Guest Editorial
Special Issue on Particle Swarm Optimization

PARTICLE swarm optimization (PSO) is one of the evolutionary computation techniques. It is a population-based search algorithm and is initialized with a population of random solutions, called *particles*. Each particle in PSO flies through the search space with a velocity that is dynamically adjusted according to its own and its companion's historical behaviors. The particles have a tendency to fly toward better search areas over the course of a search process. Since its introduction in 1995, PSO has attracted a lot of attention from researchers around the world. A lot of research results have been reported in the literature. Special sessions have being organized in several conferences including the Congress on Evolutionary Computation in 1998. The first book dedicated to PSO was *Swarm Intelligence* (San Mateo, CA: Morgan Kaufmann, 2001) coauthored by J. Kennedy, R. Eberhart, and Y. Shi.

In 2003, the first IEEE Symposium on Swarm Intelligence was held in Indianapolis, IN. More than 60 researchers from all over the world attended the symposium.

The research on PSO generally can be categorized into five parts: algorithms, topology, parameters, merging/combination with the other evolutionary computation techniques, and applications.

Originally, PSO was designed for real-valued problems. The algorithms have now been extended to cover binary and discrete problems. Two commonly used PSOs are the global version and local version of PSO. The two versions differ in the particle's neighborhood, which is generally defined as topologically nearest particles to the particle on each side. In the local version of PSO, each particle's neighborhood includes limited numbers of particles on its sides, while in the global version of PSO, it includes all the particles in the population. The global version of PSO also can be considered as a local version of PSO with each particle's neighborhood being the whole population. It has been reported that the global version of PSO converges fast, but with potential to converge to a local minimum, while the local version of PSO might have more chances to find better solutions slowly. In addition, a lot of different types of neighborhood structures have been designed and studied to improve PSO's performance. Some neighborhood structures that have been looked at include pyramid structure, start structure, "small" structure, and a von Neumann. It is recommended that the PSO with von Neumann structured neighborhood may perform better than PSOs with other regular shaped neighborhoods including the global version and local version. It is also found by some researchers that PSO with small neighborhoods might perform better on complex problems, while PSO with large neighborhoods would perform better for simple problems. Furthermore, some researchers have also looked at the dynamically changing neighborhood structures. Generally, each neighborhood structure has its strengths and weaknesses. It works better on one kind of problem, but worse on another kind of problem.

Velocity changes of a PSO consist of three parts, the "social" part, the "cognitive" part, and the "momentum" part. The balance among these parts determines the performance of a PSO. The first new parameter added into the original PSO algorithm was the inertia weight. The inertia weight is introduced to balance between the global and local search abilities. A large inertia weight facilitates global search, while a small inertia weight facilitates local search. A dynamically changing inertia weight will provide PSO better performance over a fixed value. It can be changed linearly over the course of PSO running or dynamically changed based on some measurements of the PSO performance by, say, a fuzzy rule system. Another parameter called a constriction coefficient is introduced with the hope that it can guarantee a PSO to converge. Mathematically, these two parameters are equivalent.

Another research trend is to merge or combine PSO with the other evolutionary computation techniques. Some researchers introduced operators like selection, crossover, and mutation into the PSO. By applying selection in PSO, the particles with the best performance are copied into the next generation to keep the best performing particles. By applying crossover, a pair of individuals exchange their information to have the ability to "fly" to the new search areas similar to other evolutionary algorithms. The mutation operators are applied with the expectation that the PSO can increase its ability to escape from local minima. On the other hand, some researchers "borrowed" PSO's velocity concept and applied it to evolutionary programming to guide its mutation operations in order to have a fast evolutionary programming algorithm.

PSO is simple in concept, few in parameters, and easy in implementation. It has found applications in many areas. Generally, all the application areas that the other evolutionary computation techniques are good at, are the good application areas for PSO. For example, PSO has been successfully applied in evolving neural networks, for tracking dynamic systems, and for tackling multiobjective optimization and constraint optimization problems. It has also found a lot of industrial applications. For example, PSO has been successfully applied for reactive power and voltage control, and for ingredient mix optimization.

This special issue includes seven papers which cover all five categories discussed above. In the paper, "The Fully Informed Particle Swarm: Simpler, Maybe Better," Mendes *et al.* propose a fully informed PSO algorithm based on φ coefficient analysis and their belief that there is no assumption that the best neighbor actually found a better region than the second- or third-best neighbors. In this new algorithm, all the neighbors of a particle are involved in calculating the next movement instead of using

Digital Object Identifier 10.1109/TEVC.2004.830335

the previous best positions in the original PSO algorithm. The influence of each particle on its neighbors is weighted based on its fitness value and the neighborhood size.

In the paper, "On the Computation of All Global Minimizers Through Particle Swarm Optimization," Parsopoulos and Vrahatis present a modified PSO algorithm to tackle the problem of finding all global minimizers, while avoiding local minimizers. In the modified algorithm, the so-called deflection and stretching techniques, as well as a repulsion technique are incorporated into the original PSO. The deflection and stretching techniques apply the concept of transforming the objective function by incorporating the already found minimizers into the transformed objective functions, while the repulsion technique adds the ability to guarantee that all particles will not move toward the already found minimizers. Therefore, the proposed algorithm can have the ability to avoid already found solutions and, therefore, to have more chances to find all other remaining global minimizers of the function being solved.

In the paper, "A Cooperative Approach to Particle Swarm Optimization," van den Bergh and Engelbrecht implement a cooperative (CPSO). The CPSO employs cooperative behavior to significantly improve the performance of the original PSO algorithm through using multiple swarms to optimize different components of the solution vector cooperatively. Following the suggestion by Potter, the search space is partitioned by splitting the solution vectors into smaller vectors. Two new cooperative PSO models are proposed. One of them, called CPSO-S_K, is a direct extension of Potter's cooperative coevolutionary genetic algorithm (CCGA) to the standard PSO. A swarm with n-dimensional vector is partitioned into n swarms of one-dimensional vectors with each swarm attempting to optimize a single component of the solution vector. A credit assignment mechanism is designed to evaluate each particle in each swarm. The other one, called CPSO-H_K, combines the standard PSO with the CPSO-S_K.

In the paper, "Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients," by Ratnaweera *et al.*, in addition to the time-varying inertia weight, the authors introduce into PSO time-varying acceleration coefficients. Also, a mutation operation is incorporated into PSO, which brings diversity into the population of particles. Furthermore, a PSO called "self-organizing hierarchical particle swarm optimizer" is proposed, in which only the "social" part and the "cognitive" part are kept in the algorithm, while the "momentum" part is only used for reinitializing particles when the particles have stagnated in the search space.

There are three application papers in this special issue. In the paper, "Handling Multiple Objectives With Particle Swarm Optimization," by Coello Coello *et al.*, the authors incorporate Pareto dominance into PSO to solve multiobjective optimization problems. The algorithm stores the nondominated vectors found so far in a second population of particles which are later used by the primary population of particles to update their velocities. An adaptive grid is also introduced to generate well-distributed Pareto fronts. To enhance the exploratory capabilities of the proposed PSO, special mutation operators are designed to mutate both the particles and their dynamic ranges.

In the paper, "Learning to Play Games Using a PSO-Based Competitive Learning Approach," by Messerschmidt and Engelbrecht, three-layer feedforward neural networks are utilized as game-playing agents to play a TicTacToe game. With the assumption of zero expertise in playing the game, these neural networks are trained by the PSO algorithm to predict the desirability of states in the leaf nodes of a game tree. Each neural network, i.e., an individual, is evaluated by playing it against a sample of the other neural networks, i.e., other individuals.

In the paper, "An Approach to Multimodal Biomedical Image Registration Utilizing Particle Swam Optimization," by Wachowiak *et al.*, the authors adapt a PSO algorithm for three-dimensional to three-dimensional biomedical image registration to align images. Here, the PSO is used as a search strategy to maximize the similarity metric for registering single slice biomedical images to 3-D volumes, where the images were obtained from different modalities. Because the users are the skilled clinical professionals and can provide an accurate initial transformation which plays a critical role in the search process, the user knowledge is incorporated in the initialization of the PSO.

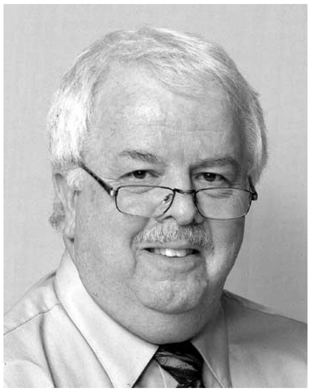
For this special issue, we received abundant responses from researchers. A total of 36 papers were submitted to us. Among them seven papers were accepted and are included in this special issue. This special issue certainly will be a milestone in the research and development of PSO.

ACKNOWLEDGMENT

The Guest Editors would like to thank the past and the current Editors-in-Chief, D. Fogel and X. Yao who helped in making this special issue a success, as well as the referees who added high quality and value to this special issue. Special thanks go to all the authors who submitted their papers to this special issue. It is they who make this special issue a reality.

RUSSELL C. EBERHART, *Guest Editor*
Indiana University–Purdue University Indianapolis
Department of Electrical and Computer Engineering
Indianapolis, IN 46202 USA

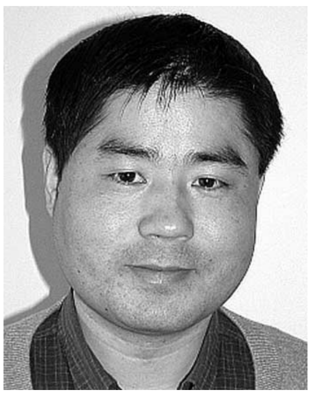
YUHUI SHI, *Guest Editor*
Electronic Data Systems, Inc.
Kokomo, IN 46902 USA



Russell C. Eberhart (M'88–SM'89–F'01) received the Ph.D. degree in electrical engineering from Kansas State University, Manhattan.

He is the Chair and Professor of Electrical and Computer Engineering, Purdue School of Engineering and Technology, Indiana University–Purdue University Indianapolis (IUPUI), Indianapolis, IN. He is coeditor of *Neural Network PC Tools* (San Diego, CA: Academic, 1990), coauthor of *Computational Intelligence PC Tools* (New York: Academic, 1996), coauthor of *Swarm Intelligence* (San Mateo, CA: Morgan Kaufmann/Academic, 2001) with J. Kennedy and Y. Shi, and coauthor of *Computational Intelligence: Concepts to Implementations* (San Mateo, CA: Morgan Kaufmann/Academic, 2004) with Y. Shi. He has published over 120 technical papers.

Dr. Eberhart was awarded the IEEE Third Millenium Medal. In 2002, he became a Fellow of the American Institute for Medical and Biological Engineering.



Yuhui Shi (M'97–SM'98) is an Applied Specialist with Electronic Data Systems, Inc., Kokomo, IN. His expertise is in the areas of computational intelligence, biomedical engineering, embedded systems, and mobile multimedia systems. He is coauthor of *Swarm Intelligence* (San Mateo, CA: Morgan Kaufmann/Academic, 2001) with J. Kennedy and R. C. Eberhart, and coauthor of *Computational Intelligence: Concepts to Implementations* (San Mateo, CA: Morgan Kaufmann/Academic, 2004) with R. C. Eberhart. He has published more than 40 technical papers in the areas of his interests.

Dr. Shi served as the General Chair of the 2003 IEEE Swarm Intelligence Symposium, the Technical Chair of the 2001 Particle Swarm Optimization Workshop, and is a member of the program committees of numerous conferences. He is the Proceedings Chair of the 2004 Congress on Evolutionary Computation (CEC 2004). He is the Co-Chair of the Task Force on Swarm Intelligence, the Evolutionary Computation Technical Committee, the IEEE Neural Networks Society, and an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.