PREMIER REFERENCE SOURCE

Innovations and Developments of Swarm Intelligence Applications

Yuhui Shi

Innovations and Developments of Swarm Intelligence Applications

Yuhui Shi
Xi'an Jiaotong-Liverpool University, China

Information Science
REFERENCE

Managing Director:	Lindsay Johnston
Senior Editorial Director:	Heather A. Probst
Book Production Manager:	Sean Woznicki
Development Manager:	Joel Gamon
Acquisitions Editor:	Erika Gallagher
Typesetter:	Nicole Sparano
Cover Design:	Nick Newcomer, Lisandro Gonzalez

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Innovations and developments of swarm intelligence applications / Yuhui Shi, editor.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-4666-1592-2 (hardcover) -- ISBN 978-1-4666-1593-9 (ebook) -- ISBN 978-1-4666-1594-6 (print & perpetual access) 1. Swarm intelligence. I. Shi, Yuhui.
Q337.3.I56 2012
006.3--dc23
2012002108

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Editorial Advisory Board

Preface

Swarm intelligence is a collection of nature-inspired optimization algorithms. Each swarm intelligence algorithm is a population-based stochastic optimization algorithm even though each has a different inspiration and/or motivation. Usually each individual in a population represents a potential solution, which may be a good or bad solution, to the problem to be solved. The purpose of a swarm intelligence algorithm is to iteratively update the population of individuals toward better and better solution areas iteration over iteration with high probability. Each different swarm intelligence algorithm has a different updating mechanism. For example, for the Particle Swarm Optimization (PSO) algorithm, the updating mechanism is to "fly" the individuals (called particles in PSO) toward better and better solution areas. Therefore its updating mechanism is to update particles' velocities dynamically according to each particle's historical flying experience and its companion's flying experience. There are many other swarm intelligence algorithms that have been reported in the literature, which include the ant colony optimization algorithm, artificial immune system, bacterial forging optimization algorithm, bee colony optimization algorithm, brain storm optimization algorithm, firefly optimization algorithm, firework optimization algorithm, fish school search optimization algorithm, intelligent water drops algorithm, and the shuffled frog-leaping algorithm, to name just a few. Compared with traditional search algorithms, swarm intelligence algorithms are less sensitive to the initial starting search points, have more capability to jump out of local minima, are suitable for wider range of optimization problems, and only require that any potential solution can be evaluated.

Even though most swarm intelligence algorithms are designed to solve unconstrained single objective optimization problems, swarm intelligence algorithms have been successfully modified and extended to solve all kind of optimization problems which include constrained single objective optimization problems, multi-objective optimization problems, constrained multi-objective optimization problems, combinatorial optimization problems, scheduling problems, etc. For the constrained optimization problems, there are several commonly used approaches to solve them using swarm intelligence algorithms such as penalty function approach which adds constraint violations into the objective function as a penalty, special encoding approach which generates only feasible solutions by designing special encoding method and/or special operators, repair approach which repairs each generated infeasible solution to be a feasible, separation of constraints and objective approach which evaluates objective function by taking consideration of constraints simultaneously, and multi-objective optimization approaches which convert a constrained optimization problem into a bi-objective or multi-objective optimization problem. For the multi-objective optimization problems, there are several commonly used approaches to solve them using swarm intelligence algorithms such as aggregating approaches which combine all objectives into a single objective, Lexicographic ordering approaches which rank the objectives in order of importance and optimize each objective independently, sub-population approaches which use a different sub-population to optimize a different objective, and Pareto-based approaches which use the concept of non-dominated solutions to find optimal solutions along the Pareto front. Among them, the Pareto-based approaches are the most commonly used approaches to solve multi-objective optimization problems using swarm intelligence algorithms.

For complicated nonlinear optimization problems to effectively and efficiently solve optimization problems, an optimization algorithm needs to possess the capability to either converge or diverge and whether to converge or diverge depends on the search state the search process is currently in. One way to solve optimization problems efficiently and effectively is to use a right algorithm with a right parameter set at the right search state. Each algorithm with a different set of parameters will perform differently and therefore will be better suitable for one kind of problem or for one search state among all different search states during the whole search process. For example, under some conditions, an optimization algorithm is preferred to have the global search capability or exploration capability, and under some other conditions, it is preferred to have the local search capability or exploitation capability. For the above purposes, studies on adaptation of swarm intelligence algorithms have been reported in the literature. For example, the adaptation of neighborhood structures or the adaptation of parameters of particle swarm optimization algorithms has kept being an active and hot research topic. Another research trend with this regard is to employ different optimization algorithms with different scales of learning capability. For example, one optimization algorithm is utilized for long-term learning, that is to learn the learning capability; another optimization algorithm is utilized for short-term learning, that is to learn the content that is required to be learnt in order to solve the optimization problem efficiently and effectively. One good example of researches on this kind of different scales of learning capability is the current research interests on memetic algorithms.

One of the challenging tasks for swarm intelligence algorithms is to solve large scale optimization problems, that is, the dimension of the problem is large, say larger than 1000. There are two kind of large scale problems. One is the kind of separable problems and the other is the kind of un-separable problems. For a separable large scale problem, its objective function value has contribution from each variable independently. Therefore, the large scale problem can be divided into several small or medium scale problems each of which will then be solved by a swarm intelligence algorithm independently. For an un-separable problem, each variable does not contribute to the problem's objective function independently, but as a whole, therefore, an un-separable problem in principle cannot be directly divided into several smaller problems as that do for separable problems. A simple and straightforward method to solve a large scale un-separable optimization problem using swarm intelligence algorithm is to transform the original un-separable optimization problem to be a new separable problem which then can be divided into several smaller problems for a swarm intelligence algorithm to solve. Certainly, such transformation may not always be possible, or even possible, it may not be easy to find the right transformation. These problems are considered as true large scale problems. For a true large scale problem, one way to solve it is to divide it into several smaller problems randomly or predeterminatively. For each smaller problem, a small number of variables will be optimized by a swarm intelligence algorithm while other remaining variables are kept to be constants which are previously determined by solving other smaller problems using swarm intelligence algorithms. All the smaller problems will be solved iteratively in the hope that the process will converge to one single solution that is good enough to the original large scale problem. There are usually overlaps of variables among all smaller problems. Another way to solve an un-separable problem is to treat the problem as a whole. In this way, the priority of the problem solving is to reduce computation cost, especially the number of function evaluations because it is the function evaluation that contributes the most to the computation cost and to the difficulty of solving a large scale problem. Therefore, it is critical to take advantage of existing knowledge such as the domain knowledge and knowledge that can be revealed by the search process itself at the risk of premature convergence. One kind of knowledge that can be revealed by the search process is the population diversity. For example, for particle swarm optimization algorithms, there are several definitions of population diversities which include position diversity, velocity diversity, and cognitive (*pbest*) diversity.

Swarm intelligence algorithms are good at solving complicated problems which can be represented as nonlinear, non-differentiable, and un-continuous functions. They have been successfully applied to solve a lot of real-world applications which cover almost all areas where there are things to be optimized. Actually, it is the successful real-world applications that are the sources of vitality for the swarm intelligence algorithms. Without good applications, the research on swarm intelligence algorithms will eventually lose its vitality. Therefore, one important and critical research direction on swarm intelligence is to find more and successful real-world applications.

WHAT IS THE BOOK ABOUT?

This book volume is about current researches on swarm intelligence algorithms and their applications. It does not intend to cover all aspects of researches on swarm intelligence algorithms. It actually is a collection of papers which were published in the 2010 issues of the *International Journal of Swarm Intelligence Research*. It can be looked as a snapshot of current research trend on swarm intelligence. This book is intended for researchers who have been working on or are interested in the research areas of swarm intelligence. It can also be used as a reference book for graduate students and senior undergraduate students who are interested in conducting their studies on swarm intelligence algorithms and/or their applications.

ORGANIZATION OF THE BOOK

This book volume consists of 17 chapters, which are organized into two sections for the convenience of reference. Section 1 includes 11 chapters, which are about current research works on particle swarm optimization algorithms and their application.

In the chapter, "*Beyond Standard Particle Swarm Optimization*," Clerc discussed Standard Particle Swarm Optimization (SPSO) algorithms. There are two versions of so-called standard particle swarm optimization algorithms, which are SPSO-2007 and SPSO-BK. The two versions are similar to each other. The similarities and differences of the two versions are discussed with regards to the following aspects: parameters settings, swarm size, initialization of the positions and velocities, confinement methods which are used to confine positions which are either too large or too small, and neighborhood topologies. Clerc further presented a formula to calculate the probabilities for a particle to be informed about the best particle's position for both SPSO-2007 and SPSO-BK, respectively. If the probability is low, the particle has more freedom to explore before it is impacted by the best particle's position. The probability may help to determine that a stagnation has occurred if the PSO has not improved its performance after a certain number of generations that any particle has been informed with probability one.

Furthermore, there are several common criticisms about SPSOs which are a PSO can easily get trapped into a local minimum, premature convergence may happen even with better topologies, the swarm size is constant, and it is sensitive to rotations of the landscape. The two versions of standard PSOs can be easily merged, but obviously with the same common drawbacks as those in the two SPSOs. Therefore, Clerc goes beyond a simple merging but suggests simple but robust changes to help PSO to escape from local minima, to have a global convergence, to have a variable swam size, and to rotate insensitively, which may be proposed to be a new flexible standard for particle swarm optimization algorithms.

In the chapter, "*Biases in Particle Swarm Optimization*," Spears *et al.* discussed biases embedded in particle swarm optimization algorithms. According to the No Free Lunch Theorem, an algorithm cannot be the best for all optimization problems, but can be the best for the optimization algorithms which are aligned with the algorithm. By considering the problem and algorithm as two vectors, the problem is aligned with the algorithm when their dot product is zero, i.e. the algorithm is well matched to the problem. In the chapter, Spears *et al.* showed that PSOs with commonly used particles updating equations are rotationally variant and can concentrate particles along paths parallel to the coordinate axes. The rotational variance is closely related to the coordinate axes bias. Spears *et al.* showed that the connection between rotational variance and coordinate axes bias is not the effect of population size, problem dimension, and PSO parameters settings. Based on this explicit connection, Spears *et al.* further created fitness function landscapes that are easy or hard for PSO to solve, depending on the rotational angle of the function landscapes. The intention of the authors is to ask users of PSOs to be aware of the bias including its cause and its effect, but not to discourage users from using PSOs.

In the chapter, "*Taguchi-Particle Swarm Optimization for Numerical Optimization*," Ting *et al.* applied Taguchi method to particle swarm optimization algorithms. The Taguchi method is an important tool for robust design. In the Chapter, Ting *et al.* proposed a hybrid Taguchi-Particle Swarm Optimization (TPSO) which combines the particle swarm optimization algorithm with the Taguchi method. The common drawback of hybrid algorithm is the extra algorithm complexity and therefore extra computational cost. To overcome this, in each generation of the proposed TPSO, the Taguchi method is run only once after conventional PSO particles' updates, therefore it does not add in too much algorithm complexity and computational cost. The Taguchi method utilizes the two-level orthogonal array and the Signal-to-Noise Ratio (SNR). With the Taguchi method and according to the two-level orthogonal array, a new particle is generated by selecting better dimensional values from two *bpest* particles which are randomly chosen. The effect of applying the Taguchi method is to create more diverse population to avoid premature convergence and to help particles to jump out of local minima, which are not good enough solutions.

In the chapter, "*Constraint Handling in Particle Swarm Optimization*," Leong and Yen proposed to apply the multi-objective optimization method to handle constraints in optimization problems with constraints. Leong and Yen first transform constrained optimization problems into unconstrained bi-objective optimization problems, in which one objective is the original objective function and the other objective is the sum of all the constraint violations. The advantage of utilizing this strategy to solve constrained optimization problems is that it needs neither penalty function nor the selection proportion balance between feasible solutions and infeasible solutions. The transformed bi-objective optimization problems are then solved by applying Pareto-based Multi-Objective Particle Swarm Optimization algorithm (MOPSO). This differs from other Pareto-based optimization algorithms where a group of solutions (Pareto-front) are required to be obtained; here only the global optimum, which is feasible, is required to be obtained by the Pareto-based MOPSO. In the implementation of MOPSO, personal best particles are updated by giving preference to feasible solutions over infeasible solutions, to non-dominated solutions over dominated solutions, so that the MOPSO can have higher possibility to find feasible solutions towards true Pareto front fast; the global best archive is updated to store only the best feasible solutions found so far and the infeasible non-dominated solutions with less constraint violation hidden information which can be exploited to guide search towards feasible solutions. Furthermore, mutation with nonlinear dynamic ranges is applied to personal and global best to facilitate the global search in early stage and local search in late stage.

In the chapter, "*Adaptive Neuro-Fuzzy Control Approach Based on Particle Swarm Optimization*," Ei-Far used a minimum velocity checking in addition to the maximum velocity checking in the velocity update in the proposed particle swarm optimization algorithm. The velocity is updated with specifically designed formula when it is below the predefined minimum velocity threshold. The purpose of the minimum velocity threshold is to keep particles to continue flying until the algorithm is terminated. Generally speaking, a large minimum velocity threshold facilitates a global search while a small minimum velocity threshold facilitates a local search. The balance between exploration and exploitation could be achieved if the minimum velocity threshold could be dynamically adjusted based on the search information revealed by the search process. When applying the proposed particle swarm optimization algorithm to solve constrained optimization problems, if a particle violates constraints, its velocity update is further modified by removing the contribution from the current velocity, that is, the velocity completely depends on only self-cognition part and social-cognition part. The proposed particle swarm optimization is applied to tune and optimize a neuro-fuzzy controller's parameters such as scaling factors, membership functions, and rule base. The designed neuro-fuzzy controller is applied to control a nonlinear single machine power system and a nonlinear inverted pendulum system, respectively.

In the chapter, "*Design of Multi-Criteria PI Controller Using Particle Swarm Optimization for Multiple UAVs Close Formation*," Zhang *et al.* discussed the issue of applying a particle swarm optimization algorithm to design a PI controller to control the close formation of multiple Uninhabited Aerial Vehicles (UAVs). The automatic cooperative control of a group of UAVs flying in close formation is a very active and hot research topic. Many researches have been conducted by utilizing many classic and modern control approaches such as Proportional plus Integral (PI) control, nonlinear adaptive control, robust control, *etc.* These approaches usually only consider one single performance index and can achieve good characteristic in frequency-domain, but not in time-domain. In this chapter, Zhang *et al.* employed a particle swarm optimization algorithm to design PI controller online. By taking consideration of overshoot, rise time, and system accumulated absolute error instead of a single performance index, the online designed PI controller can control the close formation flight of multiple UAVs with satisfaction.

In the chapter, "*Oscillation Damping Enhancement via Coordinated Design of PSS and FACTS-Based Stabilizers in a Multi-Machine Power System Using PSO*," Abido and Bamasak presented a method of applying the Particle Swarm Optimization (PSO) algorithm to design stabilizers in a multi-machine power system. Because disturbances may cause power systems to experience low frequency oscillations which may sustain and grow to cause system separation if no adequate damping is available, in this chapter Abido and Bamasak utilized the PSO algorithm to design a stabilizer to damp power system oscillation and increase system oscillation stability in a multi-machine power system. The stabilizer is designed by considering together the Power System Stabilizers (PSSs), Thyristor Controlled Series Capacitor (TCSC)-based stabilizer, and Static Var Compensator (SVC)-based stabilizer in the multi-machine power system. The design of the stabilizer is formulated as an optimization problem so that the PSO algorithm can be utilized to optimize the stabilizer parameter settings to maximize the minimum damping ratio under all system oscillating modes. The designed stabilizer is further tested on a two-area weakly-connected multi-machine power system with unstable inter-area oscillation mode to illustrate the effectiveness of the designed stabilizer.

In the chapter, "*Compensation of Voltage Sags with Phase-Jumps through DVR with Minimum VA rating Using PSO Based ANFIS Controller*," Ramakuru *et al.* used Particle Swarm Optimization (PSO) algorithm based Adaptive Neuro–Fuzzy Inference System (ANFIS) to implement a Dynamic Voltage Restorer (DVR) with minimum Volt-Amperes (VA) rating. A DVR is a power electronic device, which protects loads from typical voltage problems such as phase-angle jump and load switching on and off in a distribution system. It is a series connected custom power device to mitigate the voltage sags with phase jumps. A DVR injects required amount of VA into the distribution system to compensate the voltage sag/swell with phase jumps. To be cost effective, a DVR requires to have a minimum VA rating for a given distribution system without compromising compensation capability. In this chapter, Ramakuru *et al.* utilized a PSO algorithm to obtain an optimal angle to inject DVR voltage in series to the line impedance to have minimum VA loading on DVR and to remove phase jumps in the three-phases. Furthermore, the designed DVR is implemented with an ANFIS to work online with minimum VA loading.

In the chapter, "*Optimal Power Flow with TCSC and TCPS Modeling using Craziness and Turbulent Crazy Particle Swarm Optimization*," Roy *et al.* proposed two new versions of particle swarm optimization algorithms, i.e., Craziness Based Particle Swarm Optimization (CRPSO) and Turbulent Crazy Particle Swarm Optimization (TRPSO). In CRPSO, a predefined craziness probability is introduced to maintain the diversity of the directions of search to prevent premature convergence. In TRPSO, a minimum velocity threshold is utilized to control the velocity of particles to prevent premature convergence. The proposed two PSO algorithms are applied to solve Optimal Power Flow (OPF) in power system incorporating Flexible AC Transmission Systems (FACTS), which include Thyristor-Controlled Series Capacitor (TCSC) and Thyristor Controlled Phase Shifting (TCPS). The CRPSO and TRPSO are further compared with the Particle Swarm Optimization with Inertia Weight Approach (PSOIWA) and the Real Coded Genetic Algorithm (RGA) to illustrate their better performance.

In the chapter, "*Congestion Management Using Hybrid Particle Swarm Optimization Technique*," Balaraman and Kamaraj proposed a Hybrid Particle Swarm Optimization (HPSO) algorithm. The HPSO combines an Evolutionary Programming (EP) algorithm and a Particle Swarm Optimization algorithm (PSO). In each generation of the HPSO, first, N individuals (particles) are generated according to the PSO updating equations where N is the population size; second, the EP is utilized to generate another N individuals; the 2N individuals are then sorted in ascending order according to their fitness values and the first half of the 2N individuals will survive and go through the PSO updating equation again to generate N individuals which will be copied into the next generation. The proposed HPSO is applied to solve congestion management problem in a pool based electricity market. Congestion in transmission lines is a challenging technical problem that has to be dealt with. The congestion management problem is modelled as a constrained optimization problem. The constraints are then added into the problem's objective function as a penalty function, which is based on the degrees of constraint violations, to form the fitness function for the HPSO.

In the chapter, "*Particle Swarm Optimization Algorithms Inspired by Immunity-Clonal Mechanism and Their Applications to Spam Detection*," Tan proposed a Clonal Particle Swarm Optimization algorithm (CPSO) which combines the clonal principle in natural immune system and the Particle Swarm Optimization (PSO) algorithms. The cloning operator is employed to clone one particle into N copies in the solution space, then N new particles are generated via clonal mutation and selection processes. In CPSO, the cloning operator is applied to only the best individual in the population. To further improve the CPSO's performance, two strategies are introduced, which are the Advance-and-Retreat (AR) strategy and the Random Black Hole (RBH) strategy. With the AR strategy, the cloning operator is applied to the best individuals of several successive generations instead of only the best individual of the current generation which, as a consequence, can fine exploit the search area around promising candidate solutions, therefore improve the CPSO's search capability and convergence speed to perform better than the conventional PSO. In the RBH strategy, each dimension of a particle is considered independently. For each dimension in every generation, a particle is randomly generated to be close to the current best particle, which is regarded as a black hole for this dimension. Each dimension of each particle will be randomly assigned to be the black hole with small probability. The essence of the RBH strategy is to find other particles, which are close to the current best particle, which may happen to be the true best-fit position which has not been found by the algorithm so far. This mimics a black hole in physics, which has a huge quality (black hole). The proposed algorithm is further applied to spam detection. In this chapter, the Support Vector Machine (SVM) is utilized as a classifier for the purpose of spam detection. Basically, the SVM is to find an optimal hyper plane to have lowest classification errors. The software package LIBSVM is utilized for the implementation of the SVM for spam detection, in which there are four parameters that are required to be determined. The proposed algorithms are employed for the purpose of determining the four parameters.

Section 2 of the book includes 6 chapters, which cover various different swarm intelligence algorithms and their applications.

In the chapter, "*Unit Commitment by Evolving Ant Colony Optimization*," Vaisakh and Srinivas proposed an Evolving Ant Colony Optimization (EACO) algorithm for solving Unit Commitment (UC) problem. The objective of the Unit Commitment (UC) problem is to schedule generation units to minimize the overall cost of the power generation over the scheduled time while satisfying a set of constraints. UC problem is a nonlinear, combinatorial optimization problem which Ant Colony Optimization (ACO) algorithm is suitable for. The EACO combines Genetic Algorithm (GA) and ACO in which the ACO is utilized to solve UC problem and the GA is utilized to find an optimal set of parameters for the ACO algorithm.

In the chapter, "*Bacterial Foraging Optimization*," Passino described a parallel non-gradient optimization algorithm which is inspired by bacteria foraging process over a landscape of nutrients and which was introduced by him in 2002. He gave an insightful overview of biological foundation of bacteria foraging on which the Bacterial Foraging Optimization (BFO) algorithm was developed. In general, it is the flagella that makes bacteria swim. Bacteria swim toward high concentration of nutrients and swim away from high concentration of noxious substances. That is, bacteria will adapt to the environment that is beneficial to them and avoid the environment that is harmful to them. With the help of sensing and decision-making mechanisms, bacteria possess the capability of searching and avoidance in order to survive. With the operation of "conjugation," bacteria undergo mutation, but with very small probability. In a BFO, there are a population of bacteria or individuals x_i, ($i = 1, \ldots n$), where n is the population size, x_i is the position of the ith bacterium (or individual). For a minimization problem f(x), the BFO finding the minimum is to find the position of the bacterium where its f(x) has minimum which corresponds to the place where the nutrient concentration is the highest. There are four operations in the BFO, which are chemotaxis, swarming, reproduction, and elimination/dispersal. The BFO has been further compared with Genetic Algorithm (GA). There are analogies between the BFO and GA. The bacteria reproduction in BFO is similar to the selection operation in GA; the bacteria elimination/dispersal in BFO is similar to the mutation in GA; the nutrient concentration function in BFO is similar to the fitness function in GA; the crossover operation in GA is similar to bacteria splitting which generally is ignored in BFO.

The chapter "*Networks Do Matter*" is an expansion of the authors' work which was submitted to a competitive event held at the 2008 IEEE World Congress. Reynolds and Kinnaird-Heether utilized cultural algorithms to design a controller which is modeled as a state machine. A cultural algorithm consists of a Population Space, a Belief Space, and a communication protocol. Population space is a network of components (population of chromosomes) each of which is evaluated based on the distance it travelled in each generation; Belief space consists of a collection of knowledge sources which are updated according to experience learned by the population of chromosomes; the communication protocol connects the components together with certain topology, e.g., LBest topology, GBest topology, or square topology, which is similar to the neighborhood concept in the particle swarm optimization algorithm. The state-of-the art open source TORCS system is utilized as the racing environment which supports multiple plug and play interfaces to controllers. The implemented controller is then plugged into the TORCS racing environment and is involved in a 3D racing game. During the race, the cultural algorithm is utilized to learn the social context to optimize the controller's state handler routines (or utility functions), which are used to handle the state which the controller is in or transfer the current state to another state when certain conditions are met. Simulation results illustrated that among the three topologies, the square topology worked the best.

In the chapter, "*Honey Bee Swarm Cognition: Decision-Making Performance and Adaptation*," cognition in the honey bee swarm is discussed. Honey bee swarms possess several key features of cognition in neuron-based brains. In this chapter, Passino focused on analyzing honey bee swarm decision making performance and adaptation. He tested two basic properties of the swarm's choice process: discrimination and distraction. Discrimination is about the ability for honey bees to distinguish qualities of different nest sites; distraction is about the ability for honey bees to ignore nest sites with inferior quality. It was concluded that individual bee-level assessment noise could be effectively filtered out by natural selection and therefore does not adversely affect swarm-level decision-making performance. Simulation tests demonstrate that honey bee swarms do have a cognition process that possesses key features of neuron-based brains.

In the chapter, "*A Theoretical Framework for Estimating Swarm Success Probability Using Scouts*," swarm risk assessment is discussed by using scout agents. It is natural for users to know with a certain level of confidence whether, at least, a portion of a swarm will successfully fulfill a task before the swarm is deployed to fulfill the task. If it can be known that the swarm will not be able to fulfill the task, the swarm should not be assigned to fulfill the task at the very beginning. Under the scenario that a swarm of agents needs to travel from an initial location to a goal location with the capability of avoiding obstacles, Rebguns *et al.* presented a novel theoretical framework for the swarm risk assessment, during which phase it can predict whether a desired percentage of the swarm will succeed in fulfilling the designated task before the swarm is deployed. With the risk information gained, it can then be decided to use which deployment strategy. For example, what is a good start location for the swarm, how to attain a desired success rate by deploying a swarm with a certain size, whether it is worth to take the risk to fulfill the task at hand or not. In the framework, among the swarm a relatively small group of agents is used as expendable "scout" agents to predict the success probability of the whole swarm during the risk assessment phase. The scouts apply the standard Bernoulli trials formula or a novel Bayesian formula proposed by the authors to predict. The experimental results demonstrated that both formulas usually predict better when there is the presence of (Lennard-Jones) inter-agent forces than when their independence assumptions hold.

In the chapter, "*Distributed Multi-Agent Systems for a Collective Construction Task based on Virtual Swarm Intelligence*," a distributed multi-agent system is designed based on virtual swarm intelligence. Under dynamic environments, multi-agent systems should have the capability to create intelligent agents the behaviors of which should be able to adapt to changing environments and the skills of which should be able to improve over time. Meng and Jin proposed a Virtual Swarm Intelligence (VSI)-based algorithm to coordinate a distributed multi-robot system for a collective construction task. The Virtual Swarm Intelligence (VSI)-based algorithm consists of three parts, a Distributed Virtual Pheromone-trail (DVP) based model, a Virtual Particle Swarm Optimization (V-PSO)-based model, and a Quorum Sensing (QS)-based model. The DVP based model is developed for local communication to control the communication costs among the agents in a large-scale system to improve the efficiency of exploration; the Virtual Particle Swarm Optimization (V-PSO)-based model is developed to dynamically allocates agents to different blocks to improve the exploitation capability; the Quorum Sensing (QS)-based model is designed to allows the agents to achieve an optimal balance between exploration and exploitation.

Yuhui Shi
Xi'an Jiaotong-Liverpool University, China
December 30, 2011

Acknowledgment

I would like to thank the chapters' authors. It is them that make the book a reality. I would also like to take this opportunity to thank the National Natural Science Foundation of China and the Suzhou Science and Technology Bureau for their supports under Grant Number 60975080 and SYJG0919, respectively. Finally, I am grateful to the team at IGI Global Publishers who worked diligently with me throughout the process of editing and production. Working with Heather Probst, Jamie Wilson, and Jan Travers has been a pleasure and a learning experience.

Yuhui Shi
Xi'an Jiaotong-Liverpool University, China
December 30, 2011

Table of Contents

Section 2
Other Algorithms