International Journal of Supply and Operations Management

IJSOM

May 2015, Volume 2, Issue 1, pp. 569-594
ISSN-Print: 2383-1359
ISSN-Online: 2383-2525
www.ijsom.com

An Expert System for Intelligent Selection of Proper Particle Swarm Optimization Variants

Ellips Masehian [a*], Vahid Eghbal Akhlaghi[b], Hossein Akbaripour [a] and Davoud Sedighizadeh[c]

[a] *Industrial Engineering Department, Tarbiat Modares University, Tehran, Iran*
[b] *Industrial Engineering Department, Middle East Technical University, Ankara, Turkey*
[c] *Department of Industrial Engineering, College of Engineering, Saveh Branch Islamic Azad University, Saveh, Iran*

Abstract

Regarding the large number of developed Particle Swarm Optimization (PSO) algorithms and the various applications for which PSO has been used, selecting the most suitable variant of PSO for solving a particular optimization problem is a challenge for most researchers. In this paper, using a comprehensive survey and taxonomy on different types of PSO, an Expert System (ES) is designed to identify the most proper PSO for solving different optimization problems. Algorithms are classified according to aspects like particle, variable, process, and swarm. After integrating different acquirable information and forming the knowledge base of the ES consisting 100 rules, the system is able to logically evaluate all the algorithms and report the most appropriate PSO-based approach based on interactions with users, referral to knowledge base and necessary inferences via user interface. In order to examine the validity and efficiency of the system, a comparison is made between the system outputs against the algorithms proposed by newly published articles. The result of this comparison showed that the proposed ES can be considered as a proper tool for finding an appropriate PSO variant that matches the application under consideration.

Keywords*:* Particle Swarm Optimization; Taxonomy; PSO Variants; Expert System; Knowledge Base.

[*] Corresponding author email address: masehian@modares.ac.ir

1. Introduction

The Particle Swarm Optimization (PSO) algorithm is nowadays one of the most frequently used metaheuristics in solving optimization problems, and as a result, the number of published papers on PSO highly increases each year. This computational evolution model was developed by Kennedy and Eberhart (1995) and has been used in thousands of papers as the basic optimization methodology for solving various sciences and engineering problems. Moreover, hundreds of PSO variants have appeared in the literature during the last two decades, making the PSO one of the most researched and applied optimization algorithms. In fact, PSO has been generalized and extended in many ways, covering various aspects and issues regarding variables, particles, swarm, and process.

The creation of very different PSO-based algorithms, however, has brought about another issue, that is of choosing the best PSO variant or hybrid for a problem under consideration. The idea that which one of the existing algorithms with what parameters is able to solve the problem more efficiently is a concern for each researcher. Plenty of papers have been published so far in which a limited number of PSOs were compared for a specific problem and the best PSO was introduced. For instance, Tao *et al.* (2009) compared the ability of Rotary Hybrid Discrete PSO (RHDPSO) in beating the premature convergence and local optimum issue with Discrete PSO (DPSO) algorithm, and the results showed the lead of RHDPSO. In order to prove the top function of Fractional-Order Darwinian PSO (FODPSO) in solving Multilevel Image Segmentation problems, Ghamisi *et al.* (2013) compared FODPSO, SPSO (Species Based PSO), and basic PSO. However, none of the above papers cover more than a few PSO types and mainly address a specific optimization application, and therefore cannot be used as a guide for researchers in selecting proper PSO-based methods for their needs. On the other hand, no other system has been developed for selecting proper variants of other optimization methods like Genetic Algorithms, Simulated Annealing, Ant Colony Optimization, etc. The reason is probably the fact that metaheuristic method other than PSO have not been extended in various ways and variants as much as the PSO algorithm.

Currently, there is an increasing need for computing machines with abilities of offering knowledge and reasoning, problem solving, and producing logical methods of solving (Waterman, 1986). In this paper, trying to lay out a powerful and intelligent method for finding and selecting the best type of PSO regarding the features of a given research problem, we have proposed using an Expert System (ES), which is by definition a computer program that simulates the thinking way of a specialist in a particular field. In fact, this system identifies the logical patterns of a specialist's mentality and makes decisions according to those patterns.

In existing researches, using an ES for PSO algorithms is just limited to hybridizing the algorithms, which has not been considered as a tool to find the best PSO for solving different problems. For instance, Behera *et al.* (2010) could identify more precise algorithms in comparison with previous algorithms of hybrid PSO-fuzzy expert system (PSOFES) for power quality time series data mining.

In this paper, a comprehensive survey from among 100 different PSOs provides an opportunity for designing an ES to identify the most proper PSO for a given problem as the output. The rest of this Section briefly introduces the PSO and its variants, and Section 2 describes the ES, its components, and details of its operation. In Section 3, the validity of the proposed ES and its performance is discussed, and Section 4 provides conclusion and suggestions.

1.1 Related Work

Particle Swarm Optimization is one of the most significant algorithms in the domain of swarm intelligence (Kennedy and Eberhart, 2001). This algorithm was introduced by Kennedy and Eberhart (1995) and was inspired by the social behavior of animals like fish and birds which live together in small and large groups. In PSO, the population of candidate solutions has direct communication and reaches the solution through exchange of information. PSO fits to different continuous and discrete problems and offers proper answers for different optimization problems. It has gained a wide range of applications in a variety of fields and has been successfully applied, at an increasing rate, to solve several engineering problems. Table 1 briefly introduces some typical applications in each main field (Sedighizadeh and Masehian, 2009b).

Particles are entities in PSO that spread in the functional search space which is subjected to optimization. Each particle calculates the objective function using its own position in the space. Then, by combining the information of its present position, its best previous position, and the information of the best particle of the swarm, a particle selects a vector to move. After moving vector selection by all particles, one iteration of the algorithm comes to an end. Iterations repeat several times so that the proper answer is obtained. In fact, the collection of particles that is in search of optimizing a function act like a flock of birds searching for food. In each iteration, each particle renews its position based on equation (1) and moves to another position. In equation (2), C_1 and C_2 are coefficients of learning that balance the effect of self-knowledge and social-knowledge at the time of the particle's movement to another point in space.

$$prtpos_j^i = prtpos_j^{i-1} + prtvel_j^i \tag{1}$$

$$prtvel_j^i = \chi\left[w \times prtvel_j^{i-1} + c_1 r_1 (pbest_j^{i-1} - prtpos_j^{i-1}) + c_2 r_2 (gbest^{i-1} - prtpos_j^{i-1})\right] \tag{2}$$

In which

$prtpos^i_j$ = The position of the j^{th} particle in i^{th} iteration,

$prtvel^i_j$ = The velocity of the j^{th} particle in i^{th} iteration,

$pbest^i_j$ = The best position of the j^{th} particle,

$gbest^i_j$ = The best position within the swarm,

And

$\chi = 2 / |2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|,\ \varphi = \varphi_1 + \varphi_2,\ \varphi > 4 \cdot$

Table 1. Typical PSO applications and number of related published papers (1995-2008)

Field	Subfield	No. of Pubs.
Electrical Engineering	Electricity generation and power systems	211
	Design and control of neural networks	128
	Control applications	128
	Design and control of fuzzy systems	85
	Electronics and electromagnetic	76
	Design and optimization of communication networks	76
	Image and sound analysis	52
	Antenna design	51
	Design and restructure of electricity networks and economic load dispatching	41
	Sensor networks	40
	Design and optimization of electric motors	23
	Design and control of fuzzy-neural networks	20
	Filter design	17
	Unit commitment	15
	Fault detection and recovery	13
Computer science and Engineering	Visualization and computer graphics	20
	Making music and games	11
Mechanical Engineering	Robotics	74
	Dynamic systems	18
Industrial Engineering	Scheduling	76
	Sequencing	18
	Forecasting	33
	Maintenance planning	8
	Job and resource allocation	7
	Supply chain management	5
Civil Engineering	Civil engineering	5
	Traffic management	5
Chemical Engineering	Chemical process	15
Mathematics	Data mining	155
	Multi objective optimization	97
	Optimization of constrained problems	38
	Multi model function	19
	Modeling	19
	Traveling salesman problem	10
	Combinational optimization	4
Other applications	Miscellaneous	54
	Economical and financial applications	43
	Biological and medical applications	28
	System identification	26
	Material engineering	12
	Security and military applications	3
Total		**1779**

In the literature, the values of both these parameters are considered equal to 2 (Sedighizadeh and Masehian, 2009a). r_1 and r_2 are random numbers in the interval [0, 1] which take on different values in each iteration. Also, χ is the Constriction factor for the velocity of particle's movement. ω is inertia weight, which usually starts with larger values at the onset of process and reduces dynamically during the algorithm. The interval [0.2, 0.4] has also been suggested for ω (Sedighizadeh and Masehian, 2009b).

During the last two decades, new different types of PSO have been introduced, all of which have been derived from the Basic version. Regarding the weak and strong points of each algorithm, their success in reaching an optimal or near-optimal solution is different from problem to problem. For instance, the adaptive fuzzy PSO algorithm by Shi and Eberhart (2001) and a hybrid of PSO and a definite selection procedure (EPSO) by Miranda and Fonseca (2002) were introduced as novel PSO algorithms. In order to find optimal or near optimal solutions, Schoeman and Engelbrecht (2004) offered a vector-based PSO. The Set PSO algorithm was developed to determine Ribonucleic Acid (RNA) secondary structure by Neethling and Engelbrecht (2006). Altinoz *et al.* (2012) did a chaotic search in PSO in order to find local optimum. Immune PSO (IPSO) which is a hybrid of PSO and an Immunology-based optimization method was introduced for solving prediction and control problems (Lin *et al.*, 2008). EMPSO, which is a hybrid of PSO and Electromagnetism-like mechanism (EM), used the strategy of instant update practice velocity that was used to design Functional-link based Petri recurrent fuzzy neural system (FLPRFWS) (Lee *et al.*, 2010). The Continuous Trait-Based PSO (CTB-PSO) was developed by Keedwell *et al.* (2012) as a new variant of PSO, in which individuals within a swarm, as opposed to discrete behavior grouping, have traits based on a continuous scale.

In the following, we introduce some well-known PSO-based algorithms as examples of developments the Basic PSO has undergone after its inception:

Repulsive Particle Swarm Optimization (REPSO) (Lee *et al.*, 2008): This algorithm belongs to the class of stochastic evolutionary optimizers. There are several different realizations of REPSO, and common to all realizations is the repulsion between particles. In the repulsion mechanism, particles move away from positions that are seen as best and thus explore new areas of the search space. This can prevent the swarm from being trapped in local optima, which causes a premature convergence and leads the algorithm to fail in finding the global optimum. When the desired diversity level is reached, the algorithm tries to switch back to the attraction phase to exploit the newly-explored areas.

Particle Swarm Optimization with Passive Congregation (PSOPC) (He *et al.*, 2004): Swarms in nature keep their collective shape under two types of grouping forces: *Aggregation*, and/or *Congregation*. Aggregation may be either (1) *Passive*, in which a passive (not self-moving) swarm moves under a physical force (like a swarm of planktons floating on the water such that the flow of water keeps them together); and (2) *Active*, which is realized by an absorbent source such as food or water. On the other hand, Congregation is the absorbent supply or the group force by self, which is not by external and physical factors. Congregation, too, may be either (1) *Passive*, in which there is an attraction from one particle to others but is not shown through a social behavior; and (2) *Social*, in which there is a social behavior among the particles that strongly relates them to each other. When in some groups there is a selfish behavior in information sharing (like in fish school), that may lead to forming a passive group. A passive swarm model can be added to the PSO in order to increase its efficiency, which results in the PSOPC algorithm.

Negative Particle Swarm Optimization (NPSO) (Yang and Simon, 2005): Negative PSO is a modified form of the Basic PSO with a strategy to avoid a particle's previous worst solution and its group's previous worst based on similar formulae of the Basic PSO. These terms, however, are utilized with negative sign in the velocity updating equation. In other words, this process tries to get farther from the worst instead of getting closer to the best.

In Table 2, the velocity updating formulas of the above-mentioned variants are presented.

Table 2. Comparison of velocity updating equations in BAPSO, REPSO, PSOPC and NPSO

PSO type	Velocity updating equation, $prtvel_j^i$
BAPSO	$w\left(prtvel_j^{i-1}\right)+c_1r_1\left(pbest_j^{i-1}-prtpos_j^{i-1}\right)+c_2r_2\left(gbest^{i-1}-prtpos_j^{i-1}\right)$
REPSO	$w\left(prtvel_j^{i-1}\right)+c_1r_1\left(pbest_j^{i-1}-prtpos_j^{i-1}\right)+wc_2r_2\left(pbest_{rand}^{i-1}-prtpos_j^{i-1}\right)+wc_3r_3\left(prtvel_{rand}\right)$
PSOPC	$w\left(prtvel_j^{i-1}\right)+c_1r_1\left(pbest_j^{i-1}-prtpos_j^{i-1}\right)+c_2r_2\left(gbest^{i-1}-prtpos_j^{i-1}\right)+c_3r_3\left(prtpos_{rand}^{i}-prtpos_j^{i-1}\right)$
NPSO	$w\left(prtvel_j^{i-1}\right)+c_1r_1\left(prtpos_j^{i-1}-pworst_j^{i-1}\right)+c_2r_2\left(prtpos_j^{i-1}-gworst^{i-1}\right)$

2. The Proposed Expert System and its Components

Expert System is an intelligent computer program, which is able to simulate the judgment and behavior of the human with specialized or empirical knowledge in a specific field. It is designed to solve complex problems through rational reasoning like an expert (Waterman, 1986). The proposed ES, after receiving the information about the forward chaining problem, identifies the best hybrid or variant of the standard PSO to solve the problem and represents it along with proposed parameters as the output of the system. As shown in Figure ***1***, each ES consists of three major parts: Knowledge Base, Inference Engine, and User Interface (Giarratano and Riley, 1998). Each of these components has been described in the framework of our developed expert system.

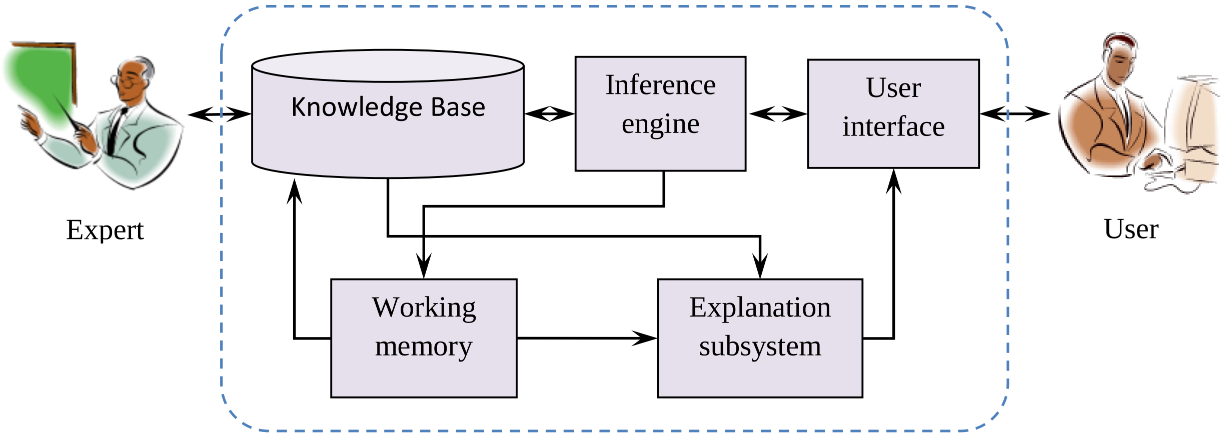

Figure 1. Framework of a typical Expert system

2.1 Knowledge Base

The Knowledge Base includes the knowledge that is used for inference. It is indeed built via acquiring the truths and skills of an expert and then, through a specific procedure, is to be represented for inference (Giarratano and Riley, 1998). Some questions are stored in the inference engine and the responses are searched for in the knowledge base. Clearly, a rich knowledge base will enable the ES to provide answers for user's questions efficiently. After studying 1779 scientific documents related to PSO algorithm published from 1995 to 2008, a comprehensive classification of PSO algorithm-based procedures was introduced by Sedighizadeh and Masehian (2009a), which consists of 22 classes in the form of four groups of Variables, Particles, Swarm, and Process. Details of this taxonomy and brief descriptions of the main attributes of PSO methods are presented in
Table ***3*** and in the rest of this subsection.

Table 3. Taxonomy of the attributes of PSO-based methods

Variables	Constrainment	Constrained/ Unconstrained
	Stochasticity	Deterministic / Stochastic
	Type	Continuous/ Integer / Continuous + Integer
	Velocity Type	Restricted / Unrestricted Velocity / Vertical Velocity / Limited Velocity / Escape Velocity / Adaptive Velocity
	Fuzziness	Fuzzy / Crisp
	Space Continuity	Continuous/ Discrete / Binary
Particles	Accordance	Adaptive/ Dissipative / Adaptive + Dissipative
	Attraction	Attractive/ Repulsive / Attractive + Repulsive
	Association	Aggregation / Passive / Active / Congregation / Passive/ Social
	Dynamics	Newtonian/ Quantum
	Hierarchy	Hierarchical/ Non-hierarchical
	Mobility	Static/ Dynamic
	Synchronicity	Synchronous/ Asynchronous
	Trajectory	Positive/ Negative
Swarm	Cooperation	Cooperative/ Un-Cooperative
	Topology	Gbest/ Lbest/ Pyramid/ Star/ Small World/ Von-Neumann / Random Graphs/
	Activity	Active/ Passive
	Divisibility	Divided / Undivided
Process	Recursiveness	Recursive/ Sideway
	Hybridization	Genetic Algorithms/ Ant Colony Optimization / Differential Evolution / Immune Systems/ Neural Networks
	Objective	Single/ Multiple
	User Interaction	Interactive/ Non-Interactive

2.1.1 Variables

Type – In the classic PSO, all variables take *continuous* real values, whereas in methods like the Combinatorial PSO (CPSO), optimization is done for problems with *mixed* continuous and integer variables.

Fuzziness – Variables in PSO can be either *Crisp* (ordinary) or *Fuzzy*. In order for the PSO to handle fuzzy variables, the vector-from representation of velocity and position variables is transformed from real vectors to fuzzy matrices, as is done in some applications such as multi objective quadratic assignment problem.

Constrainment – Variables in PSO can be *Constrained* or *Unconstrained*. In the classic PSO, velocity and position variables are constrained; that is, their values are kept within upper or lower limits. If through the updating process they exceed these limits, their value will be replaced by the limit values. In some methods such as Unconstrained PSO (UPSO), however, variables are unconstrained and can take any value.

Stochasticity – In probabilistic environments, when multiple swarms or particles cooperate, instead of using a deterministic *gbest*, necessary data is generated by stochastic models, hence introducing uncertainty in information.

Type – The parameter of particle velocity is a main factor in PSO since it specifies the direction of particles' movements. Many researchers have tried to tune this parameter using various heuristics and have obtained better results. Different strategies in this regard are Restricted Velocity, Unrestricted Velocity, Vertical Velocity, Limited Velocity, Escape Velocity, and Self-adaptive Velocity.

Continuity – In terms of the continuity of the space in which the particles are located, PSO methods can be classified into two groups: *Continuous*, and *Discrete*. In the continuous state, a particle's trajectory is changed as its position changes in some dimensions of a continuous space. In the discrete state, this change is discretized. The *Binary* space, however, is a special type of discrete space in which a particle's trajectory is created based on the probability of taking the coordinates of the particles' position a value of 0 or 1.

2.1.2 Particles

Accordance – Sometimes during a PSO run, the swarm evolution process almost comes to a halt. A probable cause is that some particles might have become inactive and unable in doing local and global search, since their positions do not improve due to extremely small velocities. One solution is to adaptively replace these inactive particles with fresh particles such that the existing relations among all particles are maintained, as is done in the *Adaptive* PSO method (for abbreviations of PSO-based methods refer to Table 4). Another reason for the halt might be the swarm's tendency to get into an equilibrium state or a local minimum, which prevents searching further areas. This problem is solved in the *Dissipative* PSO (DPSO) method by introducing negative entropy which triggers chaos among the particles and inhibits their inactivity.

Attraction – In order to prevent premature convergence in PSO, three approaches are generally adopted: *Attractive*, *Repulsive*, and *Attractive-Repulsive*. In the Attractive approach, an additive operator is employed to sum up the terms of the velocity updating equations, whereas in the Repulsive approach, a subtractive operator is utilized. As a result, the particles are attracted to, or repelled from each other, in the Attractive and Repulsive approaches, respectively. In the Attractive-Repulsive approach, the swarm evolution is performed in two successive Attractive and Repulsive phases.

Association – Particles are associated with each other according to two major patterns: *Aggregation*, and *Congregation*. In the Aggregation type, the unifying force of particles is mainly exogenous. It is divided into two subcategories: In *Passive Aggregation*, the swarm lacks any internal force to remain associated and external physical factors keep the particles linked. An example is the planktons floating in water and kept together by the flow of water. In *Active Aggregation*, an absorbing source, such as food or water causes the particles to remain linked. In the Congregation type, particles remain associated due to an endogenous force rather than by external factors. It is also divided into two subcategories: *Passive* type, in which although particles attract each other a social collective behavior is not exhibited, and *Social* type, in which there is a prevailing social behavior among the particles, which are all strongly interrelated.

Mobility – In order to increase the efficiency of the PSO, it is sometimes tried to update the particles' positions through employing dynamic mechanisms. For example, in order to reach a balance between exploitation (focusing the search) and exploration (broadening the search) in the PSO, and also maintain proper particle diversity, in the *Dynamic* and Adjustable PSO (DAPSO) algorithm, each particle's distance to the best position is calculated in each iteration for adjusting the velocity of the particles. In contrast, traditional PSO methods utilize *Static* mechanisms.

Synchronicity – Updating of the particles position and velocity equations can be either *Synchronous* or *Asynchronous*. In the Parallel Asynchronous PSO (PAPSO) method, for instance, the particles' velocity and position updating is performed continuously and based on accessible information. This algorithm designs a dynamic scheme for load-balancing through a duty-centered cyclic approach in order to reduce any unbalanced calculation load.

Dynamics – The particles in classic (and many other) PSO methods move according to the dynamics of classical *Newtonian* mechanics. Sometimes, however, the particles are set to follow *quantum* mechanics. The results of such a motion have been better, especially in high dimensions. The quantum behavior has been particularly adopted for reducing the number of parameters needed for algorithm tuning.

Hierarchy – In the *Hierarchical* approach for the PSO, particles are ordered in a dynamic hierarchical structure such that particles providing high-quality solutions are placed at higher levels of the hierarchy. High-level particles have more effect on the whole swarm.

Trajectory – In calculating the particles' trajectories, there are two main viewpoints, *Positive* and *Negative*. In the positive view (which is the same as classic view), particles adjusts their positions based on their best previous position and the best global position of the swarm. In contrast, in the negative view, particles adjust their positions according to the worst local and global positions by trying to avoid them.

2.1.3 Swarm

Activity – When there is attraction between the particles of a swarm, two different behaviors may occur in the swarm: in *Active* state, a collective behavior is communicated in the whole swarm, whereas in *Passive* state no significant and consistent behavior is observed in the swarm.

Topology – In PSO, the particles' accessibility to the information within the swarm can take on different schemes or topologies. In the *Gbest* topology (not to be confused with *gbest*), all particles are interrelated and affect each other. In the *Lbest* topology, each particle is related to only its neighboring particles, and a communication loop thus is formed. *Pyramid* is another topology which embodies the relations between the particles in 3D. In the *Star* topology a central node is affected by and effects on the whole population of particles. The *Small World* topology is a graph made up of isolated sub-swarms and particles and is in fact an instance of heterogeneity. In the *Von-Neumann* topology, the all four up, down, left, and right neighbors of a particle are located on a cycle in a 2D space. In addition to the above topologies, there are also *Random* Graph topologies created without a specific predefined structure.

Divisibility – In some PSO-based algorithms, for enhancing the algorithm's efficiency, increasing the swarm's diversity, or improving its multi objectiveness, the main swarm is divided into a number of sub-swarms. In this case the particles become *Divided*, and otherwise, *Undivided*.

Cooperation – In order to improve the performance of the classic PSO, different swarms may be used cooperatively to optimize various components of the problem, as in the *Cooperative* Co-evolutionary PSO (CCPSO) method. Otherwise, with a single swarm, the case is *Uncooperative*.

2.1.4 Process

Problem Objectives – The classic PSO can only solve *single objective* problems. However, some PSO-based methods have been developed for solving *multi objective* optimization problems, by trying to optimize several objectives using one swarm, according to the priority of the objectives.

Recursiveness – The PSO process can be *Recursive* or *Sideway*. In a recursive process, the process is adapted with current conditions through a feedback mechanism. The Sideway (one-directional) process, however, lacks a feedback mechanism and does not respond adaptively.

User Interaction – Interactive Evolutionary Computation (IEC) is an approach in Evolutionary Computation (EC) methods in which the particles' fitness functions are modified or replaced by the user's judgment. That is, the user gives opinion about each particle by taking into consideration available criteria. Trying to integrate expert opinions of users, the *Interactive* PSO (IPSO) has been developed. In IPSO, unlike in EC and IEC, the information of particles positions disseminates through the swarm throughout successive iterations, and is not limited to just one epoch. Therefore, the identification of the best particle is done by the user and not using the fitness function.

Hybridization – In order to increase the efficiency of the PSO, overcome the problem of trapping in local optima, and find better solutions by increasing the diversity of the search, the PSO has often been combined with other optimization methods, creating *Hybrids* with metaheuristics such as SA, GA, ACO, etc.

A vast range (hundred) of PSO-based algorithms are introduced in Table 4 (Sedighizadeh and Masehian, 2009b), which form the basis of our developed expert system and has a close connection with the classification shown in Table3. These two Tables provide a rich knowledge-base for every researcher in relation to PSO algorithms. For instance, in the class of problem *Space Continuity* in ***Table 3***, three states have been indicated: continuous, discrete, and binary, and on the other hand, the space continuity of each PSO-based algorithm in Table 4 matches one of these states.

Table 4. Major PSO-based methods.

Active target PSO (APSO) (Zhang *et al.*, 2008)	Hybrid Gradient PSO (HGPSO) (Noel and Jannett, 2004)
Adaptive Dissipative PSO (ADPSO) (Shen *et al.*, 2007)	Hybrid Recursive PSO (HRPSO) (Ching *et al.*, 2007)
Adaptive Mutation PSO (AMPSO) (Pant *et al.*, 2008)	Hybrid Taguchi PSO (HTPSO) (Roy and Ghoshal, 2006)
Adaptive PSO (APSO) (Xie *et al.* 2002a)	Immune PSO (IPSO) (Lin *et al.*, 2008)
Adaptive PSO Guided by acceleration information (AGPSO) (Zeng *et al.*, 2006)	Improved PSO (IPSO) (Zhao *et al.*, 2006)
Angle Modulated PSO (AMPSO) (Pampara *et al.* 2005)	Interactive PSO (IPSO) (Madar *et al.*, 2005)
Area Extension PSO (AEPSO) (Atyabi and Phon-Amnuaisuk, 2007)	Map Reduce PSO (MRPSO) (McNaab *et al.*, 2007)
Attractive-Repulsive PSO (ARPSO) (Riget and Vesterstroem, 2002)	Modified Binary PSO (MBPSO) (Yuan and Zhao, 2007)
Augmented Lagrangian PSO (ALPSO) (Sedlaczek and Eberhard, 2006)	Modified GPSO (MGPSO) (Zhiming *et al.*, 2008)
Basic PSO (BAPSO) (Lam *et al.*, 2007)	Nbest PSO (Britis *et al.*, 2002)
Behavior of Distance PSO (BDPSO) (Hui and Feng, 2007)	Negative PSO (NPSO) (Yang and Simon, 2005)
Best Rotation PSO (BRPSO) (Alyiar *et al.*, 2007)	Neural PSO (NPSO) (Brits *et al.*, 2002)
Binary PSO (BPSO) (Moraglio *et al.*, 2008)	New PSO (NPSO) (Zhang and Mahfouf, 2006)
Chaos PSO (CPSO) (Mo *et al.*, 2006)	New PSO (NPSO) (Yang and Simon, 2005)
Combinatorial PSO (CPSO) (Jarbouia *et al.* 2007)	Niche PSO (Brits *et al.*, 2005)
Comprehensive Learning PSO (CLPSO) (Liang *et al.*, 2006)	Novel Hybrid PSO (NHPSO) (Li and Li, 2007)
Constrained Optimization Via PSO (COPSO) (Aguirre *et al.*, 2007)	Novel PSO (NPSO) (Zhang *et al.*, 2008)

Table 4. Continued

Continuous Trait-Based PSO (CTB-PSO) (Keedwell *et al.*, 2012)	Optimized PSO (OPSO) (Meissner *et al.*, 2006)
Cooperative Co-evolutionary PSO (CCPSO) (Yao, 2008)	Orthogonal PSO (OPSO) (Ho *et al.*, 2008)
Cooperative Multiple PSO (CMPSO) (Felix *et al.*, 2007)	Parallel Asynchronous PSO (PAPSO) (Koh *et al.*, 2005)
Cultural Based PSO (CBPSO) (Jingbo and Hongfei, 2005)	Parallel Vector-Based PSO (PVPSO) (Brits *et al.*, 2005)
Discrete PSO (DPSO) (Kennedy and Eberhart, 1997)	Perturbation PSO (PPSO) (Yuan *et al.*, 2005)
Dissipative PSO (DPSO) (Xie *et al.*, 2002b)	Predator Prey PSO (PPPSO) (Jang *et al.*, 2007)
Divided Range PSO (DRPSO) (Ji *et al.*, 2004)	Principal Component PSO (PCPSO) (Voss, 2005)
Double Structure Coding Binary PSO (DSBPSO) (Lam *et al.*, 2007)	PSO with Craziness and Hill Climbing (CPSO) (Ozcan and Yilmaz, 2006)
Dual Layered PSO (DLPSO) (Subrarnanyam et al,. 2007)	PSO with Passive Congregation (PSOPC) (He *et al.*, 2004)
Dynamic and Adjustable PSO (DAPSO) (Liao *et al.*, 2007)	Pursuit-Escape PSO (PEPSO) (Higashitani *et al.*, 2008)
Dynamic Double PSO (DDPSO) (Cui *et al.*, 2004)	Quadratic Interpolation PSO (QIPSO) (Pant *et al.*, 2007)
Dynamic Neighborhood PSO (DNPSO) (Hu *et al.*, 2002)	Quantum Delta PSO (QDPSO) (Sun *et al.*, 2004)
Enhanced Leader PSO (ELPSO) (Jordehi, 2015)	Quantum PSO (QPSO) (Yang *et al.*, 2004)
Escape Velocity PSO (EVPSO) (Wang and Qian, 2007)	Quantum-Inspired PSO (QIPSO) (Sun *et al.*, 2004)
Estimation of Distribution PSO (EDPSO) (Kulkarni and Venayagamoorthy, 2007)	Repulsive PSO (REPSO) (Lee *et al.*, 2008)
Evolutionary Iteration PSO (EIPSO) (Lee, 2007)	Restricted Velocity PSO (RVPSO) (Lu and Chen, 2006)
Evolutionary Programming PSO (EPPSO) (Wei *et al.*, 2002)	Self-Adaptive Velocity PSO (SAVPSO) (Lu and Chen 2008)
Evolutionary PSO (EPSO) (Shi and Krohling, (2002)	Self-Organization PSO (SOPSO) (Jie *et al.*, 2006)
Exploring Extended PSO (XPSO) (Poli *et al.*, 2005)	Simulated Annealing PSO (SAPSO) (Wang and Li, 2004)
Extended PSO (EPSO) (Poli *et al.*, 2005)	Spatial Extension PSO (SEPSO) (Krink *et al.*, 2002)
Fast PSO (FPSO) (Li and Li, 2007)	Special Extension PSO (SEPSO) (Monson and Seppi, 2005)
Fully Informed PSO (FIPS) (Poli *et al.*, 2005)	Species Based PSO (SPSO) (Li, 2004)
Fuzzy PSO (FPSO) (Shi and Eberhart, 2001)	Sub-Swarms PSO (SSPSO) (Wang and Qian, 2007)
Gaussian PSO (GPSO) (secrest and Lamont, 2003)	Trained PSO (TPSO) (Gheitanchi *et al.*, 2008)
Genetic Binary PSO (GBPSO) (Sadri and Suen, 2006)	Two dimensional Otsu PSO (TOPSO) (Wei *et al.*, 2007)
Genetic PSO (GPSO) (Yin, 2006)	Two-Swarm PSO (TSPSO) (Li *et al.*, 2006)
Geometric PSO (GPSO) (Moraglio *et al.*, 2008)	Unconstrained PSO (UPSO) (Moore and Venayagamoorthy, 2006)
Greedy PSO (GPSO) (Lam *et al.*, 2007)	Unified PSO (UPSO) (Parsopoulos and Vrahatis, 2008)
Gregarious PSO (GPSO) (Pasupuleti and Battiti, 2006)	Variable Neighborhood PSO (VNPSO) (Pasupuleti and Battiti, 2006)
Heuristic PSO (HPSO) (Lam *et al.*, 2007)	Vector Evaluated PSO (VEPSO) (Omkar *et al.*, 2008)

Table 4. Continued

Hierarchical Recursive-based PSO (HRPSO) (Feng, 2005)	Velocity Limited PSO (VLPSO) (Xu and Chen, 2006)
Hybrid Discrete PSO (HDPSO) (Chandrasekaran, 2006)	Velocity Mutation PSO (VMPSO) (Xu *et al.*, 2008)
Hybrid GA–PSO (HGAPSO) (Gálvez and Iglesias, 2013)	Vertical PSO (VPSO) (Yang, 2007)

Knowledge representation is usually in the form of either rule-based or object-oriented, which is then converted to a computer language known as *coded knowledge*. In this paper, knowledge representation is done in rule-based method and the program for running the code has been the VP-Expert™ software. Table 5 following explains how the knowledge is produced and rules are formed, regarding the introduced algorithms in the Table 4. Details of 20 sample rules out of the 64 rules in the knowledge base are presented in the Table. For instance, Rule 1 indicates that if the Basic PSO is combined with the Harmony Search algorithm and the aim is to solve high-dimensional problems and the solution space is continuous, then the most appropriate PSO-based algorithm will be Novel Hybrid PSO which was developed by Li and Li (2007). Among the outputs of the ES are W_2 and W_1, which are the weights of low and high inertia rates, respectively.

Table 5. Sample rules used in the proposed expert system.

Rule 1 **IF** Hybridization = Harmony-Search **AND** Purpose = Solving_High_Dim_Problem **AND** Type_of_Space = Continuous **THEN** Type_of_PSO=Novel_Hybrid_PSO by Li_Li_2007 C_1=1.5, C_2=1.5, W_1=0.73, W_2=0.73;	***Rule 2*** **IF** Purpose = Computing_compressed_function **AND** Hybridization = None **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Map_Reduce_PSO_McNabb_et_al_2007 C_1=2, C_2=2, W_1=0.2, W_2=0.4;
Rule 3 **IF** Purpose = Design_a_Neural_Network **AND** Hybridization = None **AND** Type_of_Space = Continuous **THEN** Type_of_PSO = Dual_Layered_PSO by Subrarn_2007 DISPLAY "C_1=C_1+(n/$Iter\ max$)" DISPLAY "C_2=C_2−(n/$Iter\ max$)" W_1 = There_is_no_suggestion W_2 = There_is_no_suggestion;	***Rule 4*** **IF** Purpose = Ad_hoc_computing_networks **AND** Hybridization = None **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Trained_PSO__Gheitanchi_et_al_2007 C_1 =There_is_no_suggestion C_2 = There_is_no_suggestion W_1 = There_is_no_suggestion W_2 = There_is_no_suggestion;
Rule 5 **IF** Purpose= Solving_Scheduling **AND** Hybridization=None **AND** Type_of_Space = Binary **THEN** Type_of_PSO = Hybrid_Discrete_PSO by Chandrasek_2006 C_1=2, C_2=2, W_1=0.2, W_2=0.2;	***Rule 6*** **IF** Purpose = Avoid_quick_convergence **AND** Hybridization = None **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Escape_Velocity_PSO_Wang_et_al_ 2006 C_1=1.49, C_2=1.49, W_1=0.7, W_2=0.7;

Table 5. Continued

Rule 7 **IF** Purpose = Avoid_trapping_local **AND** Mobility = Dynamic **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Dynamic_Double_PSO_Cui_2004 C_1=1.8, C_2=1.8, W_1=0.4, W_2=1;	***Rule 8*** **IF** Purpose = Solving_high_dimensinal_problems **AND** Hybridization = intelligent_move_mechanism **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Orthogonal_PSO_Ho_et_al_2008 C_1=2, C_2=2, W_1=0.9, W_2=0.9;
Rule 9 **IF** Landscape = Multimodal **AND** Type_of_space = Continuous **AND** Hybridization = None **AND** Divisibility = Divided **THEN** Type_of_PSO=Best_Rotation_PSO_Barrera_2007 C_1=2, C_2=2, W_1=0.4, W_2=0.4;	***Rule 10*** **IF** Velocity_Type = Restricted_Velocity **AND** Hybridization=None **AND** Purpose=no_specific_purpose **AND** Type_of_space = Continuous **THEN** type_of_PSO=Restricted_Velocity_PSO_Lu_2006 C_1=1, C_2=1, W_1=1, W_2=1;
Rule 11 **IF** Fuzziness_of_variable = Fuzzy **AND** Type_of_space = Continuous **THEN** Type_of_PSO=Fuzzy_PSO_Shi_Eberhart_2001 C_1=2, C_2=2, W_1=3, W_2=3;	***Rule 12*** **IF** Purpose = Employee_feedback **AND** Type_of_space = Continuous **THEN** Type_of_PSO= Self_Organization_PSO_Jie_2006 C_1=1.5, C_2=1.5, W_1=0.73, W_2=0.73;
Rule 13 **IF** Type_of_space = Continuous **AND** Hybridization=Evolutionary_Programming **AND** **THEN** Type_of_PSO = Greedy_PSO_He_et_al_2007 C_1=C_2 = There_is_no_suggestion W_1=W_2 = There_is_no_suggestion;	***Rule 14*** **IF** Purpose = Movement_of_robot **AND** Hybridization = None **AND** Type_of_space = Continuous **THEN** Type_of_PSO=Area_Extension_PSO_Atyabi_Phon_2007 C_1 = 0.5, C_2 = 2.5, W_1 = 0.2, W_2 = 1;
Rule 15 **IF** Purpose = Opt_PSO_parameters **AND** Hybridization=None **AND** Type_of_space = Continuous **THEN** Type_of_PSO=Optimized_PSO_Meissner_et_al_2006 C_1 = There_is_no_suggestion C_2 = There_is_no_suggestion W_1 = There_is_no_suggestion W_2 = There_is_no_suggestion; DISPLAY "((C_2/C_1) = 2.14)"	***Rule 16*** **IF** Purpose = Enhance_diversity **AND** Hybridization = None **AND** Type_of_space = Continuous **THEN** Type_of_PSO=Sub_Swarms_PSO_Wang_Qian_2007 C_1 = There_is_no_suggestion C_2 = There_is_no_suggestion W_1 = There_is_no_suggestion W_2 = There_is_no_suggestion DISPLAY "C_1+C_2 = 4";

Table 5. Continued

Rule 17 **IF** Purpose = Avoid_trapping_local **AND** Hybridization = None **AND** Divisibility = Divided **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Pursuit_Escape_PSO_Higashitani_2008 C_1 = 1.5, C_2 = 1.5, W_1 = 0.729, W_2 = 0.729;	***Rule 18*** **IF** Purpose =Avoid_trapping_local **OR** Purpose = Avoid_quick_convergence **AND** Hybridization = None **AND** Type_of_space=Continuous **THEN** Type_of_PSO = Two_Swarm_PSO_Li_et_al_2006 C_1 =2.05, C_2 =2.05, W_1 =0.729, W_2 =0.729;
Rule 19 **IF** Purpose = Reducing_time_complexity **AND** Hybridization = None **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Principal_Component_PSO_Voss_2005 C_1=2, C_2=2, W_1=0.4, W_2=0.7;	***Rule 20*** **IF** Purpose = Avoid_quick_convergence **AND** Divisibility=Divided **AND** Objective=Multiple **AND** Type_of_space = Continuous **THEN** Type_of_PSO = Spatial_Extension_PSO_Krink_2002 C_1=2, C_2=2, W_1=0.6, W_2=0.9;

2.2 Inference Engine

To obtain an intended result, it is not sufficient to merely represent the knowledge with the help of rules, but another system is needed to facilitate inference by searching through the represented knowledge, analyzing the rules, receiving more information from the user, and applying the logic principles. This system is called Inference Engine.

The Expert System infers by searching in the knowledge base, according to the inference logic and analysis of rules. The engine does inference based on an either *forward chaining*, *backward chaining*, or a combination of both. When the rules are surveyed by the inference engine, the required orders will be performed if the information given by the user is confirmed by the rules. The inference engine used in this paper is the VP-Expert software. After representing the knowledge in the form of rule, it is converted to a language understandable to the VP-Expert software, which will perform the inference.

2.3 User Interface

User Interface provides connection between the ES and the user. Not only does an ES interface enable the user to answer questions, but permits him/her to interrupt the system operation by asking about the given explanations. It should be noted that expert systems can vary based on the expert(s) from whom the knowledge is extracted, and their application (Medeiros *et al.*, 2008).The user can enter information about the problem to the system by answering the system's questions. The knowledge base, questions, multiple-choice answers, and the rules for identifying the best PSO are all constructed in the database.

Figure 2 illustrates a snapshot of the VP-Expert software during its interaction with the user while answering some questions about each input parameter so that the system can do the inference according to the received answers. Also, two examples of complete interaction between the ES and the user through the User Interface are provided in Figure 3.

Figure 2. A snapshot of the VP-Expert software at the time of its interaction with the user

3. Validation of the Proposed Expert System

While the main motivation and purpose of this paper is to identify the most suitable PSO variant that corresponds to the characteristics of a given problem, in order to enrich the ES output, some supplementary information have also been provided. For example, the names of authors, dates of publications, and more importantly, the applied parameters are supplied as some extra but practical information for the user.

In order to evaluate the validity and efficiency of the output of the developed expert system, it was compared with the contents of recently published papers on PSO. For this purpose, 50 papers published from 2009 to 2013 on different PSOs were randomly selected. The system assessment process was not only done for the type of PSO, but also for parameters of C_1, C_2, W_1 and W_2. **Error! Reference source not found.** 6 shows the comparison result. Out of the compared 50 instances, 36 results (72%) were identical with those proposed in the original papers. Also, among those 36 papers, 27 cases (75%) applied precisely the same parameter values in their studies. Of course this does not mean that the parameter values prescribed by the Expert System are optimal, as in some cases researchers prefer to re-tune the parameters considering their problem features, aiming to increase the performance of their algorithms. Nevertheless, we believe that suggesting such additional parametric information can serve as a useful starting point for future researches.

Table 6. A comparison between the outputs of the expert system and the PSOs proposed by recently-published papers

No	Hybridization	Space	objective	Application/ Purpose	Other criteria	ES output	Proposed PSO
1	None	C	S	Micro-electronics	Dynamics = Quantum	Quantum PSO (Yang et al, 2004)	Quantum PSO
2	None	C	M	Multiple optimization	–	Niche PSO (Brits et al, 2002), C_1=C_2=1.2, W_1=0.2, W_2=0.7	Evolutionary Programming PSO, C_1=C_2=2, W_1=0.35, W_2=0.95
3	None	B	S	Scheduling	–	Hybrid Discrete PSO (Chandrasek, 2006), C_1=C_2=2, W_1=W_2=2	Hybrid Discrete PSO, C_1=C_2=2, W_1=W_2=omitted
4	Simulated Annealing	C	S	MIMO	–	Simulated Annealing PSO (Wang Li, 2004), C_1=C_2=2	Simulated Annealing PSO C_1=C_2=1
5	None	C	S	Multimodal Functions	Mobility = Dynamic	Dynamic and Adjustable PSO (Liao 2007)	Memetic PSO, C_1=C_2=1.492, W=72984
6	None	C	S	Movement of robot	–	AEPSO (Atyabi Phon, 2007), C_1=0.5, C_2=2.5, W_1=0.2, W_2=1	AEPSO, C_1=0.5, C_2=2.5, W_1=0.2, W_2=1
7	Simulated Annealing	C	S	supply chain management	–	Simulated Annealing PSO (Wang Li, 2004), $C1$=$C2$=2	Simulated Annealing PSO C_1=C_2=1.49, W=639
8	None	D	S	No specific purpose	–	Dissipative PSO (Xie et al, 2002), C_1=C_2=2, W_1=0.4, W_2=0.9	Dissipative PSO, C_1=C_2=2, W=0.4
9	None	D	S	Gene expression data	–	Modified Binary PSO, (Yuan Zhao, 2007), C_1=C_2=0.33, W_1=W_2=0.33	Modified Binary PSO, C_1=C_2=2, W_1=0.8, W_2=2
10	Harmony search	C	S	Solving high dimensional problems	–	Novel Hybrid PSO (Li Li, 2007), C_1=C_2=1.5, W_1=W_2=0.73	Novel Hybrid PSO, C_1=C_2=1.5, W_1=W_2=0.73
11	None	C	M	No specific purpose	Containment of variables = Constrained	Constrained PSO (Aguirre et al, 2007), C_1=C_2=1, W_1=0.5, W_2=1	Constrained PSO, C_1=C_2=1, W_1=0.5, W_2=1
12	Immune algorithm	C	S	No specific purpose	–	Immune PSO (Lin et al 2008), C_1=C_2=2, W_1=0.4,W_2=0.9	Immune PSO, C_1=C_2=2, W_1=0.4, W_2=0.9
13	None	B	S	Power systems	–	Angle Modulated PSO (Pampara, 2005), C_1=C_2=2, W_1=W_2=0.8	Angle Modulated PSO, C_1=C_2=2, W_1=W_2=0.8
14	None	C	M	Avoid quick convergence	Divisibility = Divided	Spatial Extension PSO, C_1=C_2=2, W_1=0.6, W_2=0.9	Spatial Extension PSO, C_1=C_2=2.05
15	None	C	S	Enhance diversity	–	Sub Swarms PSO (Wang Qian 2007)	Genetic PSO

Table 6. Continued

16	None	C	S	Employed to CLPSO	–	Comprehensive Learning PSO (Liang, 2006), C_1=C_2=2, W_1=0.4, W_2=0.9	Comprehensive Learning PSO, C_1=C_2=2, W_1=0.4, W_2=0.9
17	None	C	S	Avoid quick convergence	Divisibility = Divided	Pursuit Escape PSO (Higashitani 2008), C_1=C_2=1.5, W_1=W_2=0.729	Pursuit Escape PSO, C_1=C_2=1.5, W_1=W_2=0.729
18	None	C	S	Avoid quick convergence	Divisibility=Divided	Pursuit Escape PSO (Higashitani 2008), C_1=C_2=1.5, W_1=W_2=0.729	Chaos PSO, C_1=C_2=1, W_1=0.2, W_2=1.2
19	None	C	S	Ad hoc computing networks	–	Trained PSO (Gheitanchi et al 2007)	Standard PSO
20	None	C	S	No specific purpose	Landscape = Multimodal Divisibility = Divided	Best Rotation PSO (Barrera 2007), C_1=C_2=2, W_1=W_2=0.4	Best Rotation PSO, C_1=C_2=2, W_1= W_2=0.4
21	None	C	S	Reducing Time Complexity	–	Principal Component PSO (Voss_2005), C_1=C_2=2, W_1=0.4, W_2=0.7	Discrete PSO, C_1=0.3, C_2=0.4

Legend: B=Binary; C=Continuous; D=Discrete; S=Single; M=Multi Objective.

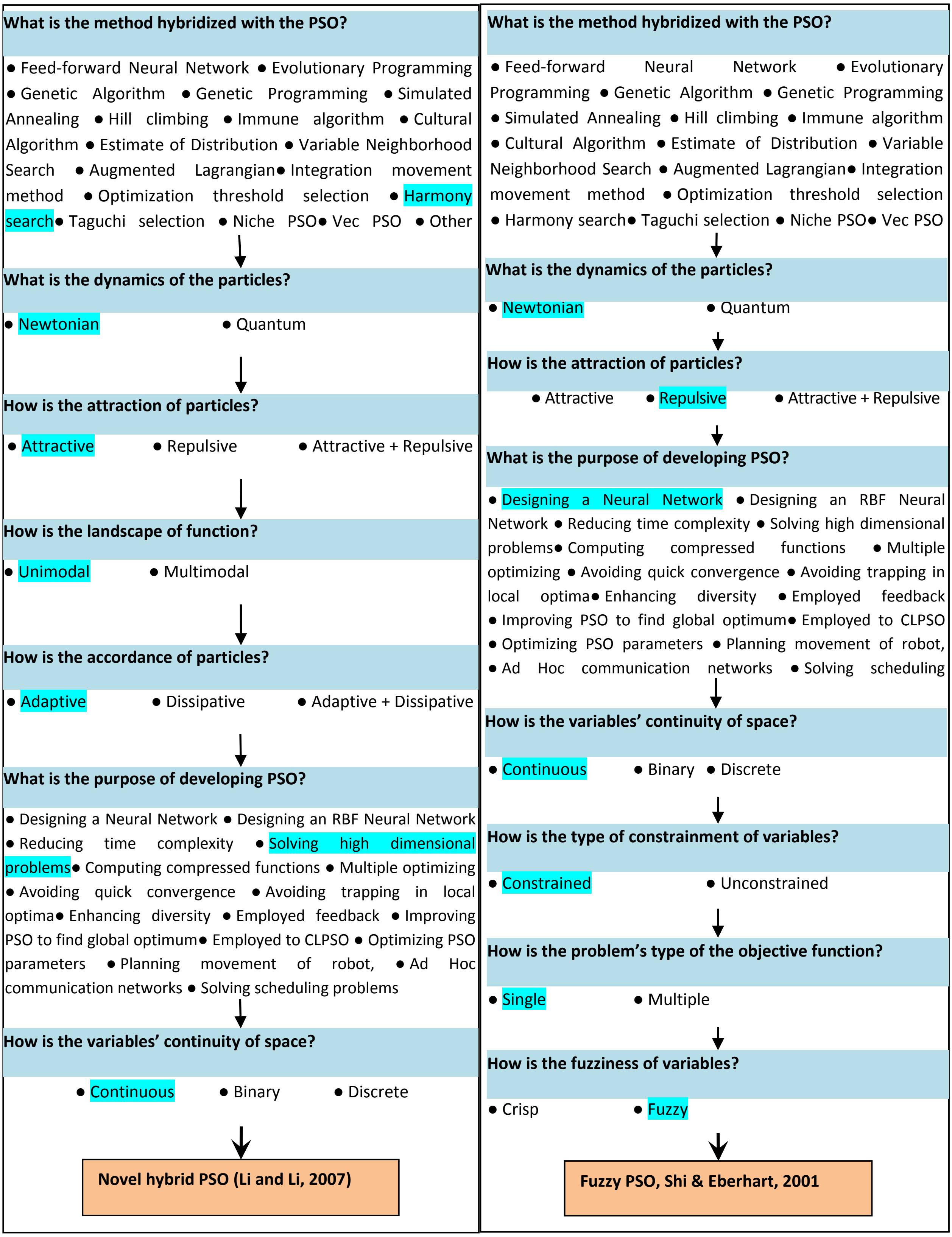

Figure 3. Questions and answers for identifying the best PSO

4. Conclusion

In this paper, using a comprehensive survey and taxonomy of methods based on the PSO algorithm, an Expert System is introduced in order to select the most proper type of PSO algorithm for a specific application. The taxonomy, which consists of 22 classes, as well as features and parameters of introduced PSOs, is utilized as input parameters to the system. Then, by integrating different acquirable information for each algorithm, the knowledge base of the ES was formed. The VP-Expert™ software is used to operationalize the proposed ES. The validity assessment of the ES, done by testing it on 50 recently-published papers from 2009 to 2013, indicated the capability of the system in identifying the best and most efficient type of PSO algorithms with 28% deviation from what were proposed in the original articles.

The proposed ES can be further improved by enriching its knowledge base and incorporating novel algorithms introduced since 2009. Also, by analyzing the reasons of the contrasts between the ES outputs and the studied papers, the number, precision, and efficiency of the rules of the current ES can be increased.

References

Aguirre A. H., Muñoz Zavala A. E. and Diharce E. V., and Botello Rionda S. (2007). COPSO: Constraints Optimization via PSO algorithm. Comunicación Técnic, (CC/CIMAT).

Altinoz O. T., Yilmaz A. E, Weber G. W. (2012). Application of chaos embedded PSO for PID parameter tuning. International J. of Computers, Communications and Control vol. 7, no. 2, pp. 204-217.

Alviar J.-B., Peña J., and Hincapié R., (2007). Subpopulation best rotation: a modification on PSO. Revista Facultad de Ingeniería. Vol. 40, pp.118-122.

Atyabi A. and Phon-Amnuaisuk S. (2007). Particle swarm optimization with area extension (AEPSO). *IEEE/CEC*, pp. 1970-1976.

Behera H. S., Dash P. K. and Biswal B. (2010). Power quality time series data mining using s-transform and fuzzy expert system. *Appl. Soft Comput. J.* Vol. 10, No. 3, pp. 945–955.

Brits R., Engelbrecht A. P., and van den Bergh F. (2002). Solving Systems of unconstrained equations using particle swarm optimization. Proc. of IEEE Conf. on Sys. Man and Cyber. (SMC), pp. 102-107.

Brits R., Engelbrecht A.P., and van den Bergh F., (2005). Niche Particle Swarm Optimization. Technical report, Department of Computer Science, University of Pretoria.

Chandrasekaran S., Ponnambalam S.G., Suresh R.K. and Vijayakumar N. (2006). A Hybrid Discrete Particle Swarm Optimization Algorithm to Solve Flow Shop Scheduling Problems. Proc. IEEE/ ICCIS, pp.1 – 6.

Chen C. Y., Feng H. M., and Ye F. (2007). Hybrid Recursive Particle Swarm Optimization Learning Algorithm in the Design of Radial Basis Function Networks. *J. of Marine Science and Technology*, Vol. 15, No. 1, pp. 31-40.

Cui Z., Zeng J. And Cai X. (2004). A guaranteed convergence dynamic double particle swarm optimizer. *Fifth World Cong. on Intell.*, Control and Automation, Vol. 3, pp. 2184 – 2188.

Felix T. S. Chan1, Kumar, V. and Mishra, N. (2007). A CMPSO algorithm based approach to solve the multi-plant supply chain problem. *Swarm Intell.: Focus on Ant and Particle Swarm Optimization*, pp. 532.

Feng H.-M. (2005). Self-generation fuzzy modeling systems through hierarchical recursive-based particle swarm optimization. *J. of Cyber. and Systems*, Vol. 36, No. 6, pp. 623-639.

Galvez A. and Iglesias A. (2013). A new iterative mutually-coupled hybrid GA-PSO approach for curve fitting in manufacturing. *Applied Soft Computing*, No. 13, Vol. 3, pp. 1491–1504.

Ghamisi P., Couceiro M.S., Martins F.M.L., and Benediktsson J.A. (2013). Multilevel Image Segmentation Based on Fractional-Order Darwinian Particle Swarm Optimization. *J. of Geoscience and Remote Sensing*, Vol. pp, Issue. 99, pp. 1-13, June 2013.

Gheitanchi S., Ali F. H. and Stipidis E. (2008). Trained Particle Swarm Optimization for Ad-hoc Collaborative Computing Networks. *Swarm Intell. Algorithms and Applications Symp*., ASIB, UK.

Giarratano J. C. and Riley G. (1998). Expert Systems: Principles and Programming. PWS Publishing Company.

He S., Wu Q.H., Wen J.Y., Saunders J.R. and Paton R.C. (2004). A particle swarm optimizer with passive congregation. *J. of Biosystems*, Vol.78, No.1-3, pp. 135-147.

Higashitani M., Ishigame A. and Yasuda K. (2008). Pursuit-Escape Particle Swarm Optimization. *Trans. On Electrical and Electronic Eng.*, (IEEJ), Vol.3, No.1, pp.136 – 142.

Ho S.-Y., Lin H.-S., Liauh W.-H., and Ho S.-J. (2008). OPSO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems. *IEEE Trans. on Systems, Man and Cyber.*, Part A, Vol. 38, No., 2, pp. 288-298.

Hu X. and Eberhart R. C. (2002). Multi objective optimization using dynamic neighborhood particle swarm optimization. Proc. of the IEEE/ CEC, pp. 1677-1681.

Hui W. and Feng Q. (2007). Improved particle swarm optimizer with behavior of distance models. *J. of Computer Eng. and Applications*, 43 (30): 30-32.

Jang W.-S., Kang H.-I., Lee B.-H., Kim K.-I., Shin D.-I. and Kim S.-C. (2007). Optimized fuzzy clustering by predator prey particle swarm optimization. In IEEE/CEC, pp. 3232-3238.

Jarbouia B., Cheikha M., Siarryb P. and Rebaic A., (2007). Combinatorial particle swarm optimization (CPSO) for partitioned clustering problem. *J. Applied Mathematics and Computation*, Vol. 192, Issue 2, 15 pp. 337-345.

Ji C., Zhang Y., Gao SH., Yuan P. and Li Zh. (2004). Particle swarm optimization for mobile ad hoc networks clustering. IEEE Int. Conf. on Networking, Sensing and Control, Vol. 1, pp. 372 – 375.

Jie J., Zeng j. and Han C. (2006). Self-Organization Particle Swarm Optimization Based on Information Feedback. Advances in natural comput.: (Part I-II: Second Int. conf., ICNC 2006, Xi'an, China.

Jingbo A. and Hongfei T., (2005). Cultural based Particle Swarm Optimization. Center for Science and Technology Development, Ministry of Education P. R. China.

Keedwell E., Morley M., and Croft D. (2012). Continuous Trait-Based Particle Swarm Optimisation (CTB-PSO). 8th International Conference, Brussels, Belgium, Vol. 7461, pp 342-343 Sept.

Kennedy J. and Eberhart R. C. (1995). Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, Vol. IV, pp. 1942-1948.

Kennedy J. and Eberhart R. C. (2001). Swarm Intelligence. Morgan Kaufmann, San Francisco, CA.

Kennedy J. and Eberhart R.C. (1997). A discrete binary version of the particle swarm algorithm. Int. IEEE Conf. on Systems, Man, and Cyber. Vol.5, pp.4104 – 4108.

Koh B-Il, Fregly B.-J., George A.-D. and Haftka R.-T. (2005). Parallel asynchronous particles swarm for global biomechanical. *Int J Number Methods Eng.*, 67(4): 578–595.

Krink T. Vesterstrom J.S. and Riget J. (2002). Particle swarm optimization with spatial particle extension. Proc. of Cong. on Evolutionary Computation, (CEC’02), Vol. 2, pp. 1474-1479.

Kulkarni R.V. and Venayagamoorthy G.K. (2007). An Estimation of Distribution Improved Particle Swarm Optimization Algorithm. Proc. IEEE/ ISSNIP, pp. 539-544.

Lam H. T., Nikolaevna P. N., and Quan N. T. M. (2007). The Heuristic Particle Swarm Optimization. Proc. of annual Conf. on Genetic and evolutionary computation in Ant colony optimization, swarm Intell. and artificial immune systems,”GECCO’07”, pp.174 – 174.

Lee C. H., Lee Y. C., and Chang F. Y. (2010). A Dynamic Fuzzy Neural System Design via Hybridization of EM and PSO Algorithms. *IAENG International J. of Computer Science*, 37:3.

Lee K. H., Baek S. W. and Kim K. W. (2008). Inverse radiation analysis using repulsive particle swarm optimization algorithm. International Journal of Heat and Mass Transfer; 51(11-12):2772–2783.

Lee, T. Y. (2007). Optimal Spinning Reserve for a Wind-Thermal Power System Using EIPSO. *IEEE/ TPWRS*, Vol. 22, No.4, pp. 1612 – 1621.

Li H. Q. and Li L. (2007). A novel hybrid particle swarm optimization algorithm combined with harmony search for high dimensional optimization problems. International Conference on Intelligent Pervasive Computing, Jeju Island, Korea.

Li H.-Q. and Li L. (2007). A Novel Hybrid Particle Swarm Optimization Algorithm Combined with Harmony Search for High Dimensional Optimization Problems. Proc. IEEE/IPC, pp. 94-97.

Li T., Lai X. and Wu M. (2006). An Improved Two-Swarm Based Particle Swarm Optimization Algorithm. Proc. IEEE/ WCICA, Vol. 1, pp.3129-3133.

Li X. (2004). Adaptively Choosing Neighborhood Bests using Species in a Particle Swarm Optimizer for Multimodal Function Optimization. Proc. of GECCO LNCS 3102, pp.105-116.

Liang J. J., Qin A. K. and Baskar S. (2006). Comprehensive Learning Particle Swarm Optimizer for Global Optimization of multimodal Functions. *IEEE Trans. Evolutionary Computation*, Vol. 10, No. 3.

Liao c.y., Lee W. P. and Chen X. (2007). Dynamic and adjustable particle swarm optimization, Proc. of the 8th Conf. on 8th WSEAS Int. Conf. on Evolutionary Computing, Vol. 8.

Lin C., Liu Y. and Lee C. (2008). An efficient neural fuzzy network based on immune particle swarm optimization for prediction and control applications. *J. of Innovative Computing, Information and Control*, Vol. 4, No.7, pp.1711-1722.

Lu H. and Chen W. (2006). Dynamic-objective particle swarm optimization for constrained optimization problems. *J. of Combinatorial Optimization*, Vol. 12, No., 4, PP. 409-419.

Lu H. and Chen W. (2008). Self-adaptive velocity particle swarm optimization for solving constrained optimization problems. *J. of Global Optimization*, Vol.41, No.3, pp. 427-445.

M. Neethling, and A. Engelbrecht. (2006). Determining RNA Secondary Structure using Set-based Particle Swarm Optimization. In Proc. IEEE Cong. on Evolutionary Computation, Vancouver, pp. 1670-1677.

Madar J., Abonyi J. and Szeifert F. (2005). Interactive particle swarm optimization. Proc. Int. IEEE conf. On Intelligent Systems Design and Applications (IEEE/ ISDA), Vol.8, No.10, pp.314 – 319.

McNabb A. W., Monson C. K. and Seppi K. D. (2007). MRPSO: Map Reduce particle swarm optimization. Proc. of the 9th annual Conf. on Genetic and evolutionary Comput., pp. 177 – 177.

Medeiros D. J., Swenson E. and DeFlitch C. (2008). Improving Patient Flow in a Hospital Emergency Department. Proc. Winter Simulation Conf. Austin, TX, pp.1526- 1531.

Meissner M., Schmuker M. and Schneider G. (2006). Optimized Particle Swarm Optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics, 7:125.

Miranda V. and Fonseca N. EPSO – Best-of-two-worlds Meta-heuristic Applied to Power System problems. In Proc. of the IEEE Congress on Evolutionary Computation, Honolulu, Vol. 2, pp. 1080-1085.

Mo Y., Chen D. and Hu S. (2006). Chaos particle swarm optimization algorithm and its application in biochemical process dynamic optimization, J. Chem. Ind. Eng. (China) 57 (9) 2123–2127.

Monson C. K. and Seppi K. D. (2005). Linear Equality Constraints and Homomorphous Mappings in PSO. *IEEE Congress on Evolutionary Computation (CEC'2005)*, Vol. 1, IEEE Service Center, Edinburgh, Scotland, pp. 73–80.

Moore P.W. and Venayagamoorthy G.K. (2006). Empirical Study of an Unconstrained Modified Particle Swarm Optimization. *IEEE/CEC*, pp. 1477-1482.

Moraglio A., Di Chio C., Togelius J. and Poli R. (2008). Geometric Particle Swarm Optimization. J. of Artificial Evolution and Applications.

Moraglio A., Di Chio C., Togelius J. and Poli, R. (2008). Geometric Particle Swarm Optimization. J. of Artificial Evolution and Applications.

Neethling M. and Engelbrecht A. P. (2006). Determining rna secondary structure using set-based particle swarm optimization. In IEEE Congress on Evolutionary Computation. CEC 2006., pages 1670 – 1677.

Noel M.M. and Jannett T.C. (2004). Simulation of a new hybrid particle swarm optimization algorithm. Proc., Of the IEEE Symp. On System Theory, pp. 150 – 153.

Omkar S. N., Mudigere D., Narayana Naik G., and Gopalakrishnan S. (2008). Vector evaluated particle swarm optimization (VEPSO) for multi-objective design optimization of composite structures. *J. Computers and Structures*, Vol. 86, No.1-2. pp .1-14.

Özcan E. and Yılmaz M. (2006). Particle Swarms for Multimodal Optimization. Lecture Notes In Computer Science; Vol. 4431, Proc. of the 8th int. conf. on Adaptive and Natural Computing Algorithms, Part I. pp. 366 – 375.

Pampara G., Franken N. and Engelbrecht A.P. (2005). Combining particle swarm optimization with angle modulation to solve binary problems. The IEEE Cong. on Evolutionary Comput., Vol. 1, pp. 89-96.

Pant M., Radha T. and Singh V.P. (2007). A New Particle Swarm Optimization with Quadratic Interpolation. Int. IEEE Conf. on Computational Intell. and Multimedia Applications, Vol.1, pp.55-60.

Pant, M., Thangaraj, R. and Abraham, A. (2008). Particle Swarm Optimization Using Adaptive Mutation. IEEE/DEXA'08, pp. 519-523.

Parsopoulos K. E. and Vrahatis M. N. (2004). UPSO: A Unified Particle Swarm Optimization Scheme. Proc. of the Int. Conf. of Computational Methods in Sci. and Eng., Vol. 1, pp. 868-873.

Pasupuleti S. and Battiti R. (2006). The Gregarious Particle Swarm Optimizer (GPSO). GECCO'06.

Poli R., Langdon W. B., and Holland O. (2005). Extending particle swarm optimization via genetic programming. In M. Keijzer *et al.* (Eds.), Lecture notes in computer science: Vol. 3447. Proc. of the 8th European conf. on genetic programming, pp. 291–300, Springer.

Rezaee Jordehi A., Jasni J., Abd Wahab N., Kadir M.Z., Javadi M.S. (2015). Enhanced leader PSO (ELPSO): a new algorithm for allocating distributed TCSC's in power systems. *Int. J. Electr. Power Energy Syst*, Vol. 64, pp. 771-784.

Riget J. and Vesterstroem J. S. (2002). A diversity-guided particle swarms optimizer - the ARPSO. Technical Report No. 2002-02. Dept. of Computer Science, University of Aarhus, EVALife.

Roy R. and Ghoshal S.P. (2006). Evolutionary computation based optimization in fuzzy automatic generation control. IEEE/ POWERI, pp.7.

Sadri J. and Suen C.Y. (2006). A Genetic Binary Particle Swarm Optimization Model. Proc. IEEE/CEC, pp.656 – 663.

Schoeman I. L. and Engelbrecht A. P. (2004). Using Vector Operations to Identify Niches for Particle Swarm Optimization. In Proc. of the IEEE Conf. on Cyber. and Intelligent Sys. PP. 361-366.

Schoeman I. L. and Engelbrecht A. P. (2004). Using vector operations to identify niches for particle swarm optimization. in Proceedings of IEEE Conference on Cybernetics and Intelligent Systems (CCIS), Vol. 1, pp. 361–366, Singapore.

Secrest B.R. and Lamont G.B. (2003). Visualizing particle swarm optimization – Gaussian particle swarm optimization. Proc. Of Swarm Intell. Symp. (IEEE/SIS), pp. 198- 204.

Sedighizadeh D. and Masehian E. (2009a). A New Taxonomy for Particle Swarm Optimization (PSO). Proc. 10th Int. Conf. on Automation Technology, pp.317-322.

Sedighizadeh D. and Masehian E. (2009b). Particle Swarm Optimization Methods, Taxonomy and Applications. *International Journal of Computer Theory and Engineering*, Vol.1, pp.486-502.

Sedlaczek K. and Eberhard P. (2006). Using Augmented Lagrangian Particle Swarm Optimization for Constrained Problems in Engineering. *J. of Structural and Multidisciplinary Optimization*, Vol. 32, No. 4, pp. 277-286.

Shen X., Wei K., Wu D., Tong Y. and Li Y. (2007). A Dynamic Adaptive Dissipative Particle Swarm Optimization with Mutation Operation. Proc. IEEE/ ICCA, pp. 586-589.

Shi Y. and Eberhart R. (2001). Fuzzy Adaptive Particle Swarm Optimization. Proc. IEEE / Cong. on Evolutionary Computation. Seoul, vol. 1, pp. 101-106.

Shi Y., and Krohling R. A. (2002). Co-evolutionary particle swarm optimization to solve min-max problems. in Proc. Cong. on Evolutionary Comput., Vol. 2, pp. 1682-1687.

Subrarnanyam V. Srinivasan D. and Oniganti R. (2007). Dual layered PSO Algorithm for evolving an Artificial Neural Network controller. IEEE/CEC, pp. 2350-2357.

Sun J., Feng B. and Xu W. (2004). Particle swarm optimization with particles having quantum behavior. Proc. IEEE/CEC, Vol. 1, pp. 325 – 331.

Tao Q., Chang H. Y., Yi Y., Gu C. Q., and Yu Y. Qo S. (2009). constrained grid workflow scheduling optimization based on a novel PSO algorithm. in 8th International Conference on Grid and Cooperative Computing, Guangzhou, China, pp. 153-9.

Voss M.S. (2005). Principal component particle swarm optimization (PCPSO). Proc., Of the IEEE Symp. On swarm Intell., pp. 401 – 404.

Wang H., and Qian F. (2007). An improved particle swarm optimizer with shuffled sub-swarms and its application in soft-sensor of gasoline endpoint. Proc. Advances in Intelligent Systems Research.

Wang H., and Qian F. (2007). An improved particle swarm optimizer with shuffled sub-swarms and its application in soft-sensor of gasoline endpoint. Proc. Advances in Intelligent Systems Research.

Wang X. H. and Li J.-J. (2004). Hybrid particle swarm optimization with simulated annealing. Proc., Of the IEEE Int. Conf. on Machine Learning and Cyber., Vol. 4, pp. 2402 – 2405.

Waterman D. A. (1986). *A guide to expert systems*, illustrated, reprint ed. California, USA: Addison-Wesley.

Wei C., He Z., Zhang Y. and Pei W. (2002). Swarm directions embedded in fast evolutionary programming . In Proc. of the IEEE/CEC, pp. 1278–1283.

Wei K., Zhang T., Shen X. and Liu J. (2007). An Improved Threshold Selection Algorithm Based on Particle Swarm Optimization for Image Segmentation. Proc. IEEE/ICNC.

Xiao-ping X., QIAN F. C. and Feng W. (2008). Research on new method of system identification based on velocity mutation Particle Swarm Optimization. (In Chinese), 44 (1) pp.31-34.

Xie X.-F., Zhang W.-J. and Yang Z.-L. (2002a). Adaptive Particle Swarm Optimization on Individual Level. Int. Conf. On Signal Processing (ICSP), pp: 1215-1218.

Xie X.-F., Zhang W.-J., and Yang Z.-L. (2002b). A Dissipative Particle Swarm Optimization. Cong. on Evolutionary Comput. (CEC), pp.1456-1461.

Xu F. and Chen W. (2006). Stochastic Portfolio Selection Based on Velocity Limited Particle Swarm Optimization. Proc. IEEE/ WCICA, Vol. 1, pp.3599-3603.

Yang C. and Simon D. (2005). A New Particle Swarm Optimization Technique. Proc. Of the Int. Conf. on systems Eng., (IEEE/ISEng'05).

Yang S., Wang M. and Jiao L. (2004). A Quantum Particle Swarm Optimization. In Proceedings of Congress on Evolutionary Computation CEC2004, vol.1. 320-324.

Yang W.-P. (2007). Vertical Particle Swarm Optimization Algorithm and its Application in Soft-Sensor Modeling. Int. Conf. on Machine Learning and Cyber. (IEEE/ ICMLC), Vol.4, pp.1985-1988.

Yao X. (2008). Cooperatively Coevolving Particle Swarms for Large Scale Optimization. Conf. of EPSRC, Artificial Intell. Technologies New and Emerging Computer Paradigms.

Yin P. Y. (2006). Genetic particle swarm optimization for polygonal approximation of digital curves. *J. of Pattern Recognition and Image Analysis.* Vol. 16, No. 2, pp. 223-233.

Yuan L. and Zhao Z.-D. (2007). A Modified Binary Particle Swarm Optimization Algorithm for Permutation Flow Shop Problem. IEEE/ICMLC, Vol.2, pp.902-907.

Yuan Z., Jin R., Geng J., Fan Y., Lao J., Li J., Rui X., Fang Z. and Sun J. (2005). A perturbation particle swarm optimization for the synthesis of the radiation pattern of antenna array. Proc. IEEE Conf. on Asia-Pacific, Vol.3, No, 4-7.

Zeng J., Hu J. and Jie J. (2006). Adaptive Particle Swarm Optimization Guided by Acceleration Information. Proc. IEEE/ ICCIAS, Vol.1, pp.351-355.

Zhang Q. and Mahfouf M. (2006). A New Structure for Particle Swarm Optimization (nPSO) Applicable to Single Objective and Multi objective Problems. Int. IEEE Conf. on Intelligent Systems, pp.176 – 181.

Zhang X., Hu W., Maybank S., Li X. and Zhu M. (2008). Sequential particle swarm optimization for visual tracking. IEEE/CVPR, pp. 1-8.

Zhang Y.-N., Hu Q.-N. and Teng H.-F. (2008). Active target particle swarm optimization: Research Articles. *J. of Concurrency and Computation: Practice & Experience*, Vol.20, No.1, pp.29 – 40.

Zhao B. (2006). An Improved Particle Swarm Optimization Algorithm for Global Numerical Optimization. Int. Conf. on Comput. Science N6, Reading, (Royaume-uni), Vol. 3994, pp. 657-664.

Zhiming L., Cheng W. and Jian L. (2008). Solving constrained optimization via a modified genetic particle swarm optimization. Proc. of Int. Conf. On Forensic applications and techniques in telecommunications, information, and multimedia and workshop, No. 49.