motivation for hybridization and architecture of hybridization. As an illustrative example, we can divide them into 'preprocessors and postprocessors', 'cooperators', and 'embedded operators' based on the relationship among all the NIC methods involved. Actually, a careful and comprehensive analysis of the classification of the hybridization would help us not only gain a deep understanding of the NIC methods but also choose the best combinations for the targeted optimization problems.

1.4 Aim of this dissertation

The aim of this dissertation is to explore the hybrid NIC methods applied to optimization. More precisely, in our work, we first study various types of hybridization of the aforementioned NIC algorithms, and propose several new hybrid NIC techniques, e.g., CSA-DE, HS-DE, and CSA-SA. Their underlying motivations, principles, structures, and algorithms are analyzed and discussed in details. We next investigate the performance of our hybrid NIC methods in handling nonlinear, multi-modal, and dynamic optimization problems. This work also examines the application of these novel optimization strategies for nonlinear function optimization, optimal LC passive power filter design, and optimization of neural networks and fuzzy classification systems. Computer simulations are made to compare the proposed techniques with conventional optimization methodologies as well as individual NIC methods. It has been demonstrated that the hybrid NIC approaches can yield superior optimization performances over the existing schemes with regard to search efficiency, convergence speed, as well as the quantity and quality of the optimal solutions obtained.

In this dissertation, a concise introduction to the optimization-related issues, including definition, classification, and traditional optimization methods, is given in Chapter 2. Chapter 3 presents a survey of seven popular NIC methods: the CSA, PSO, ACO, SA, HS, DE, and MEC. A brief review of the hybrid optimization techniques based on the above NIC algorithms, which have been proposed, developed, and reported in the literature, is provided in Chapter 4. Chapter 5 summarizes the main results of the publications. Finally, some conclusions and remarks are drawn in Chapter 6.

2. Introduction to Optimization

Generally, optimization is a process of searching for the best possible solution to a given problem, which is subject to certain constraints. Modern industry and science are rich in optimization problems [Pri99]. For example, there are different ways to design a cylindrical pressure vessel, but which one costs the least? There are various methods to construct a telecommunication network, but which one is the most reliable? This chapter briefly defines optimization and provides a background to typical optimization problems and optimization methods.

2.1 Definition of optimization

Optimization simply means finding the best solution or operating a system in the most effective way, as shown in Fig 2.1 [Hau98]. It can be considered as a process of adjusting the input in order to attain the optimal (minimal or maximal) system output. Optimization algorithms represent the search approaches for obtaining the optimal solution to an optimization problem, possibly subject to a set of constraints [Eng05]. Particularly, the search environment may change over time, the constraint conditions may restrain the search space, and the optimal solutions may be intertwined with many neighboring candidates. For any given optimization problem, optimization can consist of the following three principal components:

- Objective function: quantity to be optimized (minimized or maximized).
- Variables: inputs to the objective function.
- Constraints: limitations assigned to the variables.

Thus, the goal of an optimization method is to assign possible values to the variables so as to acquire the optimal solution to the objective function, while satisfying all constraints.

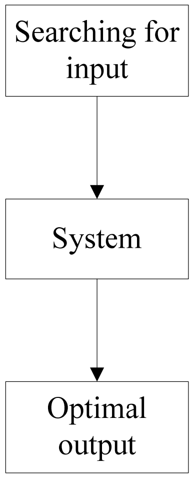

Fig. 2.1. Basic structure of optimization.

Based on their qualities, optima can be classified into either global or local optima, as shown in Fig. 2.2. Figure 2.2 illustrates a minimization problem in the feasible search space $F \subseteq S$. Obviously, only the global optima are the best solutions among all the candidates. Hence, the definitions of the global and local optima of the minimization problems are given as follows [Eng05]:

- Global minimum: The solution $x^* \in F$ is a global optimum of the objective function $\mathrm{f}(x)$, if

  $$\mathrm{f}(x^*) < \mathrm{f}(x), \quad \forall x \in F, \tag{2.1}$$

  where $F \subseteq S$.

- Local minimum: The solution $x_N^* \in N \subseteq F$ is a local optimum of the objective function $\mathrm{f}(x)$, if

  $$\mathrm{f}(x_N^*) < \mathrm{f}(x), \quad \forall x \in N, \tag{2.2}$$

  where $N \subseteq F$ is a set of the feasible points in the neighborhood of x_N^*. Note that these two definitions can be generalized to the maximization problems.

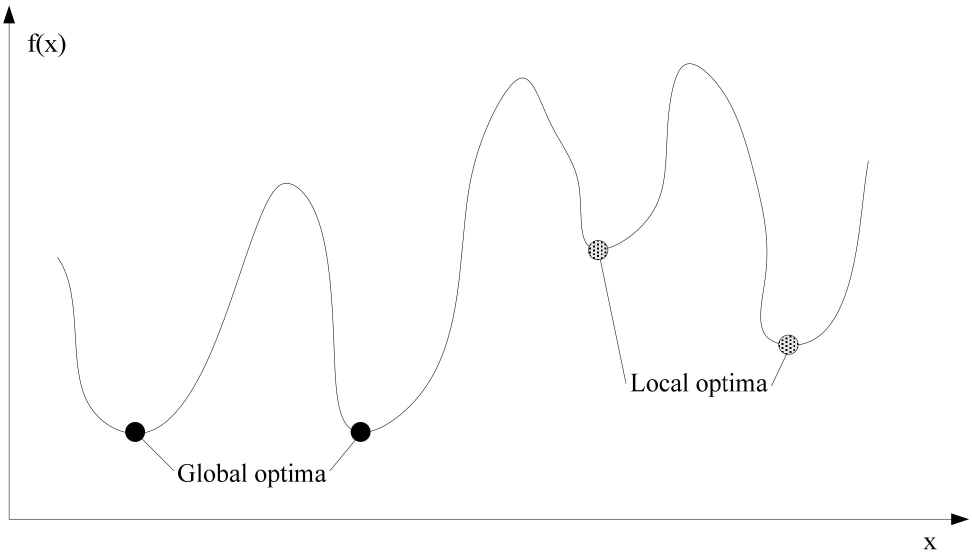

Fig. 2.2. Types of optima.

2.2 Classification of optimization problems

Figure 2.3 shows the general classification of optimization problems. The optimization problems can be classified according to three primary factors (variable, objective function, and output) as well as their different characteristics:

- The constraints to variables: The optimization problems that use only boundary constraints are referred to as unconstrained optimization problems. Constrained optimization problems have additional equality and/or inequality constraints.
- The number of variables: If there is only one variable, the optimization problem is one-dimensional. However, if more than one variable is involved, the problem is referred to as a multi-dimensional optimization problem. Usually, the more variables the problem has, the more complex it is.
- The types of variables: Optimization problems can also be classified by discrete and continuous variables. The former have integer-valued variables, whereas the later consist of continuous-valued variables.
- The nonlinearity of objective functions: A linear optimization problem has the objective functions linear with respect to all the variables. Otherwise, the optimization problem is considered as a nonlinear optimization problem.

- The environment of output: Dynamic optimization means that the output is a time-varying function; for static optimization, the output is always independent of time.
- The number of optima: The optimization problem is uni-modal, if there is only one unique optimal solution. Otherwise, it is a multi-modal optimization problem.
- The number of optimization criteria: If the quantity to be optimized is expressed using only one objective function, the problem is referred to as a uni-objective problem. A multi-objective problem specifies more than one sub-objective that needs to be simultaneously optimized. In this dissertation, we only investigate uni-objective optimization problems.

The focus of this dissertation is on nonlinear uni-modal and multi-modal optimization under both static and dynamic environments. In the following sections, we discuss multi-modal and dynamical optimization problems.

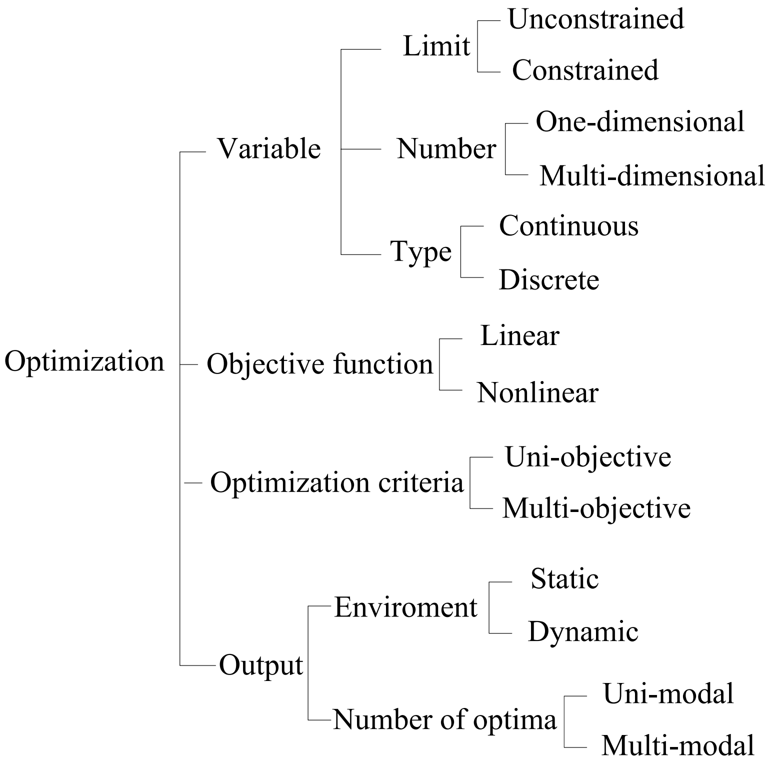

Fig. 2.3. Categories of optimization problems.

2.3 Multi-modal optimization

Multi-modal problems contain many optima including at least one global optimum and a number of local ones in the search space. The goal of the multi-modal optimization is to identify as many of these optima as possible. Hence, the multi-modality of such problems determines the difficulty for any optimization approach in terms of the quantity and quality of the optima located. For example, Figure 2.4 illustrates the solutions to the function $\mathrm{f}(x) = \sin^6(5\pi x)$ in $[0,1]$. The optimization task is to achieve all of the five global optima (maxima). Some multi-modal problems rich of many attractors surrounding the global optima are indeed challenges to the existing optimization techniques. As an illustrative example, Schaffer's function:

$$\mathrm{f}(x,y) = -\left\{0.5 + \frac{\sin^2\left(\sqrt{x^2+y^2}\right) - 0.5}{\left[1+0.001\left(x^2+y^2\right)\right]^2}\right\}, \tag{2.3}$$

where $x \in [-5,5]$ and $y \in [-5,5]$, has only one global optimum with numerous neighboring local optima within the distance of about 10^{-2}, as shown in Fig. 2.5 [Cas02] [Sch89]. Therefore, avoiding the misleading local optima is a key issue in dealing with the multi-modal optimization problems.

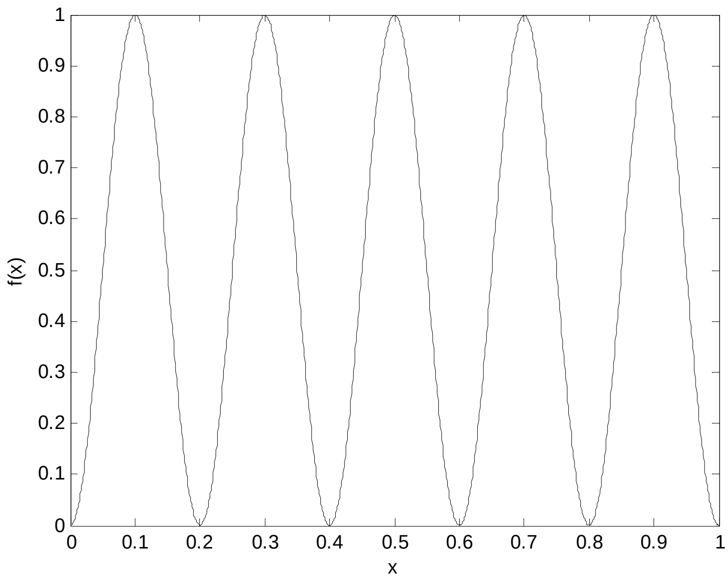

Fig. 2.4. Plot of function $\mathrm{f}(x) = \sin^6(5\pi x)$.

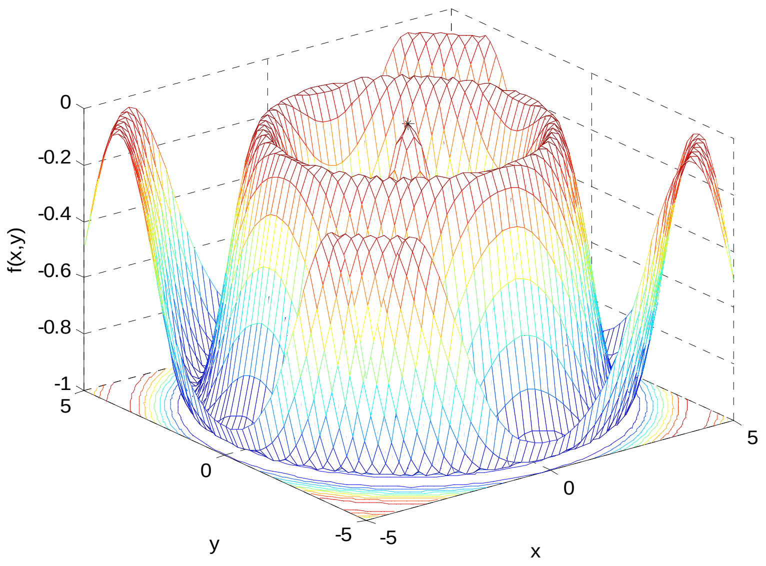

Fig. 2.5. Plot of Schaffer's function.

2.4 Dynamic optimization

Dynamic optimization problems have objective functions that can change over time, thus potentially causing variations in the optima and search space. These problems are defined (in the minimization case) as:

$$\text{minimize } \mathrm{f}[\mathbf{x},\varpi(t)],\ \ \mathbf{x}=(x_1,\cdots,x_{n_x}),\ \ \varpi(t)=\lfloor\varpi_1(t),\cdots,\varpi_{n_\varpi}(t)\rfloor, \tag{2.4}$$

where $\varpi(t)$ is a vector of time-dependent objective function control parameters. The goal is to find

$$x^*(t)=\min_{\mathrm{X}}\{\mathrm{f}[x,\varpi(t)]\}, \tag{2.5}$$

where $x^*(t)$ is the optimum at time t. Therefore, the task of a dynamic optimization algorithm is to locate the optimum and track its trajectory as closely as possible [Eng05]. The location of optimum $x^*(t)$ and value of $\mathrm{f}\left[x^*(t)\right]$ are the two criteria for detecting the variations under a changing environment. There are three types of dynamic environments:

1. Type I: Location of the optimum is subject to change.

2. Type II: Location of the optimum remains the same, but its value changes.
3. Type III: Both the location and value of the optimum change simultaneously.

For instance, Figure 2.6 illustrates the changes in three different dynamic environments that can affect $\mathrm{f}[x,\varpi(t)]$:

$$\mathrm{f}[x,\varpi(t)] = \sum_{j=1}^{n_x} \left[x_j - \varpi_1(t)\right]^2 + \varpi_2(t)\,, \tag{2.6}$$

where $n_x = 2$. Additionally, the complexity of a given dynamic objective function depends on the following three factors [Ang97] [Ram06]:

1. 'Change severity' determines the displacement of the current location position from the static environment.
2. 'Update frequency' determines the number of the generations between each movement of the base function. Generally, the higher the frequency changes, the more complex the optimization problem is.
3. 'Predictability of change' gives an indication as to whether there is a pattern or even a trend in the environment change.

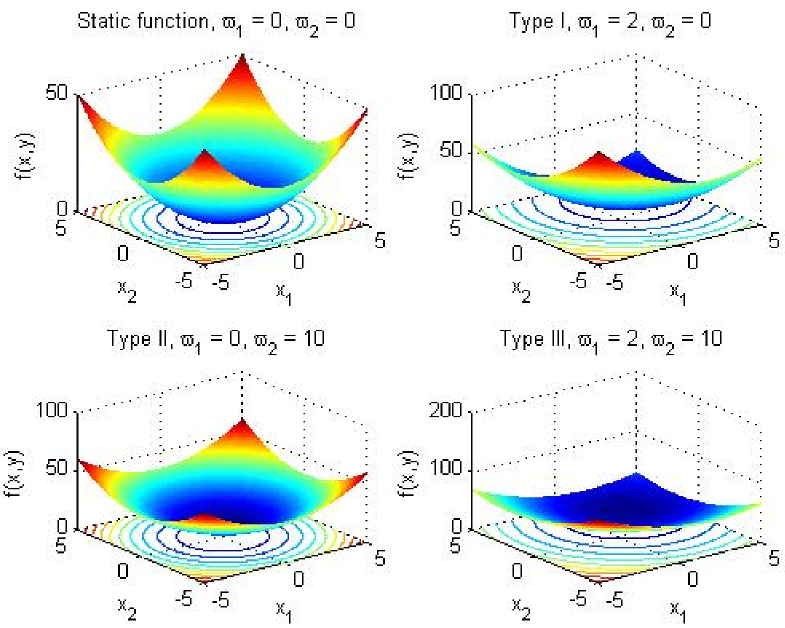

Fig. 2.6. Changes of function (2.6) under dynamic environment.

2.5 Conventional optimization algorithms

Optimization methods can be classified into two essential types: stochastic and deterministic [Jan97]. The former use random elements to update the candidate solutions, and the later involve no randomness. An important and widely used class of deterministic algorithms are gradient-based methods capable of choosing their search directions according to the derivative information of the objective functions. In this dissertation, deterministic optimization techniques, such as steepest descent search and Newton's method, are also classified as conventional optimization algorithms. Nevertheless, to simplify our discussions, we only concentrate here on the most popular conventional optimization algorithm: the steepest descent method.

The steepest descent method is based on the simple principle that from a given $\hat{\theta}$, the best direction in which to proceed is the one that produces the largest local change in the objective function:

$$\hat{\theta}_{k+1} = \hat{\theta}_k - \alpha_k g(\hat{\theta}_k),\ k = 0,1,2,\cdots, \tag{2.7}$$

where k is the iteration index, $g(\hat{\theta}_k)$ is the gradient at $\hat{\theta}_k$, and $\alpha_k > 0$ is the step size used to maintain the search stability so that the operating points do not move too far along the performance surface. Equation (2.7) indicates that the new estimate for the best value of $\hat{\theta}_k$ is equal to the previous one minus a term proportional to the gradient at the current point [Spa03]. That is, the search can start from an arbitrary $\hat{\theta}_0$, and slides down the gradient. Figure 2.7 illustrates a simple convex function minimization case. When $\hat{\theta}_k$ is on either side of the minimum, the update moves towards the minimum in that direction, which is opposite to the sign of the corresponding element in the gradient vector. For example, when $\hat{\theta}_k$ lies left of the minimum, from (2.7), the new estimate will move to the right, according to the search direction given by the positive sign of the gradient component. Apparently, the steepest descent method is stable, straightforward, and easy to apply. If the minima exist, this approach can locate them after enough iterations. However, the steepest descent method is not a universal solution to all the engineering optimization problems, because the gradient information of the objective functions is sometimes difficult or costly if not impossible to obtain. Another drawback of the steepest descent method is that it cannot be guaranteed to always find the global minimum. The ultimate solutions acquired depend on the initial $\hat{\theta}_0$. As a matter of fact, it is easily trapped into the local minima and may thus result in poor optimization performance.

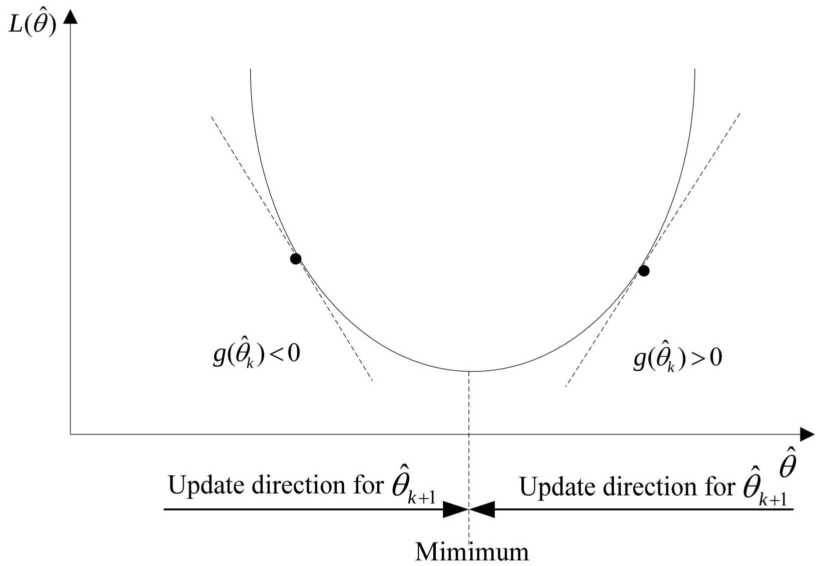

Fig. 2.7. Search using steepest descent method.

2.6 Summary

This chapter gives an overview of the optimization problems and conventional optimization methods including their definition and classification. Unfortunately, conventional optimization algorithms are not efficient at coping with demanding real-world problems without derivate information. As shown in Figs. 2.2 and 2.7, they can only find the local optimum with a high probability, if $\hat{\theta}_0$ is close to it. In other words, selection of the initial points for the deterministic optimization methods has a decisive effect on their final results. However, a foresight of appropriate starting points is not always available in practice. One common strategy is to run the deterministic algorithms with random initialization numerous times and retain the best solution; however, this can be a time-consuming procedure. Therefore, a stochastic nature should be introduced and utilized in these optimization methods so as to enhance their performance. Indeed, stochastic algorithms have become the dominant approaches to optimization during the past decades, due to the following characteristics:

- Derivative free,
- Intuitive guidelines,
- Flexibility,
- Randomness,
- Robustness,
- High parallel property,
- Global optimization.

In Chapter 3, we shall discuss several typical stochastic nature-inspired optimization techniques.

3. Nature-Inspired Optimization Methods

The development of modern science and technology requires flexible, reliable problem-solving methods. The Nature-Inspired Algorithms (NIA) represent such an emerging computing paradigm that draw their metaphorical inspiration from diverse natural sources, including the operation of biological neurons, evolution processes, and natural immune responses. The general framework for developing novel intelligent algorithms inspired from natural phenomena is illustrated in Fig. 3.1 [Tim08]. More precisely, the first step is to probe and study the natural systems from which the inspiration can be drawn. After that, this perception is used to build a simplified representation and model of the sophisticated natural phenomena. Next, the abstract model can provide us with useful principles for designing the new NIA characterizing the underlying mechanisms of the natural phenomena. Finally, the developed NIA is employed to deal with various engineering problems.

In the following sections, we will give a concise introduction to the theory and applications of several typical NIA: Clonal Selection Algorithm (CSA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Differential Evolution (DE), Simulated Annealing (SA), Harmony Search (HS) method, and Mind Evolutionary Computing (MEC). However, it should be emphasized that the well-known evolutionary computation method including the Genetic Algorithms (GA) [Wri91], Evolutionary Programming (EP) [Hoo07], and Evolutionary Strategies (ES) [Fra98] is a matured NIC technique, which has been extensively investigated by many researchers from different communities during the past decades. In this chapter, we aim at analyzing and developing only the above recently emerging NIC algorithms.

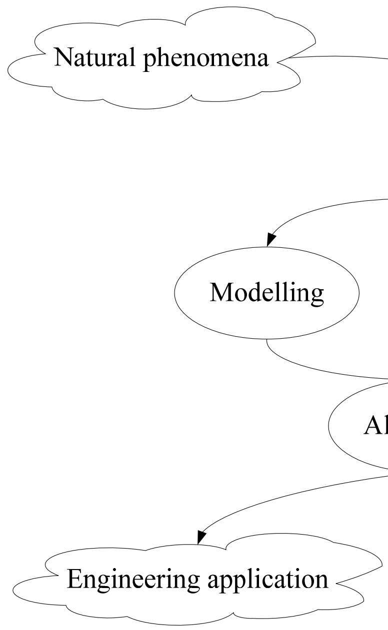

Fig. 3.1 Framework to develop novel NIA.

3.1 Clonal selection algorithm

The natural immune system is a rich inspiration resource for developing intelligent algorithms, and the extension of immunological principles to engineering has promoted a new biologically inspired computing technique, namely Artificial Immune Systems (AIS). The Clonal Selection Algorithm (CSA) inspired by the Clonal Selection Principle (CSP) is one essential branch of the AIS [Tim08] [Wan04]. The CSP describes how the immune response works with regard to an antigenic stimulus [Kim02]. To be more precise, when the B cells are stimulated by the Ag (paratope bound with epitope), they first proliferate (divide), and finally mature to the terminal (nondividing) plasma cells (see Fig. 3.2). Moreover, the proliferated B cells grow in concentration inside the immune network at proliferation rates that are proportional to the affinities of the Antibody-Antigen (Ab-Ag). In fact, those B cells with low affinities are gradually eliminated or edited. In addition to proliferating and differentiating into the plasma cells, the B cells can also differentiate into the memory B cells with longer living lives that will circulate through the blood, lymph, and tissues of the body. The clones, i.e., a set of new cells, are the progenies of the stimulated B cells. In other words, only the cells (antibodies) capable of recognizing the non-self cells (antigens) are selected, and they can, therefore, proliferate. The main ideas of the CSA borrowed from the CSP are [Cas99]:

- Maintenance of memory cells functionally disconnected from the repertoire;
- Selection and cloning of the most stimulated antibodies;
- Affinity maturation and re-selection of clones with higher affinity;
- Mutation rate proportional to cell affinity.

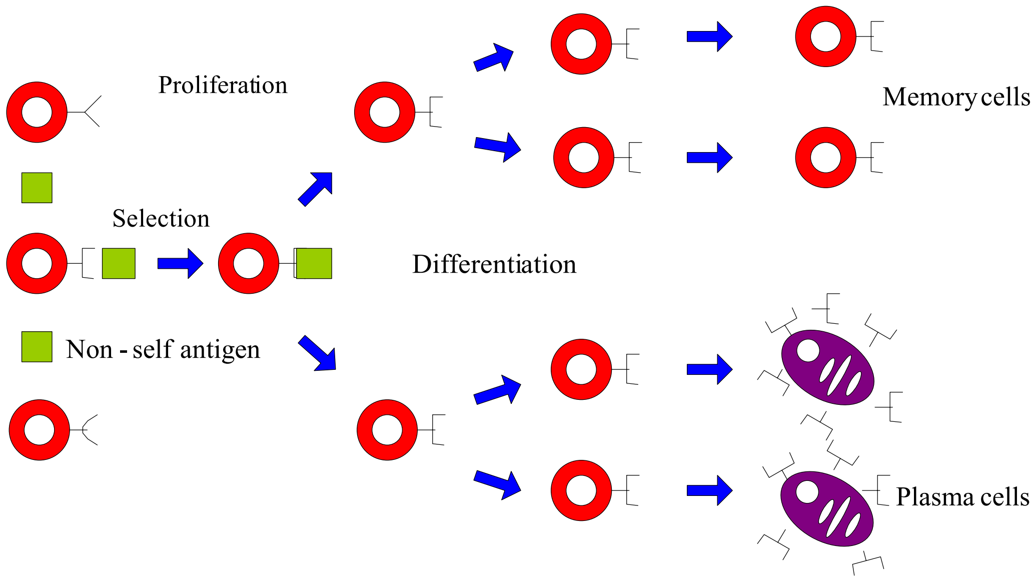

Fig. 3.2. Process of clonal selection.

The diagram of the basic CSA is shown in Fig. 3.3, in which the corresponding steps are explained as follows.

Step 1: Initialize the antibody pool (P_{init}) including the subset of memory cells (M).

Step 2: Evaluate the fitness of all the individuals in population P. The fitness here refers to the Ab-Ag affinity measure.

Step 3: Select the best candidates (P_r) from population P_{init} according to their fitness (affinity with antigen).

Step 4: Clone these best antibodies into a temporary pool (C).

Step 5: Generate a mutated antibody pool (C_1). The mutation rate of each individual is inversely proportional to its fitness.

Step 6: Evaluate all the individuals in C_1.

Step 7: Re-select the individuals with better fitness from C_1 to compose the memory set M. Other improved individuals of C_1 can replace certain members in P_{init} so as to maintain the overall antibody diversity.

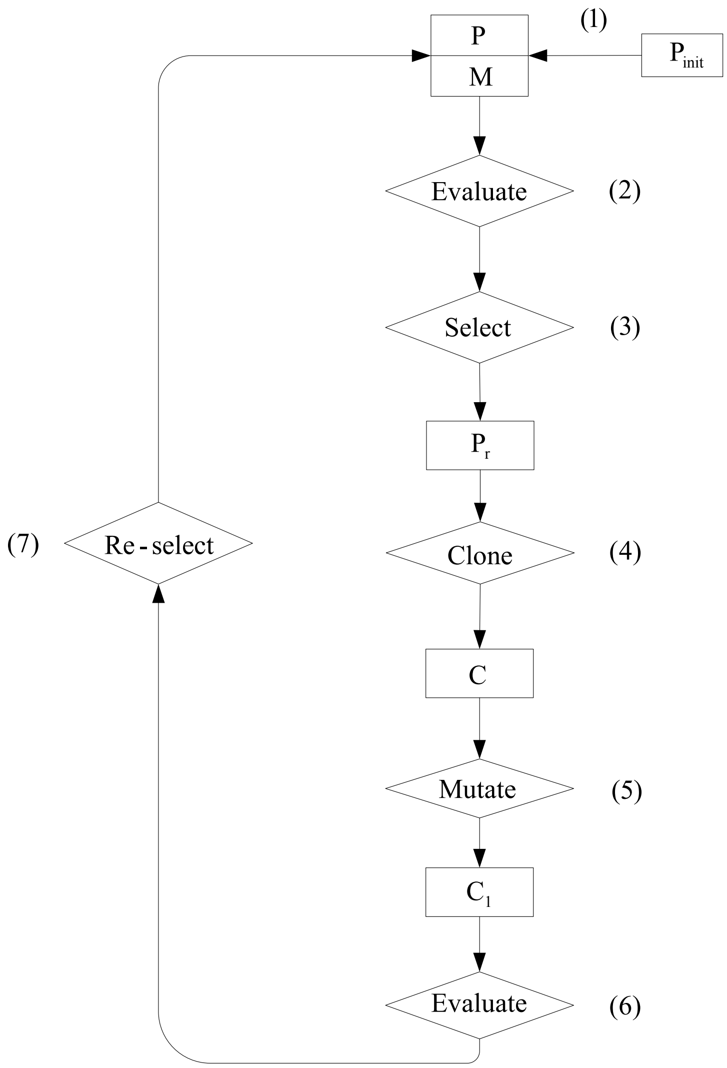

Fig. 3.3. Diagram of basic Clonal Selection Algorithm (CSA).

We should point out that the clone size in Step (4) is usually defined as a monotonic function of the affinity measure. With regard to the CSA mutation operator, the mutated values of the Abs are inversely proportional to their affinities. That is, the better fitness an Ab has, the less it may change. The idea of Abs suppression is also employed to eliminate the Abs with high self-affinities so that a diverse Ab pool is always maintained. In contrast to the popular GA [Wri06], which often tend to bias the whole population of chromosomes towards the best candidate solution, the CSA can effectively handle the demanding multi-modal optimization tasks.

Fukuda *et al.* propose a modified CSA-based immune optimization approach by merging the immune diversity, network theory, and CSA together. They explain that the main characteristics of this new method are the candidate diversity and efficient parallel search [Fuk99]. Proposed by de Castro and von Zuben, the CLONALG is one of the most widely applied CSA in practice [Cas02a]. The CLONALG utilizes the affinities of the evolving Abs. That is, the proliferation rates of the Abs are set to be proportional to their affinities. During the evolution of the CLONALG, those Abs with higher affinities are stimulated, while the ones with lower affinities are suppressed. Therefore, it can efficiently search in a diverse set of local optima to find even better solutions. de Castro and von Zuben demonstrate that compared with the GA in manipulating with the multi-modal and combinatorial optimization problems, the proposed CLONALG has a reduced computational cost.

3.2 Swarm intelligence

It is intriguing that some living creatures, e.g., ants, termites, and bees, exhibit collective behaviors despite the simplicity of the individuals that compose the swarm. Within the swarm, the relatively complex behavior is the result of the patterns of interactions among the individuals of the whole swarm over time. The main properties of this collective behavior can be summarized into four aspects, as shown in Fig. 3.4:

1. Nonsupervision: There is no single ‘leader’ to control the moving directions in the flock or hunting routes in the insect colony. The activity of any individual highly depends on its surrounding environment, which is furthermore influenced by the others.

2. Stigmergy: Cooperation and communication among individuals can be in the indirect contact form (changes made to the local environment)

3. Aggregation: Individuals trend to move towards the attractive local environment, which results in the dominant appearance of gathering.
4. Emergence: The term 'emergence' here refers to the process of deriving some new and coherent structures, patterns, and behaviors from complex systems. The group behavior of the swarm is not an inherent ability of any individuals, and is not easily predicted or deduced from the simple individual experiences.

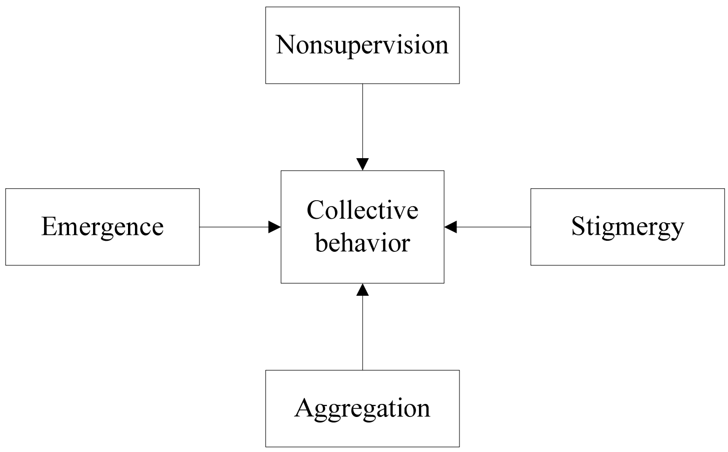

Fig. 3.4. Main features of collective behavior.

The collective behavior of unsophisticated agents interacting locally with their environment causes the coherent patterns of emergence [Ram05]. These swarming, flocking, and herding phenomena have promoted a popular NIC method, known as Swarm Intelligence (SI), which was firstly coined by Beny and Wang in the 1980s in the context of cellular robotics [Ben89] [Abr06]. The goal of the SI schemes is to model the simple behaviors of individuals as well as their interactions with the environment or neighbors so as to realize more advanced and complex techniques, which can be applied for coping with difficult optimization problems [Eng05].

The SI is generally classified into two main paradigms: Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). They are inspired by bird flocking and ant colony, respectively. The former mimics the behaviors of the individuals, whose moving directions are influenced by either the best neighbor or personal best experience record. The later models the pheromone trail-leaving and trail-following behaviors of ants, in which each ant perceives chemical pheromone concentrations in its local environment, and acts by probabilistically selecting directions based on the available pheromone concentration. In the following two sections, we shall discuss both the PSO and ACO in more detail.

3.2.1 Particle swarm optimization

As an important branch of the SI, the PSO draws its inspiration from some common properties existing in a flock of birds, school of fish, swarm of bees, and even human society. For example, the movement of bird flocking is an outcome of the individual efforts to maintain an optimal distance from their neighbors. This optimal distance makes the individuals match the velocity of the nearby flock mates without colliding. As a consequence, all the members in the flock are highly centralized. Swarming phenomena suggest the distinctive features of homogeneity, locality, collision avoidance, velocity matching, and flock centering. Thus, information exchange and sharing in the colony can provide an evolutionary advantage.

Inspired by the social behavior of the aforementioned flocking of birds and gathering of fish into schools, Kennedy and Eberhart propose the PSO method [Ken95]. In the original PSO, the position of each particle in the swarm represents a possible problem solution. The position and velocity of particle i at iteration n are denoted as x_{id}^{n} and v_{id}^{n}, respectively. The new velocity at the next iteration, v_{id}^{n+1}, is calculated using its current velocity v_{id}^{n}, the distance between the particle's best previous position p_{id}^{n} and x_{id}^{n}, as well as the distance between the position of the best particle in the swarm p_{gd}^{n} and x_{id}^{n}:

$$v_{id}^{n+1} = wv_{id}^{n} + c_1 r_1 (p_{id}^{n} - x_{id}^{n}) + c_2 r_2 (p_{gd}^{n} - x_{id}^{n}), \tag{3.1}$$

where w is the inertia weight, c_1 and c_2 are two positive constants, namely cognitive and social parameters, respectively, and r_1 and r_2 are two random values in the range of $[0,1]$. These deterministic and probabilistic parameters reflect the effects on the particle positions from both the individual memory and swarm influence. The position of particle i, x_{id}^{n}, is iteratively updated as follows:

$$x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1}. \tag{3.2}$$

Figure 3.5 shows the principle of particle update in the PSO. The optimal solutions can, thus, be acquired by choosing the best particles in the swarm after a certain number of iterations.

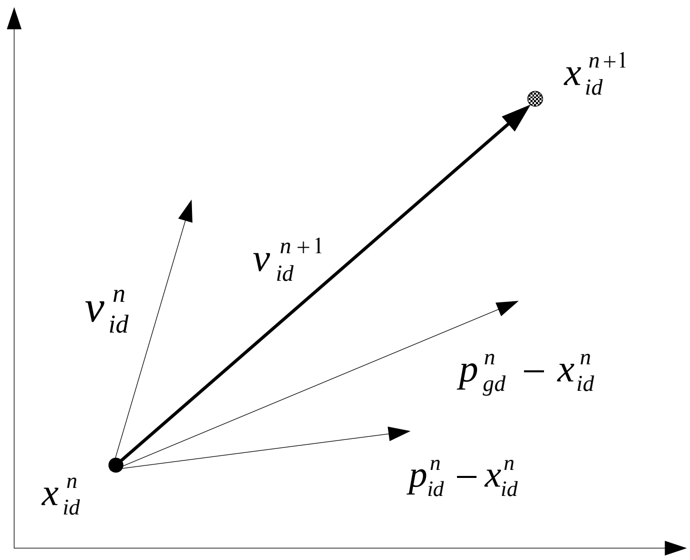

Fig. 3.5. Particle update in PSO.

The original PSO algorithm, as shown in Fig. 3.6, is summarized by the following five steps:

(1) Randomly choose the initial position and velocity within the given boundaries for each particle, and set the particle's best previous position p_{id}^{n} to be equal to the current position.

(2) Evaluate every particle in the swarm based on the objective function.

(3) Compare the current position with the particle's best previous position p_{id}^{n}. If the current position is better, update p_{id}^{n}.

(4) Move all the particles based on (3.1) and (3.2).

(5) If the preset stop criterion is met, terminate the PSO algorithm. Otherwise, return back to Step 2.

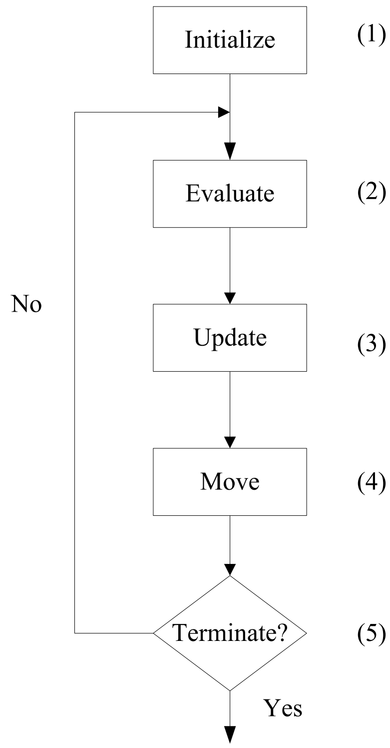

Fig. 3.6. Basic flowchart of original PSO.

Note that the PSO method has a well-balanced mechanism to efficiently utilize diversification and intensification in the search procedure, as demonstrated in Step 4 and (3.1) [Mon08]. Apparently, there are three velocity-related terms on the right-hand side of (3.1): current velocity, personal best velocity, and global best velocity. Even without the last two terms, the particles can still keep on exploring new areas by 'flying' in the same directions until they reach the given boundaries, which corresponds to the diversification in the search procedure. Intensification refers to the fact that all the particles try to converge to the p_{gd}^n and p_{id}^n. For this reason, the diversification can be appropriately weighted to emphasize the global search.

We can observe that during iterations, each particle learns from not only its own experiences but also the social behavior of the swarm, through which the search points gradually approach the optimum. In other words, taking advantage of the collective intelligence is a distinguishing property of the PSO method, which has been demonstrated to be efficient in solving various optimization problems [Par02]. However, empirical simulation experiments also show that the convergence speed of the

PSO often slows down with the growth of iterations, since stagnant particles may predominate the whole swarm [Ciu02]. Therefore, numerous improved PSO methods have been developed to overcome this drawback. For example, a fuzzy logic-based turbulence operation is used to control the velocities of all the particles [Liu07]. Zhan *et al.* deploy the crossover operator of the GA to maintain the swarm diversity and guide the particles to track the global optimum in nonlinear optimization [Zha07]. Parrott and Li divide the swarm into sub-populations based on the particle similarity so as to handle dynamic optimization problems [Par06].

3.2.2 Ant colony optimization

In recent years, collective behaviors including the foraging behavior of ants, mound construction of termites, nest-building of wasps, and web-weaving of spiders have been studied and have inspired another type of SI: the ACO. More precisely, when foraging, some ant species have the behavior of depositing a kind of chemical substance, called pheromone, through which they can communicate with each other to find the shortest path from their nest to the food source. These pheromone-leaving and pheromone-following phenomena lay a solid basis for the ACO, which is a stochastic, meta-heuristic, and population-based optimization algorithm.

The ACO can be characterized by the following [Eng05]:

- A probabilistic transition rule is used to determine the moving direction of each ant,
- Pheromone update mechanism indicates the problem solution quality.

Among the existing ACO algorithms, the Ant System (AS) is first proposed by Dorigo *et al.* for combinatorial optimization problems. Its main characteristic is that the pheromone values are updated by all the ants that have built a solution in the iterations [Dor06]. However, there are also other ACO algorithms that are targeted at continuous optimization problems. Bilchev and Parmee propose a continuous ACO approach for the local search in order to improve the quality of the solutions obtained by the GA [Bil95]. Wodrich and Bilchev extend this algorithm and introduce a bi-level search structure, in which both the local and global ants search and explore regions of the continuous functions, thus moving to the destinations with increased fitness by repeatedly searching locally and globally [Wod97]. These continuous ACO algorithms share the following features [Ho05]:

- Bi-level search functions (with both local and global search),
- GA used in global search.

Local search is indeed important in dealing with continuous optimization problems. The continuous ACO has been generalized to a hierarchical structure, in which the global search only aims at the 'bad' regions of the search space, while the goal of the local search is to exploit those 'good' regions. The basic ACO algorithm for continuous optimization (at each generation) is described as follows [Eng05]:

1. Create n_r global ants.

2. Evaluate their fitness.

3. Update pheromone and age of weak regions.

4. Move local ants to better regions, if their fitness is improved. Otherwise, choose new random search directions.

5. Update ants' pheromone.

6. Evaporate ants' pheromone.

Obviously, the continuous ACO is based on both the local and global search towards the elitist. The local ants have the capability of moving to the latent region with the best solution, according to transition probability $P_i(t)$ of region i:

$$P_i(t) = \frac{\tau_i(t)}{\sum_{j=1}^{g} \tau_j(t)}, \tag{3.3}$$

where $\tau_i(t)$ is the total pheromone at region i at time t, and g is the number of the global ants. Therefore, the better the region is, the more attraction it has for successive ants. If their fitness is improved, the ants can deposit the pheromone increment $\Delta\tau_i$, as in (3.4). Otherwise, no pheromone is left.

$$\tau_i(t+1) = \begin{cases} \tau_i(t) + \Delta\tau_i & \text{if fitness is improved} \\ \tau_i(t) & \text{otherwise} \end{cases} . \tag{3.4}$$

After each generation, the accumulated pheromone is updated as:

$$\tau_i(t+1) = (1-\rho)\cdot\tau_i(t) , \tag{3.5}$$

where ρ is the pheromone evaporation rate. We can conclude that the probability of the local ants selecting a region is proportional to its pheromone concentration. On the other hand, the pheromone is affected by the evaporation rate, ant age, and growth of fitness. Thus, this pheromone-based selection mechanism is capable of promoting the solution candidate update, which is suitable for handling the changing environments in optimization. The ACO is an efficient solution to a large variety of dynamical optimization problems, including routing, job assignment, and task scheduling [Dor06].

3.3 Simulated annealing

Based on the analogy between statistical mechanics and optimization, the SA is one of the most flexible techniques available for solving difficult optimization problems. The main advantage of the SA is that it can be applied to large-scale systems regardless of the conditions of differentiability, continuity, and convexity, which are usually required for conventional optimization methods [Fuk08]. The SA was originally proposed by Metropolis in the early 1950s as a model of the crystallization process. The SA procedure consists of first 'melting' the system being optimized at a high temperature, and then slowly lowering the temperature until the system 'freezes' and no further change occurs. At each temperature instant, the annealing must proceed long enough for the system to reach a steady state [Kir83]. The SA includes the following main features:

- Transition mechanism between states,
- Cooling schedule.

The SA method actually mimics the behavior of this dynamical system to achieve the thermal equilibrium at a given temperature. It has the remarkable ability of escaping from the local minima by accepting or rejecting new solution candidates according to a probability function. In addition, the SA method requires little computational resource. Fig. 3.7 illustrates the corresponding flowchart, and can be explained by the following steps:

1. Specify initial temperature T_0, and initialize the solution candidate.
2. Evaluate fitness E of the candidate.
3. Move the candidate randomly to a neighboring solution.
4. Evaluate the fitness of new solutions $E^{'}$.
5. If $E^{'} \leq E$, accept the new solution. If $E^{'} > E$, accept the new solution with acceptance probability P.
6. Decrease temperature T. The SA search is terminated, if T is close to zero. Otherwise, return back to Step 2.

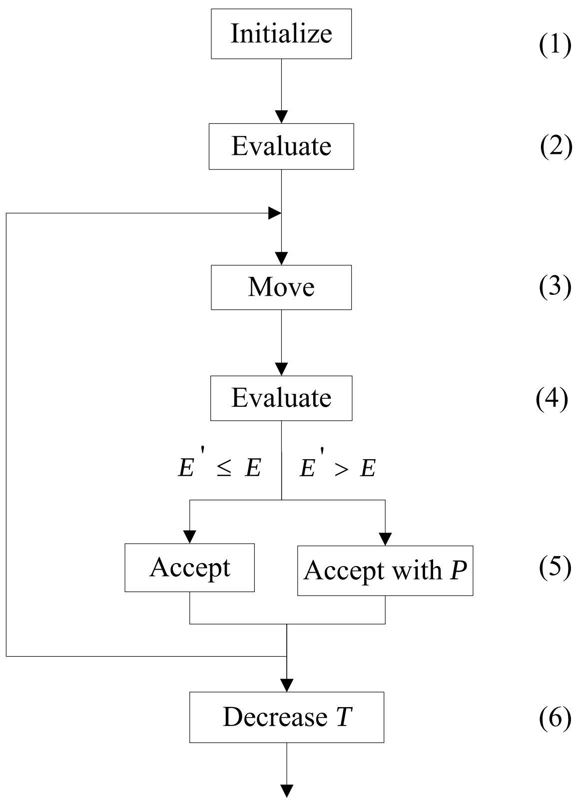

Fig. 3.7. Flowchart of basic SA method.

As we can observe that the SA algorithm simulates the process of gradually cooling a metal/crystal until the energy of the system achieves the global minimum. Each configuration of the physical system and energy of the atoms correspond to the current solution found for the optimization problem and fitness of the objective function, respectively. The temperature T is used to control the whole optimization procedure. At each generation, according to the Metropolis criterion [Sim06], the candidate is updated with the random perturbation, and the improvement of its fitness is also calculated. If $E^{'} \leq E$, the moving change results in a lower or equivalent energy of the system, and this new solution can be accepted. Otherwise, the displacement is only accepted with the probability P:

$$P = e^{\frac{-(E^{'}-E)}{T}}. \tag{3.6}$$

The temperature is updated by:

$$T(k+1) = \lambda T(k), \quad 0 < \lambda < 1, \tag{3.7}$$

where k is the number of generations, and λ is a given coefficient. As a matter of fact, the cooling schedule needs to be properly adjusted by modifying parameter λ. The main strength of the SA method is its capability of obtaining the global optimum with a great probability. However, it usually uses a large number of generations to converge.

3.4 Harmony search

When musicians compose harmony, they usually try various possible combinations of musical pitches stored in their memory. Such an efficient search for a perfect state of harmony is analogous to the procedure used for finding optimal solutions to engineering problems. Thus, harmony improvisation has inspired the emergence of a novel NIC approach, HS [Gee01]. Table 3.1 presents the comparison of harmony improvisation and optimization [Gee08]. Figure 3.8 shows a flowchart essentially describing the HS method that involves four principal steps.

Table 3.1. Comparison of harmony improvisation and optimization.

Comparison factors	Harmony improvisation	Optimization
Targets	Aesthetic standard	Objective function
Best states	Fantastic harmony	Global optimum
Components	Pitches of instruments	Values of variables
Process units	Each practice	Each iteration

Step 1. Initialize the HS Memory (HM). The HM consists of a number of randomly generated solutions to the optimization problems to be solved. For an n-dimension problem, an HM with a size of N can be represented as follows:

$$\mathrm{HM} = \begin{bmatrix} x_1^1, x_2^1, \cdots, x_n^1 \\ x_1^2, x_2^2, \cdots, x_n^2 \\ \vdots \\ x_1^N, x_2^N, \cdots, x_n^N \end{bmatrix}, \tag{3.8}$$

where $\left[x_1^i, x_2^i, \cdots, x_n^i\right]$ ($i = 1, 2, \cdots, N$) is a candidate solution. Note that the HM stores the past search experiences and plays an important role in the optimization performance of the HS method.

Step 2. Improvise a new solution $\left[x_1', x_2', \cdots, x_n'\right]$ from the HM. Each component of this solution, x_j', is obtained based on the Harmony Memory Considering Rate (HMCR). The HMCR is defined as the probability of selecting a component from the HM, and 1-HMCR is, therefore, the probability of generating it randomly. If x_j' comes from the HM, it is chosen from the j^{th} dimension of a random HM member, and it can be further mutated depending on the Pitching Adjust Rate (PAR). The PAR determines the probability of a candidate from the HM to be mutated. The improvisation of $\left[x_1', x_2', \cdots, x_n'\right]$ is similar to the production of offspring in the GA with the mutation and crossover operations [Pol02]. However, the GA usually create new chromosomes using only one (mutation) or two (crossover) existing ones, while the generation of new solutions in the HS method makes full use of all the HM members.

Step 3. Update the HM. The new solution from Step 2 is evaluated, and if it yields a better fitness than that of the worst member in the HM, it will replace that one. Otherwise, it is eliminated.

Step 4. Repeat Step 2 to Step 3 until a termination criterion is met.

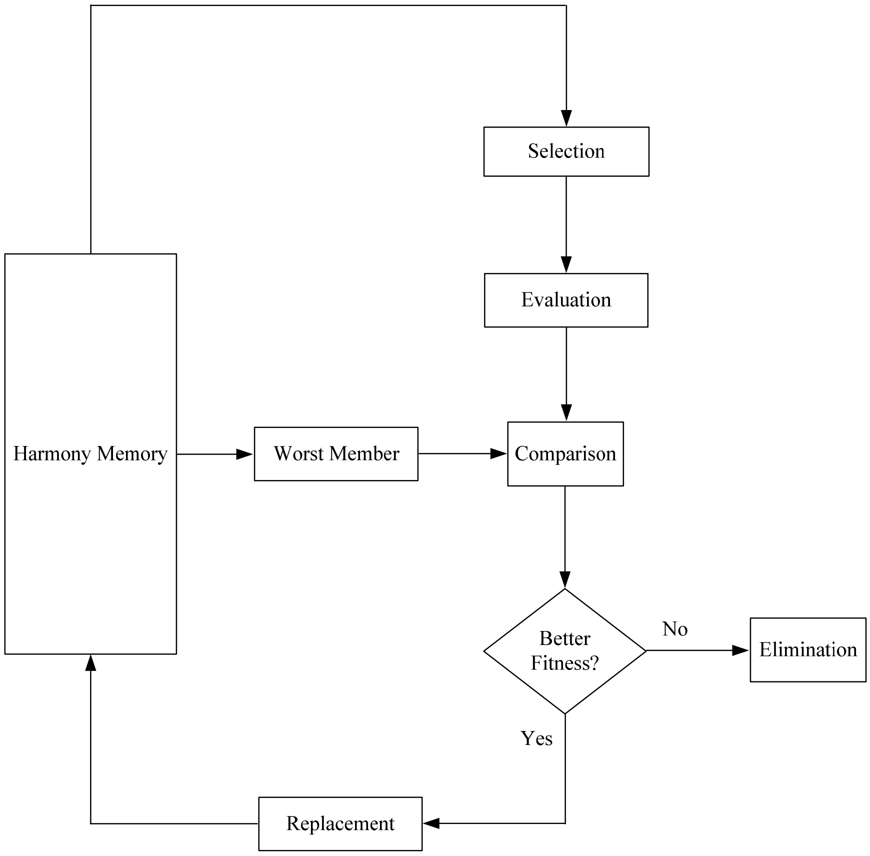

Fig. 3.8. Flowchart of HS method.

Similar to the GA and SI algorithms, the HS method is a random search technique. It does not need any prior domain knowledge beforehand, such as the gradient information of the objective functions. Nevertheless, different from those population-based approaches, it utilizes only a single search memory to evolve. Hence, the HS method imposes few mathematical requirements, and has the distinguishing advantage of computation simplicity. On the other hand, it occupies some inherent drawbacks, e.g., weak local search ability. Mahdavi *et al.* propose a modified HS method by using an adaptive PAR to enhance its optimization accuracy as well as speed up the convergence [Mah07]. To summarize, the features of multi-candidate consideration and correlation among variables contribute to the flexibility of the HS method, thus making it well suited for constrained optimal design problems [Kan04] [Gee02].

3.5 Differential evolution

The DE method is a simple but powerful population-based optimization technique first proposed by Storn and Price [Sto97]. The principle underlying DE is similar to that of other evolutionary computation methods, such as the GA. However, the uniqueness of DE is that it generates new chromosomes by only adding the weighted difference between two chromosomes to the third. In other words, unlike the GA, which rely on a predefined probability distribution function, the DE derives its mutation using the differences between randomly sampled pairs of the chromosomes [Cor99]. If the fitness of the resulting chromosome is improved, this newly generated chromosome replaces the original one. More precisely, suppose three chromosomes are under consideration in the current population: $r_1(k)$, $r_2(k)$, and $r_3(k)$. Note that $r_1(k)$ and $r_2(k)$ are randomly selected and mutually different. Figure 3.9 shows that a trial update of $r_3(k)$, $r_3'(k+1)$, is:

$$r_3'(k+1) = r_3(k) + \lambda[r_1(k) - r_2(k)], \tag{3.9}$$

where λ is a pre-determined weight. In order to further increase the diversity of the chromosomes, a 'crossover' operator is employed here to generate $r_3''(k+1)$ by randomly combining the parameters of $r_3(k)$ and $r_3'(k)$ together. If $r_3''(k+1)$ yields a higher fitness than $r_3(k)$, we get:

$$r_3(k+1) = r_3''(k+1). \tag{3.10}$$

Otherwise, $r_3''(k+1)$ is eliminated, and the above iteration procedure restarts until all the chromosomes have been successfully updated. The DE method has the following characteristics:

- Robustness,
- Inherently parallel structure,
- No need for a derivative,
- Few parameters to set,
- Simple processing,
- Reliability.

The DE method has been utilized as a global optimizer and a practical design tool for mixed variable optimization problems involving multiple and nonlinear constraints [Cor99].

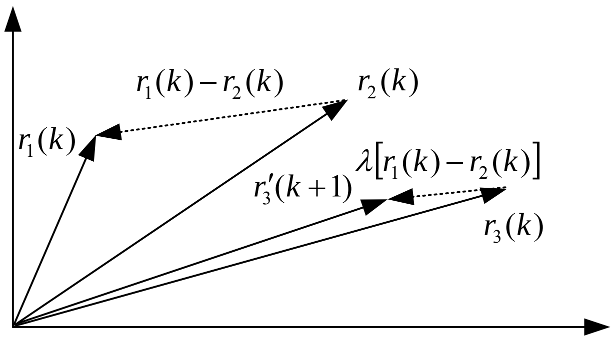

Fig. 3.9. Principle of DE method.

3.6 Mind evolutionary computing

The MEC is an evolutionary optimization approach developed by Sun in 1998 [Sun03a] [Jie06]. It is based on analysis of the human mind. As shown in Fig. 3.10, the whole population of chromosomes in MEC is divided into groups. Two billboards, local and global billboards, are used to store the evolution history. The global billboard can record winners in the global competition among the groups, while the local billboard is reserved for the winners among the individuals of each group in the local competition. The MEC employs two unique operations: similartaxis and dissimilation. In similartaxis, starting from their initial centers, individuals of every group compete against each other in local areas to become the winners, i.e., local optima. A group is considered to have matured, if no new winner appears there. The similartaxis actually serves as 'exploitation' in the MEC. On the other hand, the dissimilation is an 'exploration' process, in which individuals and groups compete to be the global winners in the solution space. The function of this operator is two-fold: (1) some best individuals are chosen as the initial scattering centers of the new groups; and (2) the current global optima are selected from the local optima of all the groups obtained by similartaxis. However, unlike the GA, there is no separate selection operation in MEC. In fact, the selection is implicitly employed in both similartaxis and dissimilation.

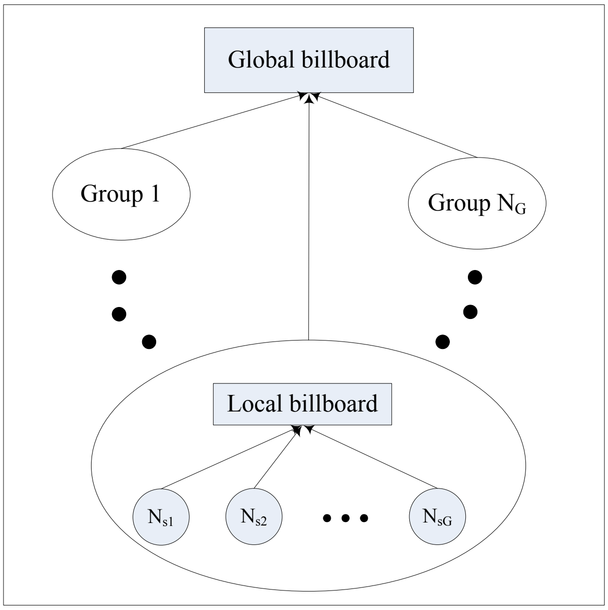

Fig. 3.10. Structure of MEC.

The iterative procedure of the basic MEC can be described as follows. Let S denote the population size, S_G the group size, and N_G the number of groups in the MEC, and $S = S_G N_G$.

Step 1. Generate S random chromosomes in the solution space, and select N_G individuals from these as the initial scattering centers for the N_G groups.

Step 2. Perform the similartaxis on these groups, i.e., for every group, $S_G - 1$ chromosomes are scattered based on a preset probability density function around the group center. The resulting S_G individuals are next evaluated and compared with each other, and a local winner is chosen as the new group center for the next generation. Update the local billboard by recording the local winners on it. The above process is repeated until all the groups are matured.

Step 3. Select the best solutions from all the local optima (winners) obtained in Step 2. Store them on the global billboard, and expunge a certain number of poor chromosomes from the billboard.

Step 4. If the given optimization criterion is satisfied, terminate the MEC. Otherwise, return back to Step 1.

Thus, Step 1 initializes the appropriate scattering group centers. Step 2 implements the similartaxis operation and serves as the local competition, in which a local optimal solution is located on the basis of the chromosomes starting from the scattering center of each group. Step 3 evaluates these local optima acquired through similartaxis and updates the global billboard. Steps 1 and 3 act together as the aforementioned dissimilation operation. Further details and some variants of the MEC can be found in [Sun03a]. Due to the contributions from the similartaxis and dissimilation employed, the MEC has been shown to outperform the GA in nonlinear multi-dimensional function optimization [Sun03b]. Guo *et al* propose a modified MEC by introducing 'forbidden zones' on both the global and local billboards so as to improve the overall search efficiency. In addition, they also explore the application of this new MEC in the optimization of PID controllers [Guo04].

3.7 Summary

This chapter has discussed a total of seven stochastic, iterative, and population-based NIC methods, which require little prior information concerning the optimization problems to be solved. Apparently, these algorithms share two common characteristics in mimicking natural phenomena:

- Inspiration drawn from nature,
- Modeling of natural processes.

The NIC techniques usually have a general flowchart, as shown in Fig. 3.11 [Cor99]. First, randomly generate an initial collection of candidate solutions (Initialize). After that, produce new members by making changes to the selected candidates, and then evaluate them (Operate). The changes may involve merging two or more existing members together or introducing random variations to the current candidates. Finally, replace those solutions that are outdated with the improved ones (Renew). Such a competitive strategy results in appropriate candidates for generating new solutions in the next generation. Hence, the key ideas underlying most NIC approaches are candidate generation, evaluation, selection, and update.

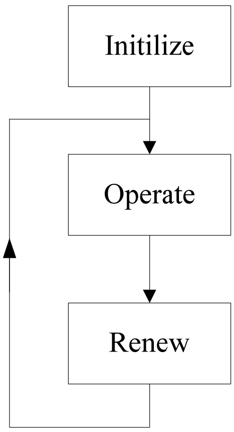

Fig. 3.11. General flowchart of NIC algorithms.

We can conclude that the above NIC methodologies share many similarities, e.g., adaptation, learning, and evolution. On the other hand, they also have some distinct differences, and each has its own advantages and drawbacks [Wol97]. Table 3.2 summarizes the advantages and shortcomings of the NIC algorithms introduced in this chapter. For example, CSA and ACO can deal with multi-modal and dynamical optimization problems, respectively. Although they have attracted considerable research attention, due to their outstanding performance compared to the conventional optimization solutions discussed in Chapter 2, the standalone NIC methods are still not efficient enough at handling uncertainty and imprecision in practice. Additionally, practicing engineers often face the difficulty of choosing the most suitable NIC methods to meet particular engineering requirements. In the next chapter, we will present and investigate several hybrid optimization schemes, which are based on the fusion of these NIC algorithms and can overcome their individual weaknesses.

Table 3.2. Advantages and disadvantages of different NIC algorithms.

NIC algorithm	**Advantages**	**Disadvantages**
ACO	Pheromone-based elitism	Over-similarity
CSA	Diversity	Slow convergence speed
DE	Effective search	High computational effort
HS	Algorithm simplicity	Outdated information
MEC	Anti-premature	Time-consuming
PSO	Information sharing	Premature
SA	Robustness	Long computation time

4. Hybrid Nature-Inspired Optimization Methods

It is well known that CSA, PSO, ACO, SA, DE, HS and MEC, discussed in the previous chapters, represent typical NIC schemes that have found successful applications in numerous engineering areas. However, all the NIC methods have their own strengths and drawbacks, since they are based on only certain phenomena in nature. Over the past decade, hybridization of the NIC algorithms has gained significant popularity, thus helping to overcome the individual drawbacks while benefiting from each other's advantages. Therefore, fusion of the NIC methods, e.g., combination of ACO and CSA, can offer us competitive solutions with improved performance for challenging optimization problems. The inspiration and fusion of these NIC techniques are shown in Fig. 4.1.

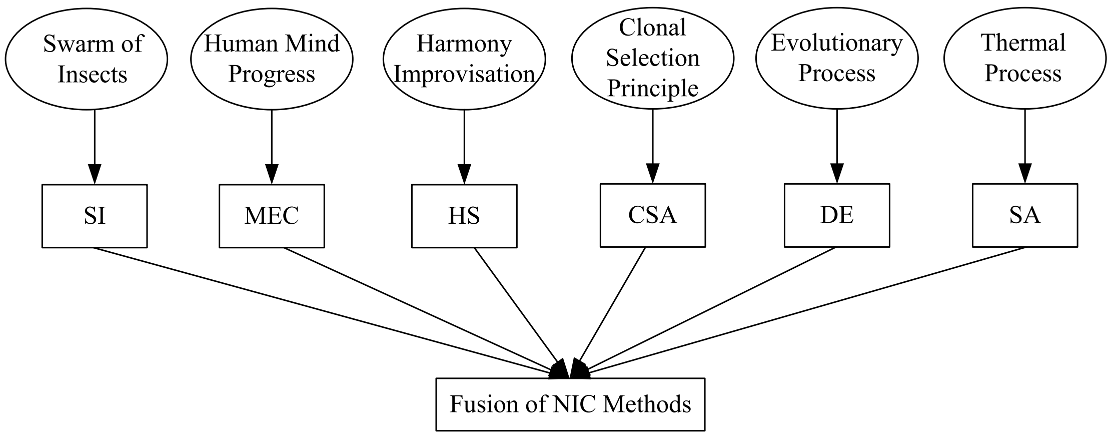

Fig. 4.1. Inspiration and fusion of typical NIC methods.

4.1 Hybridization taxonomy

Numerous hybrid NIC optimization algorithms have been proposed and studied in the literature [Gro05] [Sin03] [Yen95]. In this section, we classify these fusion strategies into two main types of hybridization: motivation for hybridization and architecture of hybridization.

4.1.1 Motivation of hybridization

The capability of overcoming the shortcomings of individual algorithms without losing their advantages makes the hybrid techniques superior to the stand-alone ones. Based on the dominant purpose of hybridization, the hybridization of the NIC methods can be divided into two types:

- Exploitation: In this type of hybridization, after an NIC algorithm has been used to the search for promising regions in the solution space, another one is next employed in the local search to further prompt the convergence to the global optimum. In other words, these hybrid methods actually perform a hierarchical optimization strategy, in which the local search component refines the 'rough' solutions obtained by the global search partner.
- Parameter optimization: This class consists of those hybridization approaches that utilize one NIC algorithm as the 'secondary' method to optimize the parameters of another. For example, the GA can be merged with fuzzy logic for the optimal generation of membership functions and fuzzy reasoning rules [Ish95].

4.1.2 Architecture of hybridization

As shown in Fig. 4.2, hybrid NIC algorithms can also be classified into three categories according to the nature of their architectures.

- Preprocessors and postprocessors (Fig. 4.2 (a)): This is the most popular hybridization type, in which the NIC techniques are applied sequentially, i.e., data/information generated by Algorithm A (preprocessor) can be fine-tuned by Algorithm B (postprocessor). For instance, Grosan *et al*. propose a fusion of two modified NIC methods, Independent Neighborhoods Particle Swarm Optimization (INPSO) and Geometrical Place Evolutionary Algorithms (GPEA), for solving difficult geometrical place problems [Gro05]. The proposed hybrid INPSO-GPEA starts with the INPSO and switches to the GPEA after a given number of iterations, e.g., 100 iterations. Actually, all the initial chromosomes used by the GPEA represent particles existing in the INPSO. Thus, the candidate solutions that have failed earlier with the INPSO are expected to converge using the additional iterations of the GPEA. In [P5], the CSA is employed to improve the fitness of the solution candidates in the HM. In other words, all the members of the HM are regarded as the individual antibodies, and they can evolve in the population of the CSA.

- Cooperators (Fig. 4.2 (b)): Figure 4.2 (b) illustrates such a hybrid system, in which the two algorithms involved simultaneously adjust each other. Common information is exchanged and shared between the algorithms during the search process. For example, the evolutionary computation method and fuzzy inference technique are fused together in a hybrid evolutionary-fuzzy system for nonlinear function approximation. These two algorithms can optimize, fine-tune, and control the parameters of each other [Abr08] [Her95].
- Embedded operators (Fig. 4.2 (c)): The hybrid NIC methods belonging to this class are characterized by their architectures, in which Algorithm B is embedded inside Algorithm A. A typical approach is to combine the local search and global search from different NIC techniques together in order to improve the convergence of the hybridization. For example, in [Gam97] [Sol02], the heuristic local search strategies, such as Tabu search [Glo97], are employed in the ACO for early detection of high-quality solutions, which can then be used either as the basis for generating new solutions or to refine the pheromone concentration. The update of the ants in their hybrid ACO is based on a joint contribution from both the local and global search operations. In [P2], the dissimilation and similartaxis in the CSA, which is embedded in the MEC, are for the global and local optimal search, respectively.

The hybrid NIC optimization methods for dealing with the nonlinear and dynamical problems are discussed in Sections 4.2 and 4.3, respectively.

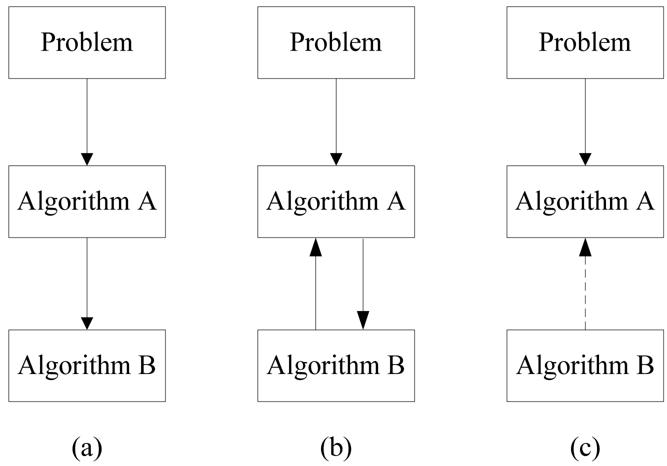

Fig. 4.2. Architectures of the hybrid NIC algorithms.

4.2 Hybrid nature-inspired optimization methods for nonlinear problems

In this section, we summarize and review a few representative hybrid nature-inspired optimization schemes for the nonlinear problems. Based on the NIC techniques involved, they can be classified into SI-based hybridization, HS-based hybridization, SA-based hybridization, and AIS-based hybridization as follows.

4.2.1 SI-based hybridization

The SI algorithms, including PSO and ACO, have the advantage of quickly locating good but approximate solutions. However, they may converge prematurely to give a relatively poor solution. To address this problem, the ACO has been fused with other NIC methods to form an efficient local search approach. Indeed, the manner in which the local search is embedded and utilized is one of the most important issues to be considered in an ACO-based hybrid system [Eng05]. Bilchev and Parmee propose a hybrid optimization algorithm, in which the ACO method is merged with a GA for the refinement search so as to improve the quality of the final solution of the GA [Bil1996]. It is further improved and used in the application of nonlinear electromagnetic devices design by Ho *et al* [Ho05]. The feature common to these hybrid ACO methods is that the continuous region can be divided into several regions, which act as the local 'stations' for the ants to move into and explore [Mat00]. Therefore, such a combination of local and global search procedures makes the hybrid algorithms efficient in optimization.

In the PSO, the behaviors of a particle as well as its neighbors are directly influenced by the global best particle, which could result in the harmful clustering and premature convergence. Therefore, many PSO variations have been developed by combining the characteristics from the GA and PSO. Shi *et al*. propose the execution of both the PSO and GA in parallel. With their hybridization, the best solutions are exchanged between the two populations of the particles and chromosomes after a predetermined number of iterations [Shi03]. In QPSO, Pant *et al.* have suggested a quadratic crossover operator [Pan07]. This nonlinear multi-parent crossover operation makes use of three particles (parents) in the swarm to produce another particle (offspring), which lies at the point of the minimum of the constructed quadratic curve passing through these three selected particles. The new particle is accepted into the swarm, irrespective of whether it might be better or worse than the present worst particle. Hence, the PSO search is not only limited to those regions around the current best location but is, in fact, more diversified. In [P2], a hybrid NIC algorithm is proposed by combining the PSO method with the CSA and MEC. This hierarchical scheme enhances both the exploitation in the local space and exploration in the global space, and can effectively manipulate with the aforementioned premature problem of the regular PSO.

4.2.2 HS-based hybridization

The HS has been shown to be powerful in identifying high-performance regions in the solution space. Unfortunately, it performs poorly in the local search. Omran and Mahdavi propose an improved HS, PSO-HS, by modifying the PAR and pitch-adjustment step [Omr08]. The idea of swarm learning is also incorporated into the original HS, thus allowing a new harmony to mimic the best harmony in the HM. This modification actually alleviates the problem of tuning the HS parameter, bw (distance bandwidth), which is difficult to specify *a priori*, and is usually based on trial and error. Their PSO-HS works efficiently for both continuous and discrete optimization problems.

Cruz and Coelho introduce the elitism of the GA and probability threshold strategy inspired by the SA to the regular HS method [Cru08]. This new meta-heuristic algorithm can offer competitive results in certain specific application fields. Li and Li embed the HS method as a global search to improve the optimization performance of the GA [Li07]. In their Novel Hybrid Real-Value Genetic Algorithms (NHRVGA), the GA chromosomes are considered as the HM members, and are updated using the HS method, which gives the NHRVGA a strong exploitation capability. Simulations demonstrate that this NHRVGA performs better than the real-valued GA in solving different benchmark problems, even those with as many as 30 dimensions. Combining the advantages of HS and fuzzy C-means analysis, Malaki *et al.* incorporate a fuzzy approach into the HS with a Fuzzy Harmony Search Clustering (FHSClust) as the pre-processing tool for the fuzzy C-means analysis [Mal08]. The hybrid technique utilizes the fuzzy-improved HS to locate the promising places of the global optimum, and feeds the results of the FHSClust to the fuzzy C-means, thus enabling the global optimal solution to be found rapidly.

The solution quality of the HS method highly depends on the harmony memory pool, which may limit the global exploration ability. In [P5], the CSA is used to optimize the harmony memory so as to keep it more diverse and efficient. This HS-CSA algorithm shows superior performance in the optimization of fuzzy classification systems. In [P4], we propose two modified HS methods to deal with uni-modal and multi-modal optimization problems. The first modified HS method is based on the fusion of the HS and DE techniques. Similar to [P5], the DE is employed to optimize the HS memory members. The second modified HS method utilizes a novel HM management approach that aims at handling multi-modal problems. In summary, some NIC methodologies have been deployed for refining the HM in order to accelerate the convergence of the hybrid HS methods.

4.2.3 SA-based hybridization

The SA can enhance the global search capability of hybrid NIC optimization methods. Mohammed *et al.* discuss an improved SA approach, in which the search control and state update procedures are separated [Moh93]. It is further used together with a

Hopfield neural network to optimize the solutions obtained from a trained multilayered perceptron. This new hybrid NIC method has been applied to the optimal design of electromagnetic devices and has yielded improved parameters compared to the conventional optimization method.

Kuo develops a novel SA-PSO to minimize the total generation costs of a power system over an appropriate period of time within various given constraints [Kuo08]. In his algorithm, the movements (velocity update) of all the particles are first generated by the PSO algorithm and are next combined with an SA judgment operator. The metropolis process of the SA calculates whether the determined movements are accepted or not, according to a temperature-controlled probability. Compared with the PSO and GA, this SA-PSO can obtain more efficient and higher-quality solutions with reduced CPU time.

Vasconcelos *et al.* propose a Modified SA (MSA) method by combining the SA with the Tabu search for the optimal design of electromagnetic [Vas96]. In the MSA, the neighborhood search of the SA is restrained by the Tabu list, thus enabling those local optima encountered to be escaped from. This MSA has been shown to be a simple but powerful optimizer for complex global optimization problems. In [P7], the SA is embedded into the CSA to enhance its global search ability. Our hybrid algorithm is applied to deal with the optimization of several nonlinear benchmark functions as well as a practical engineering design problem: pressure vessel design. Numerical simulations have demonstrated that it is superior to the regular CSA with regard to optimization efficiency.

4.2.4 AIS-based hybridization

Hajela and Lee argue that AIS, which are capable of performing schema recognition and adaptation [Cas00], should be used advantageously to improve the performance of the GA in structural optimization problems [Haj99]. Their hybrid model can enhance the convergence of a GA approach and handle the design of constraints in GA-based optimization. Coello and Cortes have developed Hajela's algorithm into a parallel version and have examined it using a larger problem set [Coe04]. Furthermore, an extension proposed by Bernardino *et al.* divides the whole chromosome population into feasible and infeasible individuals, which are optimized by the GA and AIS, respectively [Ber08]. Compared with four traditional methods, the proposed AIS-GA method achieves the best solutions with the probability of 57.14% in five constrained optimization problems.

Hou *et al.* integrate the Artificial Neural Networks (ANN) with the AIS to optimize the parameters in an IC wire bonding process [Hou08]. In their approach, the AIS consist of memory cells and suppressor cells that store the candidate solutions with the best Ab-Ag and Ab-Ab affinities, respectively. Using this approach, the memory cells can improve the evolution procedure of the Abs, and the role of the suppressor cells is to prevent the AIS from revisiting those regions that have been previously searched. The modified CSA is applied to find the optimal wire bonding process parameters using the output of the ANN as the affinity measure. Simulations show that it takes about 50 iterations (201 seconds in calculation) for the proposed optimization method to obtain the optimal solution, but 200 iterations (390 seconds) for the GA.

In [P1], the CSA is embedded into the MEC to construct a hybrid optimization method. The convergence speed of the CSA is improved by the MEC dissimilation operation, which can keep the candidate pool dynamic during iterations as well as explore more feasible solution space. In [P6], based on the fusion of the CSA and DE method, we propose a novel optimization scheme: CSA-DE. The DE is applied in order to increase the affinities of the Ab clones in the CSA. In other words, the employment of the CSA in these hybrid approaches can considerably enhance their optimization capability.

4.3 Hybrid nature-inspired optimization methods for dynamical problems

Real-world optimization problems are often dynamical with objective functions that change over time. The NIC methods are adequate solutions to these dynamical optimization problems, due to their distinguishing adaptation characteristics. For example, Ursem employs a self-organizing Multi-national GA (MGA) for a dynamical optimization that structures the GA population into sub-populations based on the detection of valleys in the fitness landscape [Urs00]. On the assumption that one of the local peaks on the multi-modal landscape may rise to become the global optimum because of changes in the environment, this multi-modal-based MGA can be deployed for manipulating dynamical optimization problems. Branke *et al.* use a forking GA to enhance its search ability under dynamic environments by having a number of smaller populations to track the most promising peaks and only a larger parent population to continuously search for new peaks [Bra00].

Blackwell and Bentley have developed a charged PSO method based on the analogy of electrostatic energy and charged particles [Bla02]. The idea is to introduce two opposing forces within the dynamics of the PSO: an attraction to the mass center of the swarm and an inter-particle repulsion. More precisely, the attraction force facilitates the PSO convergence to a single solution, while the repulsion force preserves particle diversity. Meanwhile, the neutral swarm can continue to explore the neighborhood of the optimum [Bla06].

Proposed by Blackwell and Branke, the atomic swarm method has been demonstrated to be adaptive for tracking multiple optima simultaneously with multiple swarms [Bla04]. In their atomic swarm scheme, the number of the swarms is set beforehand. When two swarms approach within a specified radius of each other, the swarm with the worse value at its attractor or global best position is randomized. In this way, the multiple swarms are prevented from converging to the same peak. The atomic swarms are also modified to form quantum swarms [Bla04]. The authors have simplified the above idea and replaced the charged particles with quantum particles that can move to random positions around the global best particle based on quantum computing principles [Han02].

Ramos *et al.* have developed another interesting self-regulating swarm algorithm, which merges the advantageous characteristics of swarm intelligence with evolutionary computation [Ram06]. The social environmental memory and cognitive map via the collective pheromone laid on the landscape are used to properly balance the exploration and exploitation nature of this hybrid search strategy. In addition, a simple evolutionary mechanism through a direct reproduction procedure linked to the local environmental features can self-regulate the exploratory swarm population in order to accelerate its global search speed.

Pulkkinen *et al.* combine the GA with SA in dealing with the dynamic combinatorial optimization problems of thermo-mechanical pulp production scheduling [Pul06]. In their hybrid algorithm, a population of candidate solutions first undergo a random walk in the search space, and then reproduce themselves. Using this approach, the GA phase takes large leaps in the search space, whereas the SA phase refines the solutions through a local search.

Most of the current research work in the dynamical optimization has concentrated on those swarms that can track a single optimum. In [Par06], the authors propose a Dynamical Species-based PSO (DSPSO) to simultaneously track multiple optima, in which the swarm population is divided into species sub-populations on the basis of their similarity. The species are located using feedback from the multi-modal fitness landscape, and they can guide the sub-populations to adaptively approach the multiple optima. A crowding mechanism is also implemented in the DSPSO to give the swarm a remarkable ability to track the dynamic optima resulting from environmental changes.

In general, for multi-population-based dynamical optimization approaches, the distribution of the individuals (diversity of solution candidates) in the search space has a crucial effect on multi-modal optimization performance. In [P8], we merge the pheromone-based elitism of the ACO with the solution diversity of the CSA in a hierarchical search scheme. The proposed hybrid algorithm has been employed to effectively tackle multi-modal problems under different time-changing environments. However, it should be pointed out that the ACO method in [P8] is a modified version of the original ACO, in which the foraging ants with their pheromone are considered as the potential solutions.

4.4 Summary

In this chapter, we have first discussed a general classification for the hybridization of the NIC methods. The existing hybrid NIC schemes are classified based on either the ‘motivation for hybridization’ or ‘architecture of hybridization’ of the techniques involved. However, we can also classify these schemes according to the targeted problems to be solved, e.g., multi-modal and dynamical optimization problems. A brief overview of some typical hybrid NIC optimization methods with applications is next presented. The common hybridization principle is that two or even more different NIC techniques are combined together, aiming at reinforcing their strengths and overcoming the weaknesses. The relationship among the NIC methods can be competitive or cooperative. We here only focus on the ability of the hybrid strategies in attacking nonlinear and dynamical optimization problems. Compared with stand-alone NIC techniques, the hybridization of these techniques has been shown to yield superior optimization performance. However, we need to emphasize that merging various types of NIC methods might increase the overall computational complexity of these hybrid approaches. Therefore, a trade-off must be made in choosing the appropriate hybridization in order to meet practical engineering needs.

5. Summary of Publications

In this chapter, we present a summary of the publications comprising the dissertation. Publications [P1] and [P2] discuss two new CSA-based optimization methods with application to optimal power filter design. Publications [P3]-[P6] concentrate on employing the fusion of various NIC techniques in solving nonlinear optimization problems, including the optimization of nonlinear functions, neural networks and fuzzy classification systems. Two hybrid immune-based approaches for constrained and dynamic optimization problems are proposed in [P7] and [P8], respectively.

5.1 [P1]

X. Wang, "Clonal selection algorithm in power filter optimization," *IEEE Mid-Summer Workshop on Soft Computing in Industrial Applications*, Espoo, Finland, June 2005.

In [P1], a novel optimization scheme inspired by the CSA is applied to the design of a LC passive filter in a diode full-bridge rectifier. The optimization aims to obtain the optimal values of an inductor and a capacitor. Simulations demonstrate that the proposed CSA-based power filter design approach is capable of optimizing the given criteria, i.e., PF and THD. Compared with other LC filter optimization methods, the CSA can avoid being trapped into local optima and provides more practical design choices, due to its diverse solution candidate pool. It has been proven to be an effective and flexible optimization method for handling challenging engineering problems.

The author is fully responsible for carrying out the research work including proposing the optimization method, implementing the CSA in the optimization of the power filter parameters, and document writing.

5.2 [P2]

X. Wang, X. Z. Gao, and S. J. Ovaska, "A hybrid optimization algorithm in power filter design," *31st Annual Conference of the IEEE Industrial Electronics Society*, Raleigh, NC, November 2005.

In [P2], we improve the performance of the regular CSA by combining it with the MEC. The nonlinear function optimization and optimal selection of power filter parameters (the same as in [P1]) are the two test-beds used. The convergence speed of the CSA is relatively slow. However, in our hybrid optimization algorithm, the MEC-based dissimilation operation keeps the Ab pool dynamical to explore larger regions in the solution space during iterations. The Abs diversity maintenance capability of the CSA and anti-premature function of the MEC are fully utilized in the proposed method, thus allowing it to evolve with a smaller size of population, while still achieving moderately better optimization performance.

S. J. Ovaska introduced the passive filter parameters optimization problem to the author, and X. Z. Gao suggested the basic scheme for solving this problem. The author designed and implemented the hybrid optimization model, which combines the CSA and MEC together.

5.3 [P3]

X. Wang, X. Z. Gao, and S. J. Ovaska, "A novel particle swarm-based method for nonlinear function optimization," *International Journal of Computational Intelligence Research*, vol. 4, no. 3, 2008.

It is well known that premature is the main disadvantage of the original PSO in solving demanding optimization problems. To overcome this drawback, we study a hybrid PSO method in [P3], which is based on the fusion of PSO, CSA, and MEC. Both the cloning function borrowed from the CSA and MEC-characterized similartaxis and dissimilation operators are embedded in the PSO. Therefore, the information sharing of the PSO, the solutions diversity of the CSA, as well as the anti-premature of the MEC are combined together. Simulation results show that our hybrid PSO method can yield enhanced optimization performance.

The author proposed and studied the idea of employing the PSO and CSA to enhance the search ability of the MEC, as well as implemented this idea using MATLAB. X. Z. Gao and S. J. Ovaska contributed to the work through their valuable comments and discussions.

5.4 [P4]

X. Z. Gao, X. Wang, and S. J. Ovaska, “Uni-modal and multi-modal optimization using modified harmony search methods,” *International Journal of Innovative Computing, Information and Control*, in press.

In [P4], we propose two modified HS methods to deal with the uni-modal and multi-modal problems. With the fusion of HS and DE, a hybrid optimization scheme, HS-DE, is first discussed. The HM members are fine-tuned by the DE to improve their affinities, and achieve better optimization behaviors. In the second modified HS method, we deploy a fish swarm-based technique to maintain the diversity of the HM members, thus making it a suitable candidate for locating multiple optima. Several simulation examples of the uni-modal and multi-modal functions have been employed to verify the effectiveness of these two new HS methods.

X. Z. Gao proposed the hybrid DE-HS algorithm for uni-modal optimization. The author cooperated with him in designing the modified HS method and using it to deal with the multi-modal optimization problems. This work was carried out under the instruction and supervision of S. J. Ovaska.

5.5 [P5]

X. Wang, X. Z. Gao, and S. J. Ovaska, “Fusion of clonal selection algorithm and harmony search method in optimization of fuzzy classification systems,” *International Journal of Bio-Inspired Computation*, vol. 1, no. 1-2, 2009.

In [P5], we present a hybrid optimization method on the basis of the CSA and HS technique. The HS memory solely depends on past search experiences, which may limit the optimization ability of regular HS. In our novel approach, the CSA is applied to improve the fitness of the solution candidates stored in the HM. That is to say, all the members of the HM are regarded as individual Abs and can evolve in the population of the CSA. This hybrid optimization algorithm is further used to optimize Sugeno fuzzy classification systems for classification of the Fisher Iris data and wine data. Computer simulations demonstrate that our systems can result in better classification performance with fewer fuzzy rules than that optimized with the original CSA and HS.

The author proposed the improved HS algorithm, which employs the CSA to maintain the diversity of the harmony memory. X. Z. Gao suggested using this hybrid method to optimize the fuzzy classification systems. S. J. Ovaska provided some advice on improving the proposed optimization approach.

5.6 [P6]

X. Z. Gao, X. Wang, and S. J. Ovaska, "Fusion of clonal selection algorithm and differential evolution method in training cascade-correlation neural network," *Neurocomputing*, in press.

In [P6], by merging the CSA and DE together, we propose a hybrid optimization method, CSA-DE. The CSA Abs are fine-tuned by the DE to increase their affinities in order to obtain improved optimization performance. We also discuss the application of the CSA-DE in the optimal construction of a Cascade-Correlation (C-C) neural network, which is an adaptive self-growing feedforward neural network. Two numerical examples, nonlinear function optimization and C-C neural network training, have been used to explore the efficiency of the proposed method. Compared with the back-propagation learning algorithm, the CSA-DE-based approach leads to a C-C neural network with less hidden nodes.

The author and X. Z. Gao together proposed this CSA-DE-based approach. X. Z. Gao further implemented and used it in nonlinear function optimization and C-C neural network optimal training. S. J. Ovaska provided his valuable comments on improvements to the algorithm.

5.7 [P7]

X. Wang, X. Z. Gao, and S. J. Ovaska, "A simulated annealing-based immune optimization method," *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*, Porvoo, Finland, September 2008.

In [P7], we develop a hybrid optimization algorithm using the principles of both the CSA and SA. The SA method often needs a considerably long time to acquire the global optimum, since the temperature has to be decreased slowly enough during the cooling procedure. In our strategy, fitness-related mutation and cloning, as well as affinity-based self-suppression of the CSA are deployed and integrated with the SA so as to improve global search and convergence speed. This new scheme is tested with several typical nonlinear optimization problems. Experiments have shown that the proposed hybrid technique is superior to certain existing algorithms in optimal pressure vessel design and can provide diverse, flexible solutions to the multi-modal problems.

The author proposed this hybrid optimization approach and applied it to nonlinear and constrained optimization. X. Z. Gao suggested the idea of merging the CSA together with the SA to achieve enhanced search performance. S. J. Ovaska provided his suggestions for algorithm improvement in solving the constrained optimization problems.

5.8 [P8]

X. Wang, X. Z. Gao, and S. J. Ovaska, "A hybrid optimization algorithm based on ant colony and immune principles," *International Journal of Computer Science and Applications*, vol. 4, no. 3, 2007.

In [P8], we investigate a hybrid optimization algorithm that combines the distinguishing features of ACO and CSA, i.e., pheromone-based elitism selection, fitness- and age-dependant mutation size, hierarchical search structure, and self-suppression. Especially, at each iteration, only those ants that can increase performance deposit an amount of pheromone proportional to their improved fitness. Thus, this hybrid optimization approach is capable of weeding outdated candidates without losing the best ones. It is also adaptive to the time-varying environment to avoid misleading the search directions of the ants. Its validity is examined in the optimization of several nonlinear functions under static and dynamic environments.

The author and X. Z. Gao proposed the fusion of the ACO and CSA to fully utilize the advantages, thus making the resulting hybrid optimization algorithm adaptive to both static and time-varying environments. S. J. Ovaska suggested this new NIC method for dynamical optimization.

6. Conclusions

6.1 Summary and scientific importance of author's work

Optimization methods can be generally classified into two major categories: deterministic and stochastic techniques. Unlike deterministic optimization methods, the stochastic approaches involve useful randomness in their search procedures to avoid becoming stuck in local optima. Stochastic approaches have been shown to be better than conventional schemes in handling various types of challenging problems by providing globally optimal, robust, flexible solutions. Inspired by certain natural phenomena, the NIC optimization methods form a new emerging type of stochastic approaches.

This dissertation has presented a brief introduction to a few important issues concerning optimization, e.g., definition, classification, and conventional optimization strategies. Altogether seven NIC optimization algorithms, including the CSA, PSO, ACO, HS, SA, DE and MEC, are next discussed in detail. This work has also reviewed some interesting hybrid NIC methods along with their applications. In the publications [P1] to [P8], we propose and study several new hybrid NIC optimization schemes and employ them in nonlinear, multi-modal, and dynamical optimization problems, such as neural networks training and optimization of fuzzy classification systems. As an example, the CSA inspired by the clonal selection principle of immunology is a principal NIC method discussed in our publications, which is an adequate candidate for the multi-modal optimization [P1]. Nevertheless, its convergence speed is relatively slower than that of the GA. In [P6], we study a hybrid CSA-DE, in which the convergence of the original CSA is significantly improved by combining the DE with the CSA-based global search. Alternatively, the CSA can be hybridized with other NIC algorithms as a general parameter optimizer. In [P5], the CSA is employed in the proposed CSA-HS to optimize the harmony memory of the HS method.

The scientific importance of the work in this dissertation is explained as follows. Firstly, we develop several novel solution candidate evaluation strategies for elite maintenance. For example, in the proposed CSA, the fitness of an Ab includes both its affinity to the

Ag and affinity to other Abs, and the self-suppression is used to eliminate those relatively weak candidates with high similarities to the outperformed ones. In the modified ACO, a pheromone-based instead of a fitness-dependent candidate selection strategy is applied to maintain the potential solutions under the changing environments in order to cope with the dynamical optimization problems. Secondly, the hybrid NIC methods can properly balance the exploitation and exploration in the solution space so as to cope with the common premature drawback and thus achieve improved optimization performances, including reliability, robustness, and acceleration of convergence. It has been demonstrated that our hybrid techniques are well capable of solving a wide variety of optimization problems even with a small number of initial candidates. To summarize, studying the hybridization of these NIC methods has allowed us to overcome the shortcomings of individual algorithms while at the same time retaining their individual advantages. Hybridization has led to the following benefits:

- Improvement of convergence performance,
- Enhanced solution quantity and quality,
- Creation of compact, reconfigurable systems.

Although the hybridization of the NIC optimization methods has been proven to be superior to the stand-alone techniques in dealing with certain practical problems, we should emphasize that the 'No Free Lunch' theorem is a fundamental barrier to the exaggerated claims of the power and efficiency of any specific optimization algorithm [Wol97]. This theorem predicts that if an algorithm is especially efficient in one type of problem, it is guaranteed to be inefficient in another type. That is to say, there is no one 'best' optimization algorithm, because whatever an algorithm gains in performance in one class of problems is necessarily offset by poor performance in the remaining ones. Therefore, formulating a universal solution that can effectively solve all types of optimization problems is difficult in practice. The way to handle the negative implication of the 'No Free Lunch' theorem is to restrict the applications of a given optimization algorithm to only a particular type of task.

6.2 Topics for future work

It is well known that a satisfactory optimization algorithm should possess the properties of robustness, efficiency, and accuracy. Unfortunately, these objectives usually conflict with one another. For instance, a robust optimization method may require a long time to converge. Hence, a comprehensive performance evaluation and comparison among the current hybrid NIC optimization methods with regard to their search effectiveness, convergence characteristics, and computational complexity need to be made. In addition, study of more and better hybridization of the NIC optimization approaches as well as fusion with the non-nature-inspired strategies is definitely a promising research topic for our future work.

NIC optimization methods have different origins, and they mimic a variety of the phenomena observed in biological systems, nature, and human society. However, they also share some similarities. For example, regardless of the fact that it is motivated by natural immune systems, the CSA can be simply considered as a special type of GA. Thus, a general framework that can cover all the NIC optimization methods should be developed and investigated. In other words, the individual NIC optimization techniques are, in fact, alternative types of our collective framework, which unifies their common conceptual bases and characteristics. This feasible framework can provide us with deep insight into the principles, architectures, and algorithms of existing NIC methodologies as well as a useful guideline for choosing the best ones in order to satisfy specific requirements. Such a framework may also lay a solid theoretical basis for the aforementioned future exploratory research on new hybrid NIC optimization schemes.

References

[Abr06] A. Abraham, C. Grosan, and V. Ramos (Eds.), *Swarm Intelligence and Data Mining*. Berlin, Germany: Springer-Verlag, 2006.

[Abr08] A. Abraham and C. Grosan (Eds.), *Engineering Evolutionary Intelligent Systems*. Berlin, Germany: Springer-Verlag, 2008.

[Ang97] P. Angeline, "Tracking extrema in dynamic environments," in *Proceedings of the 6th International Conference on Evolutionary Programming*, Indianapolis, IN, April 1997, pp 335-346.

[Ben89] G. Beni and U. Wang, "Swarm intelligence in cellular robotic systems," in *Proceedings of the NATO Advanced Workshop on Robots and Biological Systems*, Tuscany, Italy, June 1989, pp. 26-30.

[Ber08] H. S. Bernardino, H. Barbosa, A. Lemonge, and L. G. Fonseca, "A new hybrid AIS-GA for constrained optimization problems in mechanical engineering," in *Proceedings of the IEEE World Congress on Computational Intelligence*, Hong Kong, P. R. China, June 2008, pp. 1455-1462.

[Bil95] G. Bilchev and I. Parmee, "The ant colony metaphor for searching continuous design spaces," in *Proceedings of the AISB Workshop on Evolutionary Computing*, Sheffield, UK, April 1995, pp. 25-39.

[Bil96] G. Bilchev, I. Parmee, and A. Darlow, "The inductive genetic algorithm with applications to the fault coverage test code generation problem," in *Proceedings of the 4th European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, September 1996, pp. 452-457.

[Bla02] T. M. Blackwell and P. J. Bentley, "Dynamic search with charged swarms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, New York, NJ, July 2002, pp. 19-26.

[Bla04] T. M. Blackwell and J. Branke, "Multi-swarm optimization in dynamic environments," in *Proceedings of Applications of Evolutionary Computing:*

EvoWorkshops 2004: EvoBIO, EvoCOMNET, EvoHOT, EvoISAP, EvoMUSART, and EvoSTOC, Coimbra, Portugal, April 2004, pp. 489-500.

[Bla06] T. Blackwell and J. Branke, "Multiswarms, exclusion, and anti-convergence in dynamic environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 459-472, August 2006.

[Bra00] J. Branke, T. Kaubler, C. Schmidt, and H. Schmeck, "A multi-population approach to dynamic optimization problems," *Adaptive Computing in Design and Manufacturin*g, pp. 299-308, 2000.

[Cas99] L. N. de Castro and F. J. von Zuben, "Artificial immune systems: Part I-Basic theory and applications," Technical Report RT-DCA 01/99, FEEC/UNICAMP, Brazil, 1999.

[Cas00] L. N. de Castro and F. J. von Zuben, "Artificial immune systems: Part II-A survey of applications," Technical Report RT-DCA 02/00, FEEC/UNICAMP, Brazil, 2000.

[Cas02] L. N. de Castro and F. J. von Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 3, pp. 239-251, 2002.

[Cas07] L. N. de Castro, "Fundamentals of natural computing: An overview," *Physics of Life Reviews*, vol. 4, no. 1, pp. 1-36, 2007.

[Che03] Y.-M. Chen, "Passive filter design using genetic algorithms," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 1, pp. 202-207, February 2003.

[Ciu02] G. Ciuprina, D. Ioan, and I. Munteanu, "Use of intelligent-particle swarm optimization in electromagetics," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 1037-1040, March 2002.

[Coe04] C. Coello and N. Cortes, "Hybridizing a genetic algorithm with an artificial immune system for global optimization," *Engineering Optimization*, vol. 36, no. 5, pp. 607-634, October 2004.

[Cor99] D. W. Corne, M. Dorigo, and F. Glover (Eds.), *New Ideas in Optimization*. Berkshire, UK: McGraw-Hill, 1999.

[Cru08] F. Cruz, A. Coelho, and L. P. Reis, "Automatic parameterization for expeditious modeling of virtual urban environments − A new hybrid metaheuristic" in *Proceedings of the 4th International Workshop on Artificial Neural Networks and Intelligent Information*, Funchal, Portugal, May 2008, pp. 334-337.

[Dor06] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28-39, November 2006.

[Emb97] M. J. Embrechts and S. Benedek, "Identification of nuclear power plant transients with neural networks," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Orlando, FL, October 1997, pp. 912-917.

[Eng05] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*. West Sussex, England: John Wiley & Sons Ltd, 2005.

[Fra98] O. Francois, "An evolutionary strategy for global minimization and its Markov chain analysis," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 3, pp. 77-90, September 1998.

[Fuk99] T. Fukuda, K. Mori, and M. Tsukiyama, "Parallel search for multi-modal function optimization with diversity and learning of immune algorithm," *Artificial Immune Systems and Their Application*. D. Dasgupta (Eds.), Berlin, Germany: Spring-Verlag, 1999.

[Fuk08] Y. Fukuyama, "Fundamentals of particle swarm optimization techniques," *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*. K. Y. Lee and M. A. El-Sharkawi (Eds.), Hoboken, NJ: John Wiley & Sons Ltd, 2008.

[Gam97] L. M. Gambardella, E. D. Taillard, and M. Dorigo, "Ant colonies for the QAP," Technical Report IDSIA 4-97, Lugano, Switzerland, 1997.

[Gee01] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60-68, February 2001.

[Gee02] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "Harmony search optimization: application to pipe network design," *International Journal of Modeling and Simulation*, vol. 22, no. 2, pp. 125-133, 2002.

[Gee08] Z. W. Geem, "Music-inspired optimization algorithm: harmony search," full text is available at http://www.hydroteq.com/.

[Glo97] F. Glover and M. Laguna, *Tabu Search*. New York, NJ: Kluwer Academic Publishers, 1997.

[Gro05] C. Grosan, A. Abraham, and M. Nicoara, "Search optimization using hybrid particle sub-swarms and evolutionary algorithms," *International Journal of Simulation Systems, Science and Technology*, vol. 6, no. 10, pp. 60-79, 2005.

[Guo04] P. Guo, P. Han, and D. Wang, "Modifications of mind evolutionary computation and its application in 2-DOF PID controller," in *Proceedings of the 5th World Congress on Intelligent Control and Automation*, Hangzhou, P. R. China, June 2004, pp 2268-2270.

[Haj99] P. Hajela and J. S. Yoo, "Immune network modeling in design optimization," *New Ideas in Optimization*. D. Corne, M. Dorigo, and F. Glover (Eds.), London, UK: McGraw Hill, 1999.

[Han02] K. Han and J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 6, pp. 580-593, December 2002.

[Hau98] R. L. Haupt and S. E. Haupt, *Practical Genetic Algorithms*. New York, NJ: John Wiley & Sons Ltd, 1998.

[Her95] F. Herrera, M. Lozano, J. L. Verdegay, "Tackling fuzzy genetic algorithms," *Genetic Algorithms in Engineering and Computer Science*. G. Winter, J. Periaux, M. Galan, and P. Cuesta (Eds.), New York, NJ: John Wiley & Sons Ltd, 1995.

[Ho05] S. L. Ho, S. Yang, H. C. Wong, K. W. E. Cheng, and G. Ni, "An improved ant colony optimization algorithm and its application to electromagnetic devices designs," *IEEE Transactions on Magnetics*, vol. 41, no. 5, pp. 1764-1767, May 2005.

[Hoo07] A. Hoorfar, "Evolutionary programming in electromagnetic optimization: a review," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 523-537, March 2007.

[Hou08] T. Hou, C. Su, and H. Chang, "Using neural networks and immune algorithms to find the optimal parameters for an IC wire bonding process," *International Journal Expert Systems with Applications*, vol. 34, no. 1, pp. 427-436, January 2008.

[Ish95] H. Ishibuchi, K. Nozaki, N. Yamamoto, and H. Tanaka, "Selecting fuzzy if-then rules for classification problems using genetic algorithms," *IEEE Transactions on Fuzzy Systems*, vol. 3 no. 3, pp. 260-270, August 1995.

[Jan97] J. R. Jang, S. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, NJ: Prentice-Hall, 1997.

[Jie06] J. Jie, J. Zeng, and C. Han, "An extended mind evolutionary computation model for optimizations," *Applied Mathematics and Computation*, vol. 185, no. 2, pp. 1038-1049, February 2007.

[Kan04] S. L. Kang and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Computers and Structures*, vol. 82, no. 9-10, pp. 781-798, 2004.

[Ken95] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Perth, Australia, December 1995, pp. 1942-1945.

[Kim02] J. Kim and P. J. Bentley, "Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection," in *Proceedings of the IEEE World Congress on Computational Intelligence*, Honolulu, HI, May 2002, pp. 1015-1020.

[Kir83] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671-680, May 1983.

[Kuo08] C. Kuo, “A novel coding scheme for practical economic dispatch by modified particle swarm approach power systems,” *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1825-1835, November 2008.

[Li07] H. Li and L. Li, “A novel hybrid real-valued genetic algorithm for optimization problems,” in *Proceedings of the International Conference on Computational Intelligence and Security*, Harbin, P. R. China, December 2007, pp. 91-95.

[Liu07] H. Liu, A. Abraham, and W. Zhang, “A fuzzy adaptive turbulent particle swarm optimisation,” *International Journal of Innovative Computing and Applications*, vol. 1, no. 1, pp. 39-47, 2007.

[Mah07] M. Mahdavi, M. Fesangharyb, and E. Damangirb, “An improved harmony search algorithm for solving optimization problems,” *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567-1579, May 2007.

[Mal08] M. Malaki, A. Pourbaghery, and H. Abolhassani, “A combinatory approach to fuzzy clustering with harmony search and its applications to space shuttle data,” in *Proceedings of the Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on Advanced Intelligent Systems*, Nagoya, Japan, September, 2008, pp. 1154-1159.

[Mat00] M. Mathur, S. B. Karale, S. Priye, V. K. Jayaraman, and B. D. Kulkarni, “Ant colony approach to continuous function optimization,” *Industrial and Engineering Chemistry Research*, vol. 39, no. 10, pp. 3814-3822, October 2000.

[Moh93] O. A. Mohammed, R. S. Merchant, and F. G. Uler, “Utilizing Hopfield neural networks and an improved simulated annealing procedure for design optimization of electromagnetic devices,” *IEEE Transactions on Magnetics*, vol. 29, no. 6, pp. 2404-2406, November 1993.

[Mon08] A. J. Monticelli, R. Romero, and E. N. Asada, “Fundamentals of simulated annealing,” *Modern Heuristic Optimization Techniques: Theory and Applications to Power Systems*. K. Y. Lee and M. A. El-Sharkawi (Eds.), Hoboken, NJ: John Wiley & Sons Ltd, 2008.

[Omr08] M. G. H. Omran, and M. Mahdavi, "Global-best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643-656, May 2008.

[Ova06] S. J. Ovaska, A. Kamiya, and Y. Q. Chen, "Fusion of soft computing and hard computing: Computational structures and characteristic features," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 36, no. 3, pp. 439-448, May 2006.

[Pan07] M. Pant, R. Thangaraj, and A. Abraham, "A new PSO algorithm with crossover operator for global optimization problems," in *Proceedings of the 2nd International Symposium on Hybrid Artificial Intelligent Systems*, Salamanca, Spain, November 2007, pp. 215-222.

[Par02] K. Parsopoulos and M. Vrahatis, "Recent approaches to global optimization problems through particle swarm optimization," *Natural Computing*, vol. 1, no. 2-3, pp. 235-306, June 2002.

[Par06] D. Parrott and X. Li, "Locating and tracking multiple dynamic optima by a particle swarm model using speciation," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 440-458, August 2006.

[Pol02] R. Poli and W. B. Langdon, *Foundations of Genetic Programming*. Berlin, Germany: Springer-Verlag, 2002.

[Pri99] K. Price, "An Introduction to differential evolution," *New Ideas in Optimization*. D. Corne, M. Dorigo, and F. Glover (Eds.), London, UK: McGraw Hill, 1999.

[Pul06] P. Pulkkinen, T. Hakala, and R. Ritala, "Hybrid optimization algorithm for scheduling decision support computational intelligence for modelling," in *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation Jointly with International Conference on Intelligent Agents Web Technologies and International Commerce*, Sydney, Australia, November 2006, pp. 253-253.

[Ram05] V. Ramos, C. Fernandes, and A. C. Rosa, "Social cognitive maps, swarm collective perception and distributed search on dynamic landscapes," full text is available at http://arxiv.org/abs/nlin.AO/0502057.

[Ram06] V. Ramos, C. Fernandes, and A. C. Rosa, "Societal implicit memory and his speed on tracking extrema in dynamic environments using self-regulatory swarms," full text is available at http://arxiv.org/abs/cs.MA/0512003, 2006.

[Sch89] J. Schaffer, R. Caruana, L. Eshelman, and R. Das, "A study of control parameters affecting online performance of genetic algorithms for function optimization," in *Proceedings of the 3rd International Conference on Genetic Algorithms*, San Mateo, CA, June 1989, pp. 51-60.

[Shi03] X. Shi, Y. Lu, C. Zhou, H. Lee, W. Lin, and Y. Liang, "Hybrid evolutionary algorithms based on PSO and GA," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Canberra, Australia, December 2003, pp. 2393-2399.

[Sin03] A. Sinha and D. E. Goldberg, "A survey of hybrid genetic and evolutionary algorithms," Technical Report ILLIGAL 2003004, University of Illinois at Urbana-Champaign, Urbana, IL, 2003.

[Sim06] D. Simopoulos, S. Kavatza, and C. Vournas, "Unit commitment by un enhanced simulated annealing algorithm," *IEEE Transactions on Power Systems*, vol. 21, no. 1, pp. 68-76, February 2006.

[Sol02] C. Solnon, "Boosting ACO with a preprocessing step," *Applications of Evolutionary Computing Applications of Evolutionary Computing: EvoWorkshops*, E. J. W. Boers *et al.* (Eds.), Berlin, Germany: Spring-Verlag, 2002.

[Spa03] J. C. Spall, *Introduction to Stochastic Search and Optimization*. Hoboken, NJ: John Wiley & Sons Ltd, 2003.

[Sto97] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, December 1997.

[Sun03a] C. Sun, Y. Sun, and W. Wang, "A survey of MEC: 1998-2001," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Hammamet, Tunisia, October 2003, pp. 648-656.

[Sun03b] C. Sun, X. Qi, and O. Li, "Pareto-MEC for multi-objective optimization," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Hammamet, Tunisia, October 2003, pp. 5045-5049.

[Tim08] J. Timmis, P. Andrews, N. Owens, and E. Clark, "An interdisciplinary perspective on artificial immune systems," *Evolutionary Intelligence*, vol. 1, no. 1 pp. 2-26, 2008.

[Urs00] R. Ursem, "Multinational gas: multimodal optimization techniques in dynamic environments," in *Proceedings of A Recombination of the 5th Annual Genetic Programming Conference and the International Conference on Genetic Algorithms*, Las Vegas, NV, July 2000, pp. 19-26.

[Vas96] J. A. Vasconcelos, L. Krahenbuhl, and L. Nicolas, "Simulated annealing coupled with the tabu search method for continuum optimization in electromagnetics," *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 1206-1209, May 1996.

[Wan04] X. Wang, X. Z. Gao, and S. J. Ovaska, "Artificial immune systems in optimization − A survey," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, The Hague, The Netherlands, October 2004, pp. 3415-3420.

[Wri91] A. H. Wright, "Genetic algorithms for real parameter optimization," *Foundations of Genetic Algorithms*. G. Rawlins (Eds.), San Mateo, CA: Morgan Kaufmann Publishers, 1991.

[Wri06] A. H. Wright and J. N. Richter, "Strong recombination, weak selection, and mutation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, Seattle, WA, July 2006, pp. 1369-1376.

[Wod97] M. Wodrich and G. Bilchev, "Cooperative distributed search: the ants' way," *Control Cybernetics*, vol. 26, no. 3, pp. 413-445, 1997.

[Wol97] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67-82, April 1997.

[Yen95] J. Yen, J. Liao, D. Randolph, and B. Lee, “A hybrid approach to modeling metabolic systems using genetic algorithms and simplex method,” in *Proceedings of the 11th IEEE Conference on Artificial Intelligence for Applications*, Los Angeles, CA, February 1995, pp. 277-283.

[Zad96] L. A. Zadeh, “The roles of soft computing and fuzzy logic in the conception, design and deployment of intelligent system,” in *Proceedings of the IEEE Asia Pacific Conference on Circuits and Systems*, Seoul, Korea, November 1996, pp. 3-4.

[Zha07] Z. Zhan, J. Xiao, and J. Zhang, “Particle swarm optimization with a crossover operator,” in *Proceedings of the 8th International Conference on Artificial Evolution*, Paris, France, October 2007, pp. 1036-1040.

ISBN 978-951-22-9858-7
ISBN 978-951-22-9859-4 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)