Nat Comput (2008) 7:109–124
DOI 10.1007/s11047-007-9050-z

A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications

Alec Banks · Jonathan Vincent · Chukwudi Anyakoha

Received: 25 August 2006 / Accepted: 4 June 2007 / Published online: 17 July 2007

Abstract Particle Swarm Optimization (PSO), in its present form, has been in existence for roughly a decade, with formative research in related domains (such as social modelling, computer graphics, simulation and animation of natural swarms or flocks) for some years before that; a relatively short time compared with some of the other natural computing paradigms such as artificial neural networks and evolutionary computation. However, in that short period, PSO has gained widespread appeal amongst researchers and has been shown to offer good performance in a variety of application domains, with potential for hybridisation and specialisation, and demonstration of some interesting emergent behaviour. This paper aims to offer a compendious and timely review of the field and the challenges and opportunities offered by this welcome addition to the optimization toolbox. Part I discusses the location of PSO within the broader domain of natural computing, considers the development of the algorithm, and refinements introduced to prevent swarm stagnation and tackle dynamic environments. Part II considers current research in hybridisation, combinatorial problems, multicriteria and constrained optimization, and a range of indicative application areas.

Keywords Particle swarm optimization · Natural computing

1 Introduction

Conventional computing paradigms often have difficulty dealing with real world problems, such as those characterised by noisy or incomplete data or multimodality, because of their

A. Banks (✉)
Tornado In-Service Software Maintenance Team, Royal Air Force, Boscombe Down, Wiltshire, UK
e-mail: test-flt@tismt.raf.mod.uk

J. Vincent · C. Anyakoha
Software Systems Modelling Group, School of Design, Engineering and Computing, Bournemouth University, Poole, Dorset, UK

inflexible construction. Natural systems have evolved over millennia to solve such problems, and, when closely examined, these systems often contain many simple elements that, when working together, produce complex emergent behaviour. They have inspired several natural computing paradigms that can be used where conventional computing techniques perform unsatisfactorily.

This review considers Particle Swarm Optimization (PSO), a relatively recent addition to the field of natural computing, that has elements inspired by the social behaviour of natural swarms, and connections with evolutionary computation. PSO has found widespread application in complex optimization domains, and is currently a major research topic, offering an alternative to the more established evolutionary computation techniques that may be applied in many of the same domains.

Part II of the review is structured as follows. Section 2 briefly reviews the general formulation of PSO. Section 3 reviews the motivations for, and research into, hybrid algorithms, many of which involve evolutionary techniques. Section 4 highlights some recent research into the application of PSO to combinatorial problems. Section 5 considers the use of PSO for multicriteria and constrained optimization, typically associated with practical engineering problems. Section 6 then considers a range of indicative applications. Section 7 concludes.

2 General formulation

PSO (Kennedy and Eberhart 1995) is a simple model of social learning whose emergent behaviour has found popularity in solving difficult optimization problems. The initial metaphor had two cognitive aspects, individual learning and learning from a social group. Where an individual finds itself in a problem space it can use its own experience and that of its peers to move itself toward the solution

$$v_{t+1} = v_i + \varphi_1 \beta_1 (p_i - x_i) + \varphi_2 \beta_2 (p_g - x_i) \tag{1}$$

$$x_{t+1} = x_t + v_{t+1} \tag{2}$$

where constants φ_1 and φ_2 determine the balance between the influence of the individual's knowledge (φ_1) and that of the group (φ_2) (both set initially to 2), β_1 and β_2 are uniformly distributed random numbers defined by some upper limit, $\beta_{\max}$, that is a parameter of the algorithm, p_i and p_g are the individual's previous best position and the group's previous best position, and x_i is the current position in the dimension considered. This was found to suffer from instability caused by particles accelerating out of the solution space. Eberhart et al. (1996) therefore proposed clamping the velocity to a proportion of the maximum particle movement.

However, by far the most problematic characteristic of PSO is its propensity to converge, prematurely, on early best solutions. Many strategies have been developed in attempts to over come this but by far the most popular are inertia and constriction. The inertia term, ω, was introduced thus (Shi and Eberhart 1998):

$$v_{t+1} = \omega v_t + \varphi_1 \beta_1 (p_i - x_i) + \varphi_2 \beta_2 (p_g - x_i). \tag{3}$$

Later work (Eberhart and Shi 2000) indicates that the optimal strategy is to initially set ω to 0.9 and reduce it linearly to 0.4, allowing initial exploration followed by acceleration toward an improved global optimum. Constriction (Clerc and Kennedy developed 1999, published 2002), χ, alleviates the requirement to clamp the velocity and is applied as follows:

$$v_{t+1} = \chi\{v_t + \varphi_1\beta_1(p_i - x_i) + \varphi_2\beta_2(p_g - x_i)\} \tag{4}$$

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|} \quad \text{where } \varphi = \varphi_1 + \varphi_2,\ \varphi > 4. \tag{5}$$

Eberhart and Shi (2000) showed that combining them by setting the inertia weight, ω, to the constriction factor, χ, improved performance across a wide range of problems. Considerable research has been conducted into further refinement of PSO, and areas such as parameter tuning and dynamic environments, discussed further in Part I of this review.

3 Hybrids

Hybridisation is a growing area of intelligent systems research, which aims to combine the desirable properties of different approaches to mitigate their individual weaknesses. A range of PSO hybrids have been postulated, usually in the context of some specific application domain for which that hybrid is particularly well suited. A selection of these approaches is briefly surveyed here.

Hybridisation with Evolutionary Algorithms (EAs), including Genetic Algorithms (GAs), has been a popular strategy for improving PSO performance. With both approaches being population based, such hybrids are readily formulated. Angeline (1998) applied a tournament selection process that replaced each poorly performing particle's velocity and position with those of better performing particles. This movement in space improved performance in three out of four test functions but moved away from the social metaphor of PSO. Brits et al. (2002) used GCPSO (van den Bergh and Engelbrecht 2002) in their niching PSO (NichePSO) algorithm. Borrowing techniques from GAs, the NichePSO initially sets up sub-swarm leaders by training the main swarm using Kennedy's (1997) cognition only model. Niches are then identified and a sub-swarm radius set; as the optimization progresses particles are allowed to join sub-swarms, which are in turn allowed to merge. Once particle velocity has minimised they have converged to their sub-swarm's optimum. The technique successfully converged every time although the authors confess that results were very dependent on the swarm being initialised correctly (using Faure sequences).

One of the main differences between PSO and GAs is the technique used to create new potential solutions. In PSO the individuals move through the solution space through perturbations of their position (which are influenced by other swarm members), in a GA population, members breed with each other to produce new individuals. Løvebjerg et al. (2001) improved convergence speed through the application of breeding, whilst avoiding sub-optimal solutions through introducing subpopulations. This approach suffered in unimodal problems, when compared with the canonical PSO, because new offspring had no knowledge of the group best position and the value to be gained from the social metaphor of shared group knowledge was lost. In multimodal problems, this, in association with the subpopulation technique, was found to be beneficial; the newly produced offspring did not suffer from the cognitive dissonance of the older population and thus did not attempt to cluster about existing solutions.

The development of the genotype to produce a phenotype behaviour that is influenced by the environment was the inspiration behind Krink and Løvebjerg's (2002) Lifecycle model. In this work the authors allowed each candidate solution to decide what the most productive development path might be (PSO, GA or hill climber) depending on its knowledge of its recent success in improving its fitness. Once the decision is made, the individual joins a population of other candidates who share the same development programme. In experiments using common test functions, the Lifecycle model outperformed populations using the individual techniques. Zhang and Xie (2003) also used different techniques in tandem, rather than combining them, in their Differential Evolution (DE) PSO (DEPSO). In this case the DE and canonical PSO operators were used on alternate generations; when DE was in use, the trial mutation replaced the individual best at a rate controlled by a crossover constant and a random dimension selector that ensured at least one mutation occurred each time. The hybrid was found to be successful for some functions, but not all, with results indicating that DEPSO improves on PSO in problems with higher dimensionality.

Gaussian mutation was combined with velocity and position update rules by Higashi and Iba (2003) and was tested on unimodal and multimodal functions. The hybrid achieved better results than those of GA and PSO alone. Juang (2004) also incorporated mutation alongside crossover and elitism. The upper-half of the best-performing individuals, known as elites, are regarded as a swarm and enhanced by PSO. The enhanced elites constitute half of the population in the new generation, whilst crossover and mutation operations are applied to the enhanced elites to generate the other half. This process imitates the natural phenomenon of maturation and outperformed both PSO and GA in the study.

Inspiration for the Cooperative PSO (van den Bergh and Engelbrecht 2004) was provided by the more specialised Cooperative Coevolutionary Genetic Algorithm developed by Potter and de Jong (1994). This aims to minimise the exponential increase of difficulty in optimizing problems with higher dimensions through targeting each dimension as a single dimensional problem. Several cooperative strategies were developed where small swarms tackle each dimension and cross-dimension communication allows the overall solution to move toward the goal. However, this introduced the potential to stagnate due to the serial nature of swarm evaluations and sub-swarms finding pseudo-minima. These problems were countered by combining the cooperative and single swarm approaches. The problem with cooperative swarms is the large increase in algorithmic complexity. The authors avoid this aspect since performance is measured in objective function evaluations rather than execution time. The constricted PSO still compares favourably, especially in problems with lower dimensionality—a prime example of the "No Free Lunch" theorems of Wolpert and Macready (1997).

Liu and Abraham (2005) hybridised a turbulent PSO (TPSO) with a fuzzy logic controller to produce a Fuzzy Adaptive TPSO (FATPSO). The TPSO used the principle that PSO's premature convergence is caused by particles stagnating about a sub-optimal location. A minimum velocity was introduced with the velocity memory being replaced by a random turbulence operator when a particle exceeded it. The fuzzy logic extension was then applied to adaptively regulate the velocity parameters during an optimization run thus enabling coarse-grained explorative searches to occur in the early phases before being replaced by fine-grained exploitation later. The technique was evaluated on problems with low and high dimensionality and found to perform well against both. Notably, whilst the performance of canonical PSO degraded considerably as dimensionality increased, the TPSO and FATPSO remained largely unaffected.

It has been argued that the PSO metaphor is more important than the implementation itself (Kennedy 2003). Poli et al. (2005) took this to the extreme and applied GP to develop more efficient PSO algorithms on a problem-by-problem basis. The rationale behind their approach was that GP could be used to routinely evolve specialist position update algorithms for use by PSO in order to meet the needs of specific problem domains. Using several fitness measures, three of the evolved algorithms were selected for competition against four basic (human derived) algorithms taken from the literature. Using the city block sphere and Rastrigin functions (unimodal and multimodal problems respectively) the evolved algorithms outperformed their competitors. The work does suggest that, as the hardware becomes more powerful, it may be the social metaphor of PSO that is the key and the actual mechanics may be developed on the basis of the requirements of individual applications.

Recently, Jian and Chen (2006) introduced a PSO hybrid with the GA recombination operator and dynamic linkage discovery to optimize difficult real number optimization problems. Dynamic linkage discovery is a technique based on the notion that if the links between the basic building blocks of the objective function can be discovered then optimization of that problem can be improved. The approach first applies linkage discovery to the function before running the PSO with recombination operator for a specified number of generations. This is repeated until fitness improvement checks indicate no further value in applying the linkage discovery at which time the PSO with recombination operator is repeated until a given termination requirement is met. A number of standard objective functions were used in experimentation, and results were promising, with only the toughest problems, containing many local minima, evading solution.

Further hybrid approaches are discussed in later sections, in connection with specific applications.

4 Combinatorial problems

NP-hard combinatorial problems exist in scenarios such as the Travelling Salesman Problem (TSP) and scheduling and have been addressed by a variety of PSO techniques.

The fuzzy discrete PSO (Pang et al. 2004) is a matrix based approach where the TSP is represented as a matrix containing the set of cities and the TSP sequences. The particle position is a fuzzy matrix, the values of which represent the degree of membership to the corresponding elements of the TSP matrix (the particle velocity continues to represent the movement through the solution space but is also formulated as a matrix). Equation 3 is then modified to accommodate the matrix representations of positions and velocities. This approach allows the solution to a discrete problem to be sought in continuous space, since the fuzzy membership is in the real number range [0, 1]. Initial experimentation showed that the technique was viable for such problems, although further development was encouraged.

Lopes and Coelho (2005) combined PSO with Fast Local Search (FLS) and included Genetic Algorithm (GA) concepts. The GA influenced PSO is used to guide the particles at the macro level (exploration), whilst at each iteration the FLS is employed to search for locally improved solutions (exploitation). Experimentation using PSO with and without hybridisation across wide ranging instances of the TSP showed the average excesses above the known optima to be 2.5 and 87%, respectively.

Using a similar ploy, Habibi et al. (2006) also developed a hybrid PSO, this time with Ant Colony (AC) and Simulated Annealing (SA). The AC algorithm replaces the individual best element of PSO, whilst the cooling process of SA is used to control the exploration of the group best element, both of which are then applied within the PSO framework (all random numbers being generated using a Gaussian distribution function). Experimentation with known TSP instances indicated that good approximations can be found efficiently (achieving 100 and 97% of optimality in Burma14 and Berlin52, respectively).

As indicated earlier, discrete problems can be also optimized in continuous space through a suitable mapping of the problem space to the potential solutions generated. Parsopoulos and Vrahatis (2006) applied their previously developed Unified PSO (Parsopoulos and Vrahatis 2004) to the single machine total weighted tardiness problem, by utilising the Smallest Position Value (SPV) mapping mechanism (Tasgetiren et al. 2004). In the SPV scheme the schedule is produced by placing the index of the lowest valued particle component as the first item, the next lowest as the second and so on. For example, a given particle having position (3.23, 1.45, −9.34, 0.01) would represent the potential schedule (3, 4, 2, 1). This potential schedule would then be submitted to the objective function for an assessment of its fitness. In a comparison with the constricted PSO, Eqs. 4 and 5, in fully connected and ring neighbourhood schemes using 40 and 50 task problems, several versions of the Unified PSO, each having differing levels of balance between exploitation and exploration, were found to be competitive, with the most accurate being a variant that also included a level of mutation.

Chen et al. (2006) also utilised SA in a hybrid with a quantum Discrete PSO (DPSO, Yang et al. 2004) for application to the Capacitated Vehicle Routing Problem (CVRP). DPSO is similar to Kennedy and Eberhart's (1997) discrete PSO, in that the particle position represents a probability that a value will be a 1 or 0, but applies a different series of equations to calculate the probability values. To map the CVRP to the DPSO binary particles, each particle has K sections, each having N bits, where each section represents a vehicle and each bit represents a customer. If a customer is to be served by a particular vehicle the corresponding customer bit is set in the section for that vehicle. The hybrid is applied to the problem in a stepwise manner: initially, the DPSO is applied to the particles providing a globally influenced move to new locations; then, at the new locations, SA is applied to each particle to provide a local search, with each particle moving to the best location in its vicinity; the process is then repeated until the stopping criteria are met. In comparisons with GA and SA approaches, the hybrid DPSO showed large improvements in terms of accuracy and efficiency, finding the optimum several times, indicating the technique could be useful for difficult combinatorial problems.

5 Multicriteria and constrained optimization

In the real world, there is often a need to optimize a solution across multiple objectives where the various trade-offs between objectives creates a series of potential solutions across a Pareto Optimal Front. This is typical of engineering design problems. Further, many practical problems introduce constraints such that portions of the search space are invalid. Addressing Multiple Objective (MO) and Constrained Optimization (CO) problems remains an active area of research. PSO can be applied to such problems, but due to its unconstrained characteristics the technique must be modified. The following research can be seen to either redefine the manner in which the local or global best solutions are dealt with and/or to introduce an archiving system to maintain an elite set of the best non-dominated solutions. These techniques can be employed in the application of PSO to the field of Engineering Optimization (EO).

Parsopoulos and Vrahatis (2002b) presented an initial investigation into the potential of PSO to produce a Pareto front using a modified PSO system based on Schaffer's (1985) Vector Evaluated GA. The approach utilised multiple swarms, each targeting one of the objectives, the best particle from one swarm being used as the global best for another. Later work (Parsopoulos et al. 2004) indicated that the approach could be further improved through parallelisation, although performance degraded with larger numbers of swarms due to communication overheads.

At a similar time, Hu and Eberhart (2002a) developed a Dynamic Neighbourhood PSO (DNPSO) in which each particle calculates its fitness and uses the proximity of neighbouring particles against each objective function to deduce its own local best solution. As the particle moves through the problem space the particles in its neighbourhood will change as the Pareto front emerges. A later modification using an archive to maintain a record of Pareto solutions reduced computational time (Hu et al. 2003a). Coello Coello and Lechuga (2002) also produced an alternative method of determining the local best solution by dividing the search space into hypercubes and applying a form of elitism, made possible by a repository of best non-dominated vectors. The approach compared well against two competitive EAs in terms of quality, and bettered them in terms of speed.

The Sigma method was introduced by Mostaghim and Teich (2003a) to redefine the selection of each particle's local best. To do this they calculate a sigma value for each particle (based on the swarm's minimum/maximum objective function solutions) and compare it with the sigma values from an archive of particles (maintained using a clustering technique). The local best is then selected as the archive particle with the closest sigma value. The technique was compared with an EA and was competitive with two objectives, but was not as accurate with three. Further work (Mostaghim and Teich 2003b) improved the performance of the Sigma method by improving the archiving system through the use of ε-dominance.

Zhang et al. (2003) used a more simple averaging method in their determination of global best positions. After calculating the particle best positions for each objective function, the group best for each function is used to find the average group best. The group best is only used where it is closer than the particle best, otherwise a random position between the objective function positions of the particle best is used. The approach works well with two objectives but was not tested at higher levels of complexity.

Lu (2003) produced two PSO/EA hybrids to improve MO PSO. The first uses a dynamic MO EA (Yen and Lu 2002) as the framework, but replaces the crossover and mutation mechanisms with PSO, this had the effect of speeding up the optimization at the expense of the quality of the Pareto front. To overcome the reduction in quality the second hybrid reintroduces the EA crossover mechanism, which in combination with the PSO speeds up convergence and improves the quality of the Pareto front (compared with the original dynamic EA).

Baumgartner et al. (2004) utilised the conventional weighted sum technique to calculate the swarm leader and then Pareto optimality is detected using a two-stage process. First the new particle position is checked for improvement on the previous position; if it is, then a gradient-based algorithm is applied. If the particle is optimal, it is removed and archived. The approach was successfully applied to a magnetics problem.

Sierra and Coello Coello (2005) implemented a crowding based selection to control the number of leading particles, along with mutation and a ε-dominance controlled archive, and thus improved the performance of MO PSO through increased control of particle distribution and the size of the archive. The performance did suffer at the extremes of the Pareto front, but was highly competitive against other MO PSO and MO EAs. Suggested further work targets the archiving as the technique's weakness.

Xiao-hua et al. (2005) applied an Agent-Environment-Rules (AER) model to drive the particles toward the Pareto front. The AER model is inspired by Marvin Minsky's book, *Society of Mind* (1986), and each particle is endowed with the properties required by AER agents; some of which they already possess, such as cooperation, whilst others, such as competition and clonal mutation, are introduced. Experimentation showed the approach could be used to improve PSO solution diversity and convergence across a range of MO problems.

Recent applications of PSO to MO problems have included the design of combinatorial logic circuit (Luna et al. 2004), the design of high performance microwave absorption coatings (Goudos and Sahalos 2006) and the analysis of water for selective withdrawal from a thermally stratified reservoir (Baltar and Fontane 2006).

Hu and Eberhart (2002b) adopted a trial and error approach to CO using the canonical PSO with two modifications: first, particles were only initialised to feasible positions, and, second, only solutions that satisfied the constraints were used for the local and global best positions. The results of experimentation using 12 known test problems indicated that PSO was a viable alternative to GA. Parsopoulos and Vrahatis (2002c) utilised problem dependent dynamic penalty functions taken from existing EA work for CO. Three variants of PSO (inertia only, constriction factor only and a combination thereof) were compared with other EAs that used the same penalty functions and found to be competitive.

Zavala et al. (2005) explored the CO problem of reliability against cost in product design. They proposed a novel PSO that uses a ring topology and a combination of feasibility and domination in the selection of the local best particle. EA inspired perturbations are then applied to the particles to maintain diversity and exploration within the swarm.

PSO has been applied to EO in both single and multi-objective problems. Gaing (2003) applied PSO to solving the economic dispatch problem considering the generator constraints. This is a problem in power system operation with the objective of reducing the total generation cost of units while satisfying constraints of the ramp rate limit and prohibited operating zone. Hu et al. (2003b) used PSO to solve some EO problems: the design of a pressure vessel, welded beam design, minimisation of the weight of a tension/compression spring and Himmelblau's non-linear optimization problem. Their approach involved starting the population with only feasible solutions and the particles keeping only feasible solutions in their memory after updating. Robinson and Rahmat-Samii (2004) applied PSO to the design of a profiled corrugated horn antenna. The PSO also incorporated the invisible wall boundary approach to handle constraints.

6 Applications

PSO has been applied, largely in the research laboratory, to problems from a wide range of domains. In this section, a cross-section of applications of particle systems, aside from that of computer animation, is briefly surveyed, with pointers to the relevant literature for those interested in investigating particular domains in further detail. Utilising PSO as a paradigm can be subdivided into two main approaches: the first exploits its ability to optimize efficiently, which often requires adapting PSO to meet the specific needs of the problem; the second adapts the problem to allow the use of PSO. A third, less common, approach uses the original social metaphor of individual and group learning to provide further insight into how individuals within groups behave.

The strength of Artificial Neural Networks (ANNs) in pattern matching applications is often lessened by problems due to the size of associated optimization aspects or changing environments where the ANN would need to be taken offline for re-training. Conradie et al. (2002) investigated the possibility of augmenting standard neurocontrollers in industrial processes with a PSO based algorithm known as Adaptive Neural Swarming. PSO was selected for its ability to match the weights of the controller to the changing environment without taking the controller off-line, thus balancing the need to adapt with the requirement to generalise. Experimentation using a simulated non-linear bioreactor indicated that profitability could be significantly improved without destabilising the process.

To overcome the speed problem of a practical ANN based face recognition system, caused by the high-dimensionality of the classification stage and the size of the search stage, Sugisaka and Fan (2005) improved the search stage of the recognition process by expressing it as an integer non-linear optimization problem and applying an extended PSO algorithm. During experimentation the approach was found to be competitive, but was outperformed by a system using a more computationally efficient classifier. This suggests that whilst PSO can accelerate the overall process, the search aspect is not the area where most efficiency gains can be made. Other work using PSO in collaboration with ANNs has been carried out by Ismail and Engelbrecht (1999), van den Bergh and Engelbrecht (1999, 2000), Voss and Feng (2001, 2002, 2003), in combination with a Group Method Data Handling ANN (psychological-social metaphor only), Mendes et al. (2002), Xiao et al. (2003), Chen et al. (2004), Georgiou et al.(2004) and Karpat and Özel (2006).

In addition to ANNs, PSO has also been successfully applied to training Neural Fuzzy Networks (NFNs), which are a hybridisation of ANNs with fuzzy controllers. Lin et al. (2006) utilised a PSO hybrid with a local approximation method (to initialise the swarm to an approximate solution) and multi-elites strategy (to slow convergence and thus try to avoid sub-optimal solutions) combined with Recursive Singular Value Decomposition to improve the learning process of a Takagi-Sugeno-Kang type NFN. The hybrid learning algorithm replaced the more usual back propagation algorithm, with each particle representing a potential set of antecedent parameters. Experimentation indicated that the approach offered a more accurate and efficient training regime. In a different approach to NFNs, Mehran et al. (2006) employed PSO to assist with the division of the input space in a locally linear NFN. The input space division is a divide-and-conquer technique, with the division lines being produced by the Local Linear Model Trees (LOLIMOT) learning algorithm, and PSO is applied to the process to reduce the number of redundant local models produced. This is achieved by representing each potential network as a particle (having components that represent the input space partitions); particle movement represents sliding the input space partitions to produce new input space divisions, and the fitness function is the error produced by that model. The use of PSO proved successful in producing smaller errors and consuming fewer models during the learning process.

Colour image quantisation is the process of reducing the number of colours used to reproduce an image. It takes the desired colour for a pixel in the image and attempts to match it as closely as it can to the available colour map. Omran et al. (2005) extended PSO, in a technique known as PSO-CIQ, to match the desired colours with a given colour map.

Their approach used each particle to represent a candidate colour map, thus the swarm represents a number of candidates. Pre-processing of the particles in each time-step using the K-means clustering algorithm is used to reduce the search space before each pixel in the image is assigned to the cluster with the closest match. A mean squared error fitness function is then applied before the canonical PSO is used to move the particles. Experimentation indicated the approach compared well with several other well-known soft computing colour image quantization techniques (self-organising maps and the genetic C-means algorithm).

Recommender systems, software tools designed to make recommendations to users of Internet *e*-commerce sites, are a further area in which the utilisation of PSO as a tuning mechanism has been investigated. Ujjin and Bentley (2003) compared PSO with a GA and a Pearson algorithm as tuners of a collaborative filtering system using a dataset containing 22 features. They found that, in general, the PSO based system was faster, particularly when compared with the GA, and more accurate than its rivals in matching user profiles.

In the field of computational biology, Chang et al. (2004) utilised HPSO-TVAC (with minor modifications to prevent the algorithm from introducing artificial local minima) to improve protein motif discovery. A protein sequence motif is a short sequence of amino acids that exist within a protein family. Being able to identify and classify motifs can assist with the more complex process of classifying unknown proteins. The optimization process encodes the symbolic representations of the acids into numerical form, and the sequences are then subjected to a fitness function that scores the unknown pattern against existing patterns. The approach was compared with a manual method and a combinatorial method with neuro-fuzzy optimization. The modified HPSO-TVAC easily outperformed the former and matched the latter in terms of quality. The PSO based method had the additional strength of speed during the optimization phase (i.e., post-initial motif generation) although premature convergence was cited as problematic. Hsiao et al. (2005) also exploited the speed and efficiency of PSO in a similar biological application, this time to improve bimolecular sequence alignment in much larger DNA or protein sequences. To enable the use of PSO the problem was first adapted to form a suitable solution space. This involved creating particles that consisted of sequence arrays, which were then quality scored by an objective function that produced three performance indices (matched score, mismatched score and gap penalties) before the particles were moved using the PSO algorithm. The technique outperformed other established alignment methods in both speed and matching ability, with the advantages being most notable in sequences with lengths over 2000.

Wachowiak et al. (2004) developed a bespoke PSO system, in which a third social influence was added, to optimize the global search element of biomedical image registration. Image registration is the process of taking several 2-D images from various sources, such as Computer Assisted Tomography (CAT) and Magnetic Resonance Imaging (MRI) scans, and combining them into a 3-D image. The new PSO influence was that of a clinical expert who initially positioned the images in approximately the right vicinity, effectively preventing the swarm from being drawn away from a known good search area by local optima. In experiments the bespoke PSO was also hybridised with several other evolutionary techniques such as crossover and mutation, these further improved the approach and promising results achieved. A further medical science application has been proposed by Li et al. (2005), in which PSO is used to assist in the selection of optimal beam angles in intensity-modulated radiotherapy and thus assisting in safer and more effective treatment of tumours. To achieve this, candidate beam angles are mapped to particles that are optimized to form a beam intensity map, which is then optimized using a conjugate gradient algorithm, and evaluated using a fitness function prior to position update. During simulation based experimentation on representative problems (prostrate and head and neck) the technique was shown to match or improve on results achieved using a GA.

The equipment used in such medical fields is also being affected through the possibility of enhancing engineering design using PSO. Das et al. (2005a) applied a modified PSO (DV-PSO, Das et al. 2005b) to improve the design of Infinite Impulse Response (IIR) filters used in biomedical imaging such as digital mammography. To enable the use of DV-PSO the problem was reformulated as a constrained minimisation problem with each trial solution being presented as a particle in the search space. Results showed the technique to be simpler than other recent approaches, whilst still providing faster and better solutions.

Initial progress has also been made in the study of modelling biomechanical movement to develop understanding in the field of kinesiology. Khemka et al. (2005) used PSO to optimize a biomechanical model of a football kick. The model simulates the 17 different muscle groups involved in kicking a ball, a complex leg movement that results in a 56-dimension search problem, and includes realistic constraints, such as the toes must not hit the floor, which were imposed using heavy fitness penalties. Results were found comparable to those from sports equipment manufacturer research.

Financial forecasting traditionally adopts statistical techniques, with varying degrees of success, and usually on the assumption of linearity. Ko and Lin (2004) adopted a hybrid of PSO with statistical techniques, essentially with the aim of optimizing the inputs to a regression process. The approach, when compared with statistical and GA approaches, was found to offer improved performance. Other financial sector implementations of PSO have been proposed by Nenortaite and Simutis (2004), Kendall and Su (2005) and Nenortaite (2005).

Balci and Valenzuela (2004) improved on the PSO based solution to the Unit Commitment Problem (UCP) presented by Ting et al. (2003) by using PSO combined with Lagrangian Relaxation (LR). The UCP is described as a non-linear, mixed integer combinatorial problem for optimizing the generation of electricity in a competitive power supply market. In this complex hybridisation the PSO is used to search for the optimal Langrangian multipliers that are then used in the sub-problems created in the LR process, which are in turn solved using Dynamic Programming. A standard 10-unit problem was used to compare the approach with other optimization techniques, including PSO, and was found to be several times more efficient whilst maintaining a competitive solution quality. Application to other areas of electric power systems has included voltage control (Yoshida et al. 2001), dynamic security border identification (Kassabalidis et al. 2002), optimal power flow (Abido 2002), distribution state estimation (Naka et al. 2003), post-failure restoration (Jiménez and Cedeño 2003) and generation expansion planning (Kannan et al. 2004).

Foo et al. (2005) reduced the optimal wavelength converter placement problem in wavelength routed optical networks to a binary optimization problem. Working as part of a larger system optimization paradigm, in which the best routing scheme is initially found using either a traffic engineering (TE) aware or equal-cost multipath (ECMP) algorithm, a binary PSO (BPSO) was then applied to locate the optimal position for the converters. The effectiveness of the process was assessed in comparisons of the blocking probability and network efficiency against the number of converters. In terms of blocking probability, BPSO was found to work best with ECMP and in efficiency tests the BPSO process was found to be most useful where large numbers of converters are in use since in small systems all the combinations are generally tried anyway.

The social metaphor of PSO has also been exploited to provide insight into the causes of organisational inertia. Brabazon et al. (2005) produced a model (OrgSwarm) of strategic adaptation in large profit making organisations using the swarm particles to represent organisations and the NK model (Kauffman and Levin, 1987) to represent the strategic landscape. In dynamic environments, in which organisations usually exist, they found that without the application of some inertia and a type of elitism (that ensured only improving moves were made by organisations), the particles performed a random search constantly attempting to relocate the optimal position, thus concluding that some strategic inertia could benefit organisations by preventing unnecessary organisational change.

7 Conclusions

Particle swarm optimization, in its present form, has been in existence for roughly a decade. In that time, it has gathered considerable interest from the natural computing research community and has been seen to offer rapid and effective optimization of complex multidimensional search spaces, with adaptations to multiple objective and constrained optimization. Whilst Part I of this review considered the origins and development of the PSO paradigm, Part II has focussed on recent research in some of the more complex active areas of research: hybridisation, combinatorial problems, multiple objective and constrained optimization. Hybridisation, in particular, is seen as a useful way of combining PSO's strengths, such as rapid optimization, with other useful techniques, particular from the somewhat related field of evolutionary computation, which help alleviate some of the challenging aspects of PSO performance, such as stagnation.

Considerable research has been invested in adapting and refining PSO algorithms to cope with multiple objectives and optimization in the presence of constraints, both of which are important steps to facilitating engineering design optimization and whilst appreciable levels of success have been seen in these areas in recent years, they remain active research topics. The rapid development of the paradigm has led to an explosion in applications. In this review, a variety of domains are briefly touched upon, including the optimization of artificial neural networks, neural fuzzy networks, image processing and medical imaging, computational biology, financial forecasting, optimization of electricity generation and network routing. Naturally, there are many more applications of PSO within the literature, and it has been seen to offer general principles that can be readily adapted to a very wide range of domains.

This paper has offered a timely review of this increasingly important natural computing paradigm. It has found PSO to be highly effective and adaptable to diverse application requirements, with considerable potential for hybridisation and integration into a range of intelligent systems. Challenges remain, in areas such as dynamic environments, avoiding stagnation, handling constraints and multiple objectives, and these are pertinent research foci evident from the literature. Like evolutionary algorithms, PSO has become an important tool for optimization and other complex problem solving. The next decade will no doubt see further refinement of the approach and integration with other techniques, as well as applications moving out of the research laboratory and into industry and commerce. Further understanding of the relative strengths of PSO and other techniques, and the challenges in deploying a PSO based system are required. However, PSO is certainly a welcome addition to the optimization toolbox.

References

Abido MA (2002) Optimal power flow using particle swarm optimization. Int J Elect Power Energy Syst 24(7):563–571

Angeline PJ (1998) Using selection to improve particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, Anchorage, Alaska

Balci HH, Valenzuela JF (2004) Scheduling electric power generators using particle swarm optimization combined with the Lagrangian relaxation method. Int J Appl Math Comput Sci 14(3):411–421

Baltar AM, Fontane DG (2006) A generalized multiobjective particle swarm optimization solver for spreadsheet models: application to water quality. In: Proceedings of the twenty sixth annual American geophysical union hydrology days, 20–22 March 2006

Baumgartner U, Magele C, Renhart W (2004) Pareto optimality and particle swarm optimization. IEEE Trans Magn 40(2):1172–1175

Brabazon A, Silva A, de Sousa TF, O'Neill M, Matthews R, Costa E (2005) Investigating strategic inertia using orgswarm. Informatica 29:125–141

Brits R, Engelbrecht AP, van den Bergh F (2002) A niching particle swarm optimizer. In: Proceedings of the fourth Asia-Pacific conference on simulated evolution and learning

Chang BCH, Ratnaweera A, Halgamuge SK, Watson HC (2004) Particle swarm optimization for protein motif discovery. Genet Program Evolvable Mach 5:203–214

Chen Y, Dong J, Yang B, Zhang Y (2004) A local linear wavelet neural network. In: Proceedings of the fifth world congress on intelligent control and automation, Hangzhou, P.R. China, pp 1954–1957, 15–19 June 2004

Chen A, Yang G, Wu Z (2006) Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. J Zhejiang Univ Sci A 7(4):607–614

Clerc M, Kennedy J (2002) The particle swarm: explosion, stability and convergence in a multi-dimensional complex space. IEEE Trans Evol Comput 6:58–73

Coello Coello CA, Lechuga MS (2002) MOPSO: a proposal for multiple objective particle swarm optimization, in congress on evolutionary computation (CEC'2002), vol 2, IEEE Service Center, Piscataway, New Jersey, pp 1051–1056, May 2002

Conradie AVE, Miikkulainen R, Aldrich C (2002) Adaptive control utilising neural swarming. In: Proceedings of the genetic and evolutionary computation conference, New York, USA

Das S, Konar A, Chakraborty UK (2005a) An efficient evolutionary algorithm applied to the design of two-dimensional IIR filters. In: GECCO 2005: proceedings of the 2005 conference on genetic and evolutionary computation, pp 2157–2163

Das S, Konar A, Chakraborty UK (2005b) Improving particle swarm optimization with differentially perturbed velocity. In: GECCO 2005: proceedings of the 2005 conference on genetic and evolutionary computation, pp 177–184

Eberhart RC, Shi Y (2000) Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the IEEE congress evolutionary computation, San Diego, CA, pp 84–88

Eberhart RC, Simpson P, Dobbins R (1996) Computational intelligence PC tools, chap. 6. AP Professional, San Diego, CA, pp 212–226

Foo YC, Chien SF, Low ALY, Teo CF (2005) New strategy for optimizing wavelength converter placement. Opt Express 13(2):545–551

Gaing Z-L (2003) Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 18(3):1187–1195

Georgiou VL, Pavlidis NG, Parsopoulos KE, Alevizos PhD, Vrahatis MN (2004) Optimizing the performance of probabilistic neural networks in a bioinformatics task. In: Proceedings of the EUNITE 2004 conference, pp 34–40

Goudos SK, Sahalos JN (2006) Microwave absorber optimal design using multi-objective particle swarm optimization. Microwave Opt Technol Lett 48:1553–1558. Published online in Wiley InterScience (http://www.interscience.wiley.com)

Habibi J, Zonouz SA, Saneei M (2006) A hybrid PS-based optimization algorithm for solving traveling salesman problem. In: IEEE symposium on frontiers in networking with applications (FINA 2006), Vienna, Austria, 18–20 April 2006

Higashi N, Iba H (2003) Particle swarm optimization with Gaussian mutation. In: Proceedings of the IEEE swarm intelligence symposium 2003 (SIS 2003), Indianapolis, Indiana, USA, pp 72–79

Hsiao YT, Chuang CL, Jiang JA (2005) Particle swarm optimization approach for multiple biosequence alignment. In: Proceedings of the IEEE international workshop on genomic signal processing and statistics 2005, Rhode Island, USA, 22–24 May 2005

Hu X, Eberhart RC (2002a) Multiobjective optimization using dynamic neighbourhood particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation (CEC 2002), Honolulu, Hawaii, USA

Hu X, Eberhart RC (2002b) Solving constrained nonlinear optimization problems with particle swarm optimization. In: Proceedings of the sixth world multiconference on systemics, cybernetics and informatics 2002 (SCI 2002), Orlando, USA

Hu X, Eberhart RC, Shi Y (2003a) Particle swarm with extended memory for multiobjective optimization. In: IEEE swarm intelligence symposium 2003, Indianapolis, IN, USA

Hu X, Eberhart RC, Shi Y (2003b) Engineering optimization with particle swarm. In: IEEE swarm intelligence symposium 2003, Indianapolis, IN, USA

Ismail A, Engelbrecht AP (1999) Training product units in feedforward neural networks using particle swarm optimization. In: Proceedings of the international conference on artificial intelligence, Durban, South Africa, pp 36–40

Jian M, Chen Y (2006) Introducing recombination with dynamic linkage discovery to particle swarm optimization. In: Proceedings of the genetic and evolutionary computation conference (GECCO2006), pp 85–86

Jiménez JJ, Cedeño JR (2003) Application of particle swarm optimization for electric power system restoration. PowerCON 2003, Special Theme: BLACKOUT

Juang C-F (2004) A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. IEEE Trans Syst Man Cybern – Part B: Cybern 34(2):997–1006

Kannan S, Slochanal SMR, Subbaraj P, Padhy NP (2004) Application of particle swarm optimization technique and its variants to generation expansion planning problem. Elect Power Syst Res 70(3):203–210

Karpat Y, Özel T (2006) Swarm-intelligent neural network system (SINNS) based multi-objective optimization of hard turning. Trans NAMRI/SME 34:179–186

Kassabalidis IN, El-Sharkawi MA, Marks RJI, Moulin LS, Alves da Silva AP (2002) Dynamic security border identification using enhanced particle swarm optimization. IEEE Trans Power Syst 17(3):723–729

Kauffman S, Levin S (1987) Towards a general theory of adaptive walks on rugged landscapes. J Theor Biol 128:11–45

Kendall G, Su Y (2005) A particle swarm optimization approach in the construction of optimal risky portfolios. In: Proceedings of the 23rd IASTED international multi-conference artificial intelligence and applications, Innsbruck, Austria, pp 140–145, 14–16 Feb. 2005

Kennedy J (1997) The particle swarm: social adaptation of knowledge. In: Proceedings of the international conference on evolutionary computation, IEEE, Piscataway, NJ, pp 303–308

Kennedy J (2003) Bare bones particle swarms. In: Proceedings of the IEEE swarm intelligence symposium 2003 (SIS 2003), Indianapolis, Indiana, USA, pp 80–87

Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, Piscataway, NJ, pp 1942–1948

Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: Proceedings of the conference on systems, man and cybernetics, Piscataway, New Jersey, pp 4104–4109

Khemka N, Jacob C, Cole G (2005) Making soccer kicks better: a study in particle swarm optimization. In: Proceedings genetic and evolutionary computation conference (GECCO2005), pp 382–385

Ko PC, Lin PC (2004) A hybrid swarm intelligence based mechanism for earning forecast. In: Proceedings of the second international conference information technology for application

Krink T, Løvbjerg M (2002) The lifecycle model: combining particle swarm optimization, genetic algorithms and hillclimbers. In: Proceedings of parallel problem solving from nature VII (PPSN 2002). Lecture notes in computer science (LNCS) no 2439, pp 621–630

Li Y, Yao D, Yao J, Chen W (2005) A particle swarm optimization algorithm for beam angle selection in intensity-modulated radiotherapy planning. Phys Med Biol 50:3491–3514

Lin C-J, Hong S-J, Lee C-Y (2006) The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition. In: 2006 International joint conference on neural networks, Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, 16–21 July 2006

Liu H, Abraham A (2005) Fuzzy adaptive turbulent particle swarm optimization. In: Proceedings of fifth international conference on hybrid intelligent systems (HIS'05), Rio de Janeiro, Brazil, 6–9 November 2005

Lopes HS, Coelho LS (2005) Particle swarm optimization with fast local search for the blind travelling salesman problem. In: Proceedings of fifth international conference on hybrid intelligent systems (HIS'05), Rio de Janeiro, Brazil, 6–9 November 2005

Løvbjerg M, Rasmussen TK, Krink T (2001) Hybrid particle swarm optimizer with breeding and subpopulations. In: Proceedings of the genetic and evolutionary computation conference (GECCO-2001)

Lu H (2003) Dynamic population strategy assisted particle swarm optimization in multiobjective evolutionary algorithm design, 2003. IEEE Neural Network Society, IEEE NNS Student Research Grants 2002 – Final Reports

Luna EH, Coello Coello CA, Aguirre AH (2004) On the use of a population-based particle swarm optimizer to design combinational logic circuits. In: Zebulum RS, Gwaltney D, Hornby G, Keymeulen D, Lohn J, Stoica A (eds) Proceedings of the 2004 NASA/DoD conference on evolvable hardware. IEEE Computer Society, Los Alamitos, California, pp 183–190, June 2004

Mehran R, Fatehi A, Lucas C, Araabi BN (2006) Particle swarm extension to LOLIMOT. In: Proceedings of the sixth international conference on intelligent systems design and applications (ISDA'06)

Mendes R, Cortez P, Rocha M, Neves J (2002) Particle swarms for feedforward neural network training. In: Proceedings of the 2002 international joint conference on neural networks (IJCNN 2002), pp 1895–1899

Minsky M (1986) The society of mind. Simon and Schuster, New York

Mostaghim S, Teich J (2003a) Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: 2003 IEEE swarm intelligence symposium proceedings, IEEE Service Center, Indianapolis, Indiana, USA, pp 26–33, April 2003

Mostaghim S, Teich J (2003b) The role of ε-dominance in multi objective particle swarm optimization methods. In: Proceedings of the 2003 congress on evolutionary computation (CEC'2003), vol 3. IEEE Press, Canberra, Australia, pp 1764–1771, December 2003

Naka S, Genji T, Yura T, Fukuyama Y (2003) A hybrid particle swarm optimization for distribution state estimation. IEEE Trans Power Syst 18(1):60–68

Nenortaite J (2005) Computation improvement of stockmarket decision making model through the application of grid. Inf Technol Control 34(3):269–275

Nenortaite J, Simutis R (2004) Stocks' trading system based on the particle swarm optimization algorithm. In: Bubak M, van Albada GD, Sloot PMA, Dongarra JJ (eds) Workshop on computational methods in finance and insurance. Computational science – ICCS 2004: 4th international conference. Proceedings, Part IV, Kraków, Poland, 6–9 June 2004

Omran MG, Engelbrecht AP, Salman A (2005) A color image quantization algorithm based on particle swarm optimization. Informatica 29:261–269

Pang W, Wang K, Zhou C, Dong L (2004) Fuzzy discrete particle swarm optimization for traveling salesman problem. In: Proceedings of the fourth international conference on computer and information technology (CIT'04)

Parsopoulos KE, Vrahatis MN (2002b) Particle swarm optimization method in multiobjective problems. In: Proceedings of the ACM symposium on applied computing (SAC 2002), pp 603–607

Parsopoulos KE, Vrahatis MN (2002c) Particle swarm method for constrained optimization problems. In: Proceedings of the Euro-international symposium on computational intelligence 2002

Parsopoulos KE, Vrahatis MN (2004) UPSO: a unified particle swarm optimization scheme. In: Proceedings of the international conference on computational method in science and engineering (ICCMSE 2004). Lecture series on computer and computational sciences. VSP International Science Publishers, Zeist, The Netherlands, pp 868–873

Parsopoulos KE, Vrahatis MN (2006). Studying the performance of unified particle swarm optimization on the single machine total weighted tardiness problem. In: Sattar A, Kang BH (eds) AI 2006, LNAI 4304, Springer-Verlag, pp 1027–1031

Parsopoulos KE, Tasoulis DK, Vrahatis MN (2004) Multiobjective optimization using parallel vector evaluated particle swarm optimization. In: Proceedings of the IASTED international conference on artificial intelligence and applications (AIA 2004), vol 2. ACTA Press, Innsbruck, Austria, pp 823–828, February 2004

Poli R, Langdon WB, Holland O (2005) Extending particle swarm optimization via genetic programming. In: Keijzer M, Tettamanzi A, Collet P, van Hemert J, Tomassini M (eds) Proceedings of eighth European conference, EuroGP 2005. Lausanne, Switzerland, March 30–April 1 2005

Potter MA, de Jong KA (1994) A cooperative coevolutionary approach to function optimization. In: Proceedings of the third conference on parallel problem solving from nature. Springer, Berlin, Germany, pp 249–257

Robinson, Rahmat-Samii (2004) Particle swarm optimization in electromagnetics. IEEE Trans Anten Propagat 52(2):397–407

Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: Genetic algorithms and their applications: proceedings of the first international conference on genetic algorithms, pp 93–100

Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In: Proceedings of the IEEE international conference on evolutionary computation. IEEE Press, Piscataway, NJ, pp 69–73

Sierra MR, Coello Coello CA (2005) Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance. In: Coello Coello CA, Aguirre HA, Zitzler E (eds) Evolutionary multi-criterion optimization. Third International conference, EMO 2005. Lecture notes in computer science, vol 3410. Springer, Guanajuato, México, pp 505–519, March 2005

Sugisaka M, Fan X (2005) An effective search method for neural network based face detection using particle swarm optimization. IEICE Trans Inf Syst E88-D(2):214

Tasgetiren F, Sevkli M, Lian YC, Gencyilmaz G (2004) Particle swarm optimization algorithm for single machine weighted tardiness problem. In: Proceedings IEEE congress on evolutionary computation, pp 1412–1419

Ting TO, Tao MVC, Loo CK, Ngu SS (2003) Solving unit commitment problem using hybrid particle swarm optimization. J Heurist 9(6):507–520

Ujjin S, Bentley PJ (2003) Particle swarm optimization recommender system. In: Proceedings of the IEEE swarm intelligence symposium 2003 (SIS 2003), Indianapolis, Indiana, USA, pp 124–131

van den Bergh F (1999) Particle swarm weight initialization in multi-layer perceptron artificial neural networks. In: Bajic VB, Sha D (eds) Development and practice of artificial intelligence techniques. IAAMSAD, Durban, South Africa, pp 41–45

van den Bergh F, Engelbrecht AP (2000)Cooperative learning in neural networks using particle swarm optimizers. South Afr Comput J (26):84–90

van den Bergh F, Engelbrecht AP (2002) A new locally convergent particle swarm optimizer. In: Proceedings of the IEEE conference systems, man and cybernetics, Hammamet, Tunisia

van den Bergh F, Engelbrecht AP (2004) A cooperative approach to particle swarm optimization. IEEE Trans Evol Comput 3:225–239

Voss MS (2003) Social programming using functional swarm optimization. In: Proceedings of the 2003 IEEE swarm intelligence symposium (SIS03). Purdue University, Indianapolis, Indiana, USA, 24–26 April 2003

Voss MS, Feng X (2001) Emergent system identification using particle swarm optimization. In: Complex adaptive structures conference, Hutchinson Island, FL

Voss MS, Feng X (2002) A new methodology for emergent system identification using particle swarm optimization (PSO) and the group method of data handling (GMDH). In: Proceedings 2002 genetic and evolutionary computation conference, New York, NY, 9–13 July

Wachowiak MP, Smolikova R, Zheng Y, Zurada JM, Elmaghraby AS (2004) An approach to medical biomedical image registration utilizing particle swarm optimization. IEEE Trans Evol Comput 8(3):289–301

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Xiao X, Dow ER, Eberhart RC, Ben Miled Z, Oppelt RJ (2003) Gene clustering using self-organizing maps and particle swarm optimization. In: Proceedings of second IEEE international workshop on high performance computational biology, Nice, France

Xiao-hua Z, Hong-yun M, Li-cheng J (2005) Intelligent particle swarm optimization in multiobjective optimization. In: 2005 IEEE congress on evolutionary computation (CEC'2005), vol 1. IEEE Service Center, Edinburgh, Scotland, pp 714–719, September 2005

Yang SY, Wang M, Jiao LC (2004) A quantum particle swarm optimization. In: Proceedings of the 2004 IEEE congress on evolutionary computation

Yen GG, Lu H (2002) Dynamic population size in multiobjective evolutionary algorithm. In: Proceedings 9th IEEE congress on evolutionary computation, pp 1648–1653

Yoshida H, Kawata K, Fukuyama Y, Takayama S, Nakanishi Y (2001) A particle swarm optimization for reactive power and voltage control considering voltage security assessment. In: Proceedings of power engineering society winter meeting, p 498

Zavala AEM, Diharce ERV, Aguirre AH (2005) Particle evolutionary swarm for design reliability optimization. In: Coello Coello CA, Aguirre AH, Zitzler E (eds) Evolutionary multi-criterion optimization. Third international conference, EMO 2005. Lecture notes in computer science, vol 3410. Springer, Guanajuato, México, pp 856–869, March 2005

Zhang WJ, Xie XF (2003) DEPSO: hybrid particle swarm with differential evolution operator. In: IEEE interenational conference on systems, man and cybernetics (SMCC), Washington DC, USA, pp 3816–3821

Zhang LB, Zhou CG, Liu XH, Ma ZQ, Liang YC (2003) Solving multi objective optimization problems using particle swarm optimization. In: Proceedings of the 2003 congress on evolutionary computation (CEC'2003), vol 4. IEEE Press, Canberra, Australia, pp 2400–2405, December 2003