Swarm Intelligence Techniques and Their Adaptive Nature with Applications

Anupam Biswas and Bhaskar Biswas

Abstract Swarm based techniques have huge application domain covering multiple disciplines, which include power system, fuzzy system, forecasting, bio-medicine, sociological analysis, image processing, sound processing, signal processing, data analysis, process modeling, process controlling etc. In last two decades numerous techniques and their variations have been developed. Despite many variations are being carried out, main skeleton of these techniques remain same. With diverse application domains, most of these techniques have been modified to fit into a particular application. These changes undergo mostly in perspective of encoding scheme, parameter tuning and search strategy. Sources of real world problems are different, but their nature sometimes found similar to other problems. Hence, swarm based techniques utilized for one of these problems can be applied to others as well. As sources of these problems are different, applicability of such techniques are very much dependent on the problem. Same encoding scheme may not be suitable for the other similar kind of problems, which has led to development of problem specific encoding schemes. Sometimes found that, even though encoding scheme is compatible to a problem, parameters used in the technique does not utilized in favor of the problem. So, parameter tuning approaches are incorporated into the swarm based techniques. Similarly, search strategy utilized in swarm based techniques are also vary with the application domain. In this chapter we will study those problem specific adaptive nature of swarm based techniques. Essence of this study is to find pros and cons of such adaptation. Our study also aims to draw some aspects of such problem specific variations through which it can be predicted that what kind of adaptation is more convenient for any real world problem.

A. Biswas (✉) · B. Biswas
Department of Computer Science and Engineering, Indian Institute of Technology (BHU), Varanasi 221005, India
e-mail: anupam.rs.cse13@iitbhu.ac.in; abanumail@gmail.com

B. Biswas
e-mail: bhaskar.cse@iitbhu.ac.in

Q. Zhu and A.T. Azar (eds.), *Complex System Modelling and Control Through Intelligent Soft Computations*, Studies in Fuzziness and Soft Computing 319,
DOI 10.1007/978-3-319-12883-2_9

1 Introduction

Advancement of technology has led real world problems to become complex and more challenging. To acquire requisite quality of such advanced technology, associated problems needed to solve intelligently and efficiently. In these days intelligent techniques become very popular for solving such technology oriented problems. There are variety of ways to solve a single problem, so it is very crucial to decide exactly which cases an intelligent technique needs to be adopted. There are few aspects of such decision. Firstly, the problem in hand has to be feasible for an intelligent technique. There has to be a plug point in the problem where such techniques are to be plugged in. Secondly, even if problem is found suitable for intelligent techniques, the best possible technique has to choose from the archive of numerous techniques available. Lastly, which of the multiple versions of selected technique will be most suited for the problem has to be decided.

Before considering any intelligent technique, feasibility analysis of the problem as well as available techniques have to be carried out. Once suitable technique is found, arises another key issue in implementation of the technique with respect to the problem. As far as swarm intelligence techniques are concerned, mostly were developed for solving optimization problems. Again when we say optimization problem, it covers huge domain. There are different types of optimization problems and have special characteristics of each. Very basic notion of optimization is to find best possible solution from a set of solutions (referred as solution space) to any problem. Corresponding problem with solution space can be summarized with some functions, generally referred as objective function or fitness function. Some problems require constraints along with the functions to define solution space, that case problem is referred as constrained optimization problem. If problem is defined with linear objective functions and constraints, problem is called linear optimization problem, otherwise it is termed as nonlinear. Hence, optimization problems can be categorized as linear or nonlinear on the basis of linearity in the problem definition. Objective functions can be continuous or discrete, accordingly problems are referred as continuous and combinatorial optimization problem respectively. Most of the real world problems experience several constraints, sometime those constraints are defined with nonlinear functions. Often such problems require multiple objective functions to optimize with necessary constraints, referred as multi-objective optimization problem. Solution space may have several preferred solutions, each of them represents best solution and cannot be dominated by other. In this case we have best solution set instead of one best solution, such problems are referred as multi-modal optimization problems. Determination of possibles best solution set is very important to engineering problems, but due to constraints present in the problem, best solutions may not always be realized. Both single and multi-objective problem experiences such hurdles along with diverse constraints and linearity. Todays technology oriented problems become more complex with these issues associated with the problem.

Each kind of optimization problem can be solved very efficiently with some specific techniques. These techniques include mathematical approaches such as linear programming (Matoušek and Gäartner 2007; Todd 2002; Wiki 2014), non-linear programming (Borwein and Lewis 2010; Ruszczynski 2006) and iterative techniques (Wiki 2014) as well as numerous heuristic approaches such as Evolutionary Algorithms (Rechenberg 1994; Schwefel 1994; Yan et al. 2005), Genetic Algorithms (Deb et al. 2002; Holland 1975), Swarm Intelligence (Dorigo 1992; Karaboga 2005; Kennedy and Eberhart 1995; Rashedi et al. 2009; Shah-Hosseini 2008, 2009) and other nature inspired methods. In these days heuristic approaches have gained popularity specially Genetic Algorithms and Swarm Intelligence techniques. Applications of almost all fields utilize swarm intelligence techniques in their specific problems. As mentioned above, swarm intelligent techniques are utilized mainly for solving optimization problem. These techniques are inspired from chemical, biological and physical phenomenon of nature. Extensively used approaches in various applications include Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995), Ant Colony Optimization (ACO) (Dorigo 1992), Artificial Bee Colony (ABC) (Karaboga 2005), Gravitational Search Algorithm (GSA) (Rashedi et al. 2009) and Intelligent Water Drop (IWD) (Shah-Hosseini 2008, 2009). In this chapter we will try to cover these popular swarm based techniques in perspective of their applicability to numerous problems. These techniques have undergone several changes. We will study those changes with respect to their applications and try to draw key issues behind such changes. Variety of applications of different domains as well as frequent variation within the technique create chaos. Induced several confusions regarding selection of suitable technique for the application. We will try to address those issues and summarize them in a generalized manner for all the applications. We will also try to generalize issues regarding variations of techniques and their applicability throughout the chapter.

Rest of the chapter is organized as follows: Section 2 surveys various works which presents applications of swarm based techniques and extrapolates trade off between applications and swarm based techniques. Section 3 provides brief introduction to popular swarm based approaches and generalize these techniques into common framework. Section 4 addresses issues related objectives behind incorporation of swarm based approaches, probable plug points in any application and suitable problem types. Considering these constraints a generalized framework is presented for any application and such techniques. Section 5 presents various encoding schemes of applications to fit swarm based techniques and encoding related changes in techniques. Section 6 explains strategic changes of swarm based techniques. Section 7 illustrates parameter tuning related issues of swarm based techniques. Section 8 discusses various application related problems, advantages and difficulties. Finally, concluded in Sect. 9.

2 Related Work

Several application centric studies on Swarm Intelligent (SI) techniques have been done. Main focus of this brief survey is to cover the kind of works related to the application have been done rather than covering those works and draw application related issues. Numerous studies have been done related to application and SI techniques. Most of those studies were specific to a particular approach (Al Rashidi and El-Hawary 2009; Borwein and Lewis 2010; Chandrasekhar and Naga 2011; Janacik et al. 2013; Kameyama 2009; Kothari et al. 2012; Kulkarni and Venayagamoorthy 2011). Some studies are even more specific to single problem of one particular application domain (Al Rashidi and El-Hawary 2009; Chandrasekhar and Naga 2011; Kulkarni and Venayagamoorthy 2011). There are several applications in single domain which have utilized SI techniques (Al Rashidi and El-Hawary 2009; Kulkarni and Venayagamoorthy 2011).

Several surveys on various applications of SI techniques covering multiple domain have been done (Chandra Mohan and Baskaran 2012; Monteiro et al. 2012). Such studies are often done by considering a single variant of any SI technique such as multi-objective (Reyes-Sierra and Coello 2006), parallel versions (Chu et al. 2003; Schutte et al. 2004; Vanneschi et al. 2012) and so on to different application domains. With these studies one can have an intuitive idea about problems of various domains which could absorb such approach. Problem specific study, associated issues of the problem and study of solution to that problem which covers numerous techniques including SI techniques have often done (Al Rashidi and El-Hawary 2009; Kulkarni and Venayagamoorthy 2011). Such study gives an overall idea of various techniques to solve that problem, but internal matters of those techniques remain unclear, which is the source of confusion regarding which one of the available techniques will be better for the problem.

Generalized studies that covers all application domains where intelligence techniques are utilized, have not been done yet. Studies done earlier also not based on applicability. No clear idea has been come out from previous studies about the way which would be better for incorporating any intelligent techniques into an application. Numerous studies about applications and swarm techniques (Al Rashidi and El-Hawary 2009; Chandra Mohan and Baskaran 2012; Chandrasekhar and Naga 2011; Eslami et al. 2012; Kameyama 2009) have been done. But, study related to usability of any intelligent technique in an application, like what points have to be taken care of, which changes have to be done in application in order to fit the technique in hand has not been done yet. At present, a single approach have several variants and each variant has specific characteristics (Chandra Mohan and Baskaran 2012; Kameyama 2009; Monteiro et al. 2012). Suitable variants of any SI technique for any particular application has not been done yet. Problems faced (Yang et al. 2007) during implementation of SI techniques into an application and solutions to those problems are given. Whether those problems can be arise in other application or if arises whether same solution can be suitable for that application, such issues have not been surveyed yet. Off course, different variants of SI techniques proposed, addition of new parameters and tuning of various parameters have often been surveyed (Kameyama 2009), sometime such study extended to various applications.

Intuition behind introduction of a new technique to solve any problem is to apply in real life problems and solve associated problems more efficiently. Hence, generalized study would be more realistic and helpful to the society of diverse application domain rather than any particular application specific study. SI techniques in particular covers huge application domain. In this case, generalized study with comparative analogy to the applications of multiple domain will be more helpful. Implementer of any domain can have an intuitive idea about applicability of such techniques into the specific applications of any domain.

3 Swarm Intelligences

Swarm intelligent (SI) techniques are heuristic stochastic search processes. SI approaches can be generalized as follows: all approaches are initiated with a set of solutions called population, then in successive steps each candidate of the set learns collectively from other candidates and adapts itself in accordance to the solution space. Strategy incorporated and learning mechanism of these techniques mostly mimic the natural facts and phenomena. Such nature inspired mathematical models can be plugged into one framework. In this section we will brief popular SI techniques and try to wrap up in one generalized framework.

3.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) originally introduced by Kennedy and Eberhart in 1995 (Kennedy and Eberhart 1995). Basic intuition behind PSO was simulation of cooperative learning mechanism of bird's flocking. Flying birds in flock show learning through individual's experience and follow other. One of them leading the flock and other follows that leader. Once leader changes, immediately all other individual including previous leader begin following it. This process continues until reach their destination. Kennedy and Eberhart formulated this process into a mathematical model with two very simple equations. One of those equations was analogous to the position and other one was analogous to the velocity of bird or particle. Experience of individual particle was conserved as personal best i.e. any particle experienced best position so far. Experience of flock or swarm or population was conserved as global best i.e. the best position experienced by the swarm so far. These experiences were used to learn and control velocity of particle. Finally, particles moved to new position with learned velocity. So each particle in solution space are associated with position and velocity. Velocity and position equation proposed by Kennedy and Eberhart in original PSO are shown below:

$$V_i(t+1) = V_i(t) + C_1R_1[X_{pb}(t) - X_i(t)] + C_2R_2[X_{gb}(t) - X_i(t)] \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{2}$$

Here, t denotes discrete time steps, $X_i(t)$ denotes the position vector of particle i and $V_i(t)$ denotes the velocity vector of a particle in the solution space at time step t, $X_{pb}(t)$ be the position vector of particle i's personal best position so far and $X_{gb}(t)$ be the position vector of global best particle so far. R_1 and R_2 are the vectors of uniform random values in range (0, 1). Parameters C_1 and C_2 are the cognitive acceleration coefficient and social acceleration coefficient respectively. $X_i(t+1)$ and $V_i(t+1)$ denote new position and velocity vector at time step $t+1$ respectively.

This strategy can be easily fitted into the generalized SI process as shown in Fig. 1. Only requirement is to represent particle set as vectors of position and velocity. Once we have initial velocity and position of each individual particle, we can update iteratively with strategic Eqs. (1) and (2) until the attain desired approximated position. Actual strategy behind the PSO lies in these two equations. This basic version of PSO has been modified significantly over the years to improve performance, those have been studied in coming sections in perspective of applications.

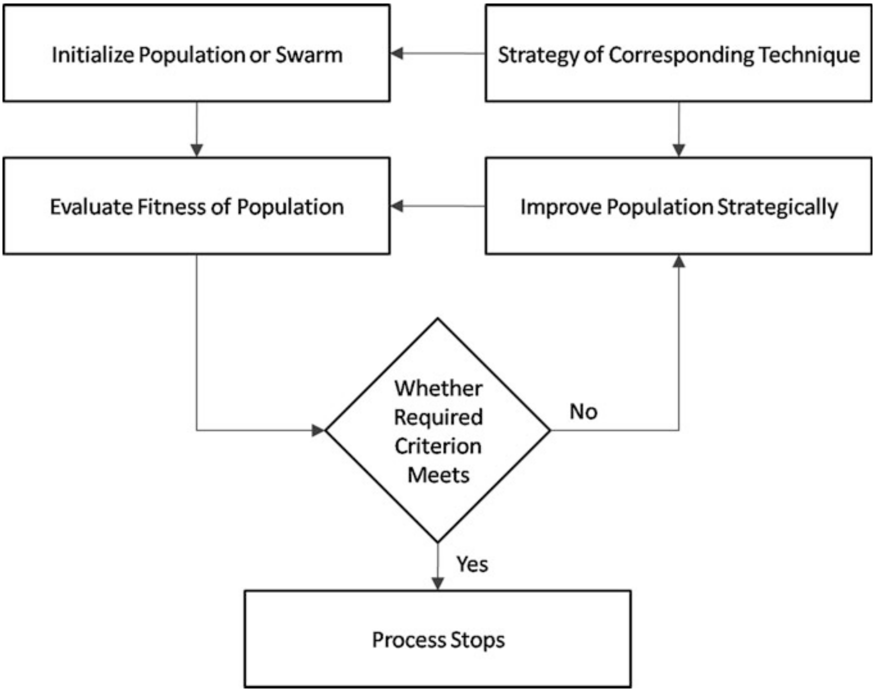
Initialize Population or Swarm
Strategy of Corresponding Technique
Evaluate Fitness of Population
Improve Population Strategically
Whether Required Criterion Meets
No
Yes
Process Stops

Fig. 1 Generalized flow diagram of swarm intelligence technique. Strategy of any SI technique can be inserted into the generalized process once population is initialized in accordance of the strategy

3.2 Ant Colony Optimization

Ant colony optimization (ACO) is inspired by foraging behavior of ants. Ants deposit pheromone to mark the favorable path and that path is followed by other member of the colony to collect foods. Over the time pheromone evaporates and hence, relative attraction to that specific path decreases. As much time an ant takes to travel to and fro from source to destination, proportionate amount of pheromone evaporates by that time. Summarily smaller path takes less time, which implies less evaporation, so density of pheromone becomes higher. Therefore, if one ant finds a good (i.e., short) path from the source to destination, other ants are more likely to follow that path. Artificial ant system has to do two main tasks: updating pheromone and selection of path with maximum pheromone densities to reach destination. To acquire ACO method, solution space is represented with graph. The pheromone τ_{ij}, associated with the edge joining nodes i and j, is updated as follows:

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \sum_{k=1}^{m} \varDelta\tau_{ij} \tag{3}$$

where, ρ is the evaporation rate, m is the number of ants, and $\varDelta\tau_{ij}$ is the quantity of pheromone deposited on edge (i, j) by ant k. $\varDelta\tau_{ij}$ is measured with a ratio Q/L_k if ant k used edge (i, j) in its tour, if not considered 0 value, where Q is a constant, and L_k is the cost of the tour constructed by ant k.

ACO adopts probabilistic approach to select path. This probability depends on priori desirability represented by attractiveness η_{ij} of the move and posteriori τ_{ij} of the move, indicating how desirable it has been in the past to make that particular move. When an ant k is at node i then the probability of going to node j is given by the equation below:

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha} \cdot \tau_{ij}^{\beta}}{\sum_{j \in allowed_j} \tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}} \tag{4}$$

Here, τ_{ij} is the amount of pheromone deposited for moving from node i to j, α is a parameter to control the influence of τ_{ij} which takes greater than equal to 0 value, β is another parameter to control the influence of η_{ij} which takes value greater than equal to 1. η_{ij} is the attractiveness of edge (i, j) which is considered as inverse of cost of the edge (i, j).

In each iteration ants add new transition to construct final solution and update pheromone level in the path. Once solutions are represented with graph and initialize ants accordingly, ACO can be fitted in general framework.

3.3 Artificial Bee Colony

Karaboga introduced Artificial Bee Colony (ABC) (Karaboga 2005) algorithm by modeling foraging behavior of bee. Artificial bee colony have groups of three kind of bees employed bee, onlooker bee and scout bee. Employed bee and onlooker bee are responsible for determining amount of food present in the food source. One employed bee is assigned for every source of food. Employed bee become scout if food source assigned to it is exhausted.

Basic strategy of ABC consists three steps: (1) move the employed and onlooker bees onto the food sources to determine food amounts, (2) determine scout bees from employed bees, and (3) direct scout bees to search new food source. Position of food source found by scout bees represents a possible solution to the problem that has to be optimized. Onlooker bees are placed probabilistically to each food source. As the quality of food source increases, the probability with which the food source chosen by onlooker bees is also increases. Natural scout bees have no guidance for exploring new food sources, but artificial bee can explore solution space as a task to find new food sources strategically. This has led to fast discovery of feasible solution that are represented by food sources. If within the predetermined number of trials a solution representing a food source does not improved, then that food source is abandoned by associated employed bee and converted to a scout.

This strategy can be incorporated into generalized SI framework by interpreting group of solutions as artificial bees. Divide population into three groups as mentioned above and perform task accordingly.

3.4 Gravitational Search Algorithm

Main inspiration of Gravitational search algorithm (GSA) was taken from Newton's law of universal gravitation and laws of motion. Every point in solution space is considered as a searching agent having mass. Searching agents of GSA are considered as objects and their performance is measured by their masses. Due to gravitational force each object attracts other objects. Object with lower masses will be attracted towards the object with heavier mass. Heavy masses represent comparatively better solution and moves slowly. Another fact of physics is also considered, according to which objects in space do not feel uniform gravitational force due to varying gravitational constant. Actual value of gravitational constant G depends on the age of the universe. So gravitational constant $G(t)$ at time t can be computed as follows:

$$G(t) = G(t_0) \times \left(\frac{t_0}{t}\right)^{\beta}, \quad \beta < 1 \tag{5}$$

Here, $G(t_0)$ is the value of the gravitational constant at the first cosmic quantum-interval of time t_0. This varying $G(t)$ is used to compute total force F_i^d exerted on agent i from direction d in the space. Due to this force agents are accelerated towards each other. Acceleration of any agent i having mass M_i at time t in direction d is computed with the equation presented below:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i(t)} \tag{6}$$

The position velocity of each agent is updated with laws of motion as follows:

$$v_i^d(t+1) = R \times v_i^d(t) + a_i^d(t) \tag{7}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{8}$$

Here, $x_i^d(t+1)$ and $v_i^d(t+1)$ are the position and velocity of agent i in direction d at time $t+1$. R is an uniformly distributed random variable within range $(0, 1]$. This strategy can be fitted into generalized framework of SI by considering population as set of searching agents.

3.5 Intelligent Water Drop

Intelligent Water Drops (IWD) algorithm was introduced by Shah-Hosseini (2008). IWD simulates flow of river water. It seems that natural river often follows favorable paths among lots of possible different paths on the ways from the source to destination. Paths through which water flows in rivers may have several twists and turns, but always chose best possible path. Intelligence behind those twists and turns are the key inspiration of the IWD algorithm. Those approximate best paths are resulted by the actions and reactions, which occur among the water drops and the water drops with the soil. Considering these aspects IWDs are created with two properties soil and velocity. Solution space is represented with a graph. IWDs are distributed over the graph and starts moving through edges. An IWD flows from a source to a destination. Initially IWDs have velocity but zero soil. During movement from one node to another, removes soil from the path and gain speed. Increment in velocity of an IWD is non-linearly and inversely proportional to the soil present in the path. Therefore, the IWD become faster in a path with less soil.

Three things happening during movement of IWDs in graphically represented solution space. Firstly, IWDs gain velocity and gather soil from the path they moved through. Secondly, proportionate amount of soil is removed from the path of the graph through which they move. Time factor is used during removal and addition of soil. Less the time to pass IWD through a path can remove larger amount of soil. The time is proportional to the velocity of the IWD and inversely proportional to the distance between the two nodes. Lastly, IWD needs to chose path to next node from the multiple paths. Mechanism to chose path is that the IWD prefers path with less soil. Hence, paths with less soil have higher chance to get chosen by an IWD.

The strategy looks very similar to ACO. Pheromones are deposited through the path an ant moves and with the time pheromone decreases as it evaporates. In case of IWD, soil is removed from the path when an IWD moves through it. Only difference is changes to the path is constant in case of ACO, whereas in case of IWD these change are dependent on velocity and soil gained by an IWD. This strategy can be fitted to the generalized SI framework once we have initial soil and velocity. In coming sections application and variation related issues with SI techniques explained above are addressed, and also extended to other remaining SI techniques in a generalized form.

4 Applicability of SI Techniques

An application is a composition of several sub-applications or modules. Each module may be an explicit and working application. Considering the feature selection problem of pattern recognition, it has several applications such as Medical disease diagnosis (Selvaraj and Janakiraman 2013), salient object detection (Singh et al. 2014) etc. Salient object detection can be used in surveillance systems (Graefe and Efenberger 1996), image retrieval (Amit 2002; Gonzalez and Woods 2002), advertising a design (Itti 2000) etc. All these abstract level applications in background require feature selection. Often intelligent techniques are used for feature selection (Selvaraj and Janakiraman 2013; Singh et al. 2014). Similarly, several real world applications at different level of abstraction require intelligent techniques in background to solve associated problems. Sometime, intelligent techniques are hybridized with classical methods (Ranaee et al. 2010). Hence, even if an application in hand not directly incorporate intelligent techniques in it, alternatively can be hybridized with classical approaches to improve efficiency.

Application systems may have several control parameters that decide overall behavior of the system. Behavior of any system can be depicted as function of control parameters of that system. To optimize such functions optimization techniques are utilized. Intuitively there may have two kinds of objectives behind using any optimization technique. First one is about finding optimal values of the function for the system. Second one is to find optimal settings of control parameters for the system. Both objectives are interrelated, as optimal value of the function implies optimal settings of control parameters. Normally it looks both are inseparable, but depending on how the application will going to be benefited with the optimization techniques, one can get clear indication about the notion of objective behind incorporation of such techniques. Both kind of objectives can be understood with very general application such as Traveling Salesman Problem (TSP) (Dorigo et al. 2006; Shah-Hosseini 2009). Normally, main aim of TSP is to find shortest route to visit all cities. It is clear that salesman would be benefited with shortest route obtained through any optimization technique. If a vendor gives salary to the salesman depending on the distance, then the route obtained is irreverent to the vendor. In this case simply the minimum distance matters, no matter what route salesman have followed. TSP fitness function is considered as sum of distances among all cities, here distance between any two cities acts as control parameter to the system. To utilize optimization technique, both vendor and salesman will consider same function as both needed minimum distance to cover all cities. But at the end, salesman uses the route obtained with minimum distance i.e. the control parameters and vendor uses minimum distance i.e. the function value.

Concept of background objective for incorporating SI techniques can be understood with other examples. Equalizer plays very important role in digital transmission system. Das et al. (2014) used Artificial Neural Network (ANN) for channel and co-channel equalizer. ANN has three main components i.e. weight, network topology and transfer function. ANN with specific topology and transfer function, performs well when weights are suitable. ANNs are trained to obtain suitable weight and this application used PSO to do so. This application is of second kind as it grasps trained ANN to equalizer not the fitness value. Similarly, application to feature selection mostly used features corresponding to the best accuracy (Ranaee et al. 2010; Selvaraj and Janakiraman 2013; Singh et al. 2014).

Whatever may be the intuition behind utilization of SI techniques finally the technique has to be plugged in somewhere in the application. Implementer has to detect such plug points in the application. Indication to those plug point can be obtained by analyzing problems of application. Those plug points may be unclear or dissimilar to the considered SI technique, in those cases encoding mechanism (described in Sect. 5) is used to unite. A generalized model to incorporate of SI technique in any application is presented in Fig. 2. SI techniques are very suitable for NP-hard problems. Those complex problems suffer exponential worst case

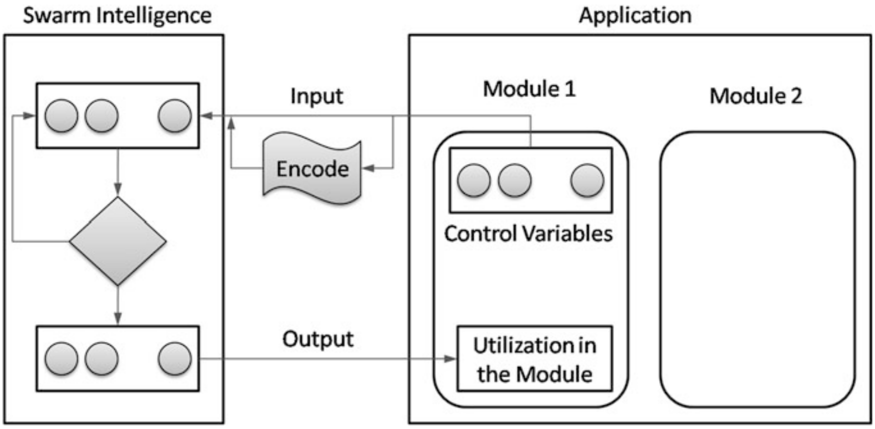

Fig. 2 View of an application with swarm intelligence technique

complexity. Hence, these problems require approximated solution which is relatively good. SI techniques resulted high quality solution to those problem (Dorigo et al. 2006; Shah-Hosseini 2009). Real life application often experiences such problem. In power system applications, problems such as optimal power flow is a NP-hard problem (Alzalg et al. 2011). As explained above, an application have several modules. Those modules may be of different kind of problems. Some of those problems may be of NP-hard category. Summarily, one can get an indication to adopt SI techniques to solve those problems. For example clustering is a NP-hard problem (Kulkarni and Venayagamoorthy 2011). Many application of various domains such as social network analysis (Honghao et al. 2013; Kumar and Jayaraman 2013), image processing (Hancer et al. 2012), wireless sensor network (Kulkarni and Venayagamoorthy 2011) used clustering for their respective problems. This is notable that all these applications adopt SI techniques related to clustering. Hence, detection of problem complexity can be used as good indicator to decide whether associated application needed to grasp SI technique to solve the problem. Objective behind such grasping may vary with problem to problem and can be decided on the basis of problem. As mentioned above, clear objective and problem complexity of associated problem can be handy to decide on applicability of SI techniques.

5 Encoding Schemes

SI techniques act as black box to the application and to implementers act as white box. These techniques take control parameters and fitness function of an application as input and gives optimal solutions as output. Irrespective of the application, optimization techniques will result solution to any input. Those results may be suitable for the application or may not be. It is the implementer who can judge and manipulate the inputs with respect to the considered technique. Results are very much dependent on representative inputs of the application to the SI technique. Hence, representation of control parameters of an application in terms of input to any SI technique is very important. Decision on such issues are very sensitive to the application.

SI techniques use inputs to evaluate fitness or objective function that effects control parameters of application and this function has to be optimized. Generally SI techniques use Boolean or Decimal values to evaluate fitness function. If application's control parameters are not in the same form as the considered technique, then control parameters have to be encoded so that it become required kind of values that are to be used by the technique. Otherwise, application control parameters can be used directly as input to the SI technique. The generalized mechanism of encoding scheme is presented in Fig. 3 with suitable examples. When application control parameter values are Boolean, actually in this case value of any control parameter indicates its selection if value is 1 and deselection if value is 0. During encoding into Decimal, control parameter values are taken together as Boolean string. All control parameter values can be considered at once to represent a single Boolean string or grouped into

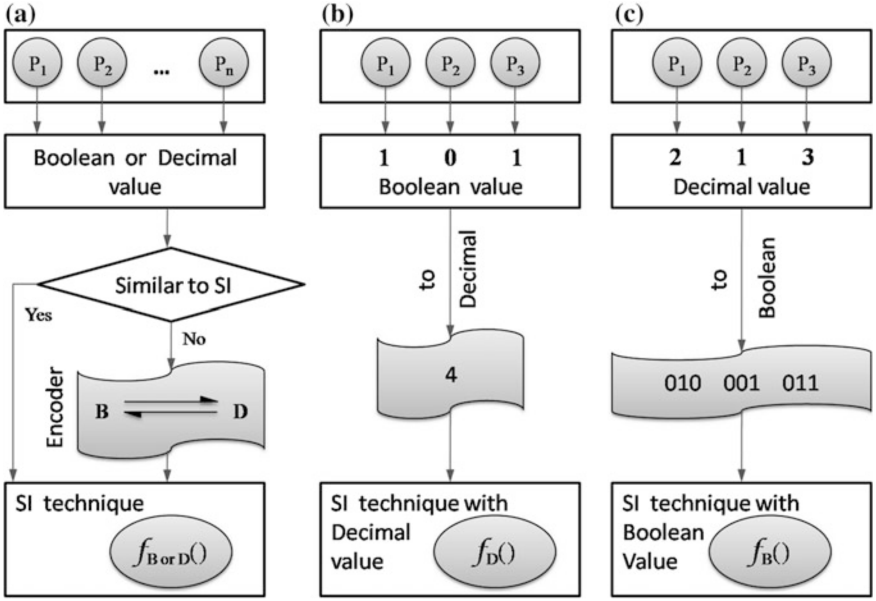

Fig. 3 Encoding mechanism of control parameters of any module of an application corresponding to the inputs of SI technique. Encoding of Boolean to Decimal or Decimal to Boolean is presented in (**a**), **b** shows Boolean to decimal encoding where SI technique uses decimal inputs to evaluate objective function, and **c** shows decimal to Boolean encoding where SI technique uses Boolean inputs to evaluate objective function

different strings. In part (b) of Fig. 3 shows a single string to encode into Decimal value. When application control parameter values are Decimal, they are either defined as continuous or discrete values within some range. Each individual value of control parameter is encoded into Boolean value of predefined bit size. In part (c) of Fig. 3 shows encoding into Boolean value of bit size 3. Each Boolean string is considered as different input to multi-dimensional objective function of SI technique or considered as single Boolean string for uni-dimensional objective function.

Despite of different encoding schemes, it is better to consider such a SI technique which uses similar values as control parameters of application to avoid possible pitfalls. Though such decisions are constraint to the application. Most of the SI techniques including PSO, ACO and GSA use Decimal representation. Even if encoding is not needed when inputs of SI technique and application control parameters are similar, results obtained may not be good. Hence, strategic changes (discussed in Sect. 6) to existing technique is adopted. In some cases require both kinds of encoding schemes. For example feature selection problem uses both Boolean and Decimal representation. Boolean representations are done by representing 1 if particular feature is selected and 0 otherwise. String of these binary values represented as individual particles or solutions to the SI techniques (Chakraborty and Chakraborty

2013; Selvaraj and Janakiraman 2013). However, assigning of weights to each feature is done through Decimal representation (Singh et al. 2014). Features in the vector indicate whether they are selected or not with predefined threshold values. Each vector represents a solution. Another example is in (Das et al. 2014), particles are considered as individual network, which comprises links and weights of these links between two neurons and transfer function. Presence and absence of link can be represented as Boolean, weights can be represented as Decimal and transfer function may give Boolean values. Hence author used both kinds of encoding to represent particles.

Improperly detected control parameters may not be suitable for the application and may show unexpected results. Even if suitable control parameters are indentified, it is a matter of judgment whether to encode as Decimal or Boolean. Sometimes, applications may have control parameters that can not be represented either as Decimal or Boolean. In such cases needed some functions to transform them either to Decimal or to Boolean. Moreover, some applications don't even have clear control parameters. Implementer needs to be identified suitable control parameters from the application in those cases.

6 Strategic Changes

Over the decades SI techniques have been undergone several strategic changes to improve efficiency, reliability, scalability and solution quality. Those changes were carried out irrespective of applications. Often used benchmark functions to test those techniques. But, when those improved variants are applied to some real life problems, found that they are not comply with the problem. It seems that they perform better on some specific problems. Even though performed well on benchmark functions, it cannot be generalize that same technique will perform better in all problems as stated in no free lunch (NFL) theorem (Wolpert and Macready 1997). Same method may show best result on some problem or even on standard benchmark functions but may show worst result in other problems. Hence, problem specific improvements were done to suit real life problem, but always have exceptions. For example PSO often found better to feature selection when hybridized with other approaches, but on the contrary hybridized ACO shows degraded results on the same problem (Kothari et al. 2012).

Introduction of new strategy to the existing technique have added extra overhead to the technique. Though shows better results than previous actual techniques, they become more complex. Implementer has to look over more constraints in order to realize actual variant of the technique. If we take a look at strategic changes of PSO, it has undergone several modifications to the PSO originally introduced by Kennedy and Eberhart (1995). Following the original version of PSO discrete variant was also introduced by Kennedy and Eberhart (1997). Concept of constriction factor was introduced by Cleric and Kennedy (2002). Fully informed PSO (FIPS) was proposed by Mendes (2004), where every particle is informed about others experience. This requires too high computation time. Linearly varying coefficients along with linearly varying weight introduced in Ratnaweera et al. (2004). Implantation of linear variation requires two parameters to define upper and lower limit. Hence, to perceive this concept requires six additional parameters, two for each of three core parameters. Similar parameter related issues can be noted in hybridized versions with Chemical Reaction Optimization (CRO) (Nguyen et al. 2014), which requires eight parameter to maintain, five for CRO and three for PSO. Orthogonal learning approach proposed by Zhan and his colleague (Zhan et al. 2011), where used concept of orthogonal experiment design (OED). Factor analysis part of this strategy consumes much computation time as compared to PSO without OED. However, this version have resulted high quality solutions. Recently, Ganapathy and his colleague proposed ortho-cyclic circle concept (Ganapathy et al. 2014) to PSO, which considers two level particles. Particles are identified as belonging to a group referred as circles. These circles are also considered as high level particles. Maintenance of two level particle and identification of circles has added extra burden and make concept little bit complex. Similar problem can be noted in case of ACO. Min-Max ant system (MMAX) (Stützle and Hoos 2000) added bound on pheromone deposit, which required to maintain two extra parameters. In rank-based ant system (ASrank) (Bullnheimer et al. 1997), solutions are ranked in accordance of their cost. Ants are allowed to deposit pheromones proportional to their derived solution cost. Ranking of ants based on the solution cost is a time consuming process.

Advancement of strategy of a technique implies accumulation of new concept, which may be complex in real sense. Basic and original version of SI techniques found to be simpler than advanced versions of the same method. Hence, instead of made strategic changes sometime parameters of existing SI technique are tuned to improve results. Parameter tuning schemes on various SI techniques are explained in Sect. 7. Moreover, Complex advance version cannot fit into the applications easily. Hence, basic versions are used enormously even though results are comparatively degraded than advanced versions. Methodology of SI technique (it may be advanced version or basic version) often found to be incompatible with applications, those cases the strategy has to be modified so that it comply with that application.

7 Parameter Tuning

SI techniques have specific parameters to perceive their strategic needs by tuning those parameters. Collective aim of such tuning is to improve efficiency, raise effectiveness, increase applicability of these techniques to practical problems. In perspective of applications, parameters of SI techniques are tuned to comply with application environment. Every application has its own environment and needs to solve associated problems with respect to those environment. Techniques incorporated (including SI techniques) to solve such problems have to be practised in terms of application environment. So, parameter values of any SI technique has to be tuned in accordance of environment of the application. Some application may have constraints for those cases require additional parameters to fit the technique in it. Hence, such application specific parameter tunings cannot be generalized for other applications.

To account maximum application domains for any SI technique, require generalized parameters settings which can be used in any application and also needs to ensure that those setting will result better. So, most of such tunings are carried out with benchmark functions to test tuned parameter. Benchmark function based parameter adjustments can be considered as generalized settings for applications of different domain.

Generalized parameter tuning can be of two kinds, constant value based tuning and strategic tuning. Universally used PSO have three parameters C_1, C_2, ω. Initial version of PSO had only two parameters C_1 and C_2. Later on suggested one more parameter because, uncontrolled velocity often led to move particles much ahead of optimal solution as expected, which implies divergence of particles from the objective, resulting slow convergence of the process to the optimal solution. Shi and Eberhart (1998a) has observed that velocity of particle has to be controlled in order to control search scope of particle. To overcome this problem they introduced new parameter inertia weight to control velocity of particle. Cleric and Kennedy has done similar control by introducing constriction factor c instead of inertia weight (Clerc and Kennedy 2002). They showed values of two coefficient have to be 1.4962, while inertia weight has to be 0.7968. Kennedy and his colleague (Kennedy et al. 2001) shows value of C_1 and C_2 has to be 2. Apart from these constant value based tuning, parameters of PSO also tuned strategically to act more friendly to the applications of different domains. Shi and Eberhart (1999) has varied inertia weight ω linearly. Sometime added new parameters which might be for realization of strategic tuning. For example, to vary ω of PSO there has to be upper bound ω_{up} and lower bound ω_{low}. These newly introduced parameter also has to be tuned to obtain better range of ω for varying linearly. Shi and Eberhart (1998b) showed values of ω_{up} and ω_{low} are 0.9 and 0.4 respectively. Hence, tuning of parameters not only done as constant value based or strategic tuning, but also can be done both simultaneously. Parameter values obtained through constant value based tuning may not suit some application, but strategic tuning can fit all application which utilizes SI technique.

8 Discussion

Adaptation have to be done both from application side as well as SI techniques to comply with each other. In order to fit SI techniques into diverse applications of different domains have to be generalized. Such generalization has to be done in SI techniques with respect to any application or benchmark problems. Generally, control parameters of application varies with different application domain and the environment of the application. SI techniques' parameters and strategies have to be adjusted in accordance of control parameters of any application. Hence, parameters and strategy of an SI technique have to be tuned in such a way so that it can be grasped in any application without too many alternations both in SI techniques as well as application control parameters.

Structure of most SI techniques are very similar and can be generalized in single framework. Most of SI techniques treat solutions in similar way, only difference is in their strategy. Generally, SI techniques represent solutions in Boolean string or Decimal string. Some of the SI techniques consider multi level solution representation. Hence, to incase SI techniques in a generalized framework, solution representation has to be done in perspective of strategy incorporated into the approach. Once presentation of initial swarm is done with respect to strategy, swarm can be updated strategically in a common loop by absorbing strategy of corresponding approach.

SI techniques always have to go hand in hand with its application to solve common problems. In order to mastery any of the SI techniques in applicability to applications of various domains, compatibility of its strategy with the application has to be increased. Applicability of any SI technique can be increased if its parameters and strategy are generalized. Major issues observed for generalization of SI techniques in perspective of applications can be stated as below:

- Diversity of applications.
- Numerous techniques available creating confusion.
- Constraints in applications.
- Encoding of application.
- Multiple variants of single technique.

Issues related to applicability of SI techniques to any application:

- Selection of suitable technique for an application is very important task and always have confusions. Generally, simpler and effective techniques are the good criterion for selection of SI technique.
- Normally have two objectives behind the utilization of any SI technique, either it is the objective function value or the control parameters.
- Application associated problems have to be encoded in perspective of the selected technique to realize problem in that technique.
- Problem characteristics have be similar to problems where selected technique performs comparatively better.
- During implementation may face confusion regarding which one to adapt, is it the application or technique.
- Adaptation of application in terms of technique selected will be the better option, but of course selected technique has to be good one.
- Parameters of selected technique have to be tuned with respect to application to get good quality solution.

Advantages of incorporation of SI techniques into applications:

- Concept of some techniques such as PSO are very simple and competitive to other approaches.
- Simpler techniques just need to identify application problem's control parameters in order to consume strategy of the technique.
- Techniques with less parameters and simple strategy can easily fit into variety of applications of different domain.
- Generally SI techniques requires only a few steps i.e. initialize swarm and iterate with corresponding strategy.
- SI provides framework to solve collective or distributed problems through a global view model with decentralized control.
- Can easily solve non-linear complex problems.
- Can easily solve hard problems such as NP-hard and NP-complete problems, shows high quality approximated solution.
- Problem can be solved without any guidance through local shared information and cooperative learning of SI technique.

Disadvantages of SI techniques in perspective of applications:

- Application complexity creates problems during absorption of SI techniques.
- Some advanced variants show very high quality solutions but technique become complex and difficult to implement.
- SI techniques always give approximated result and have no grantee of good solution.
- Most of the SI techniques have parameters which have to be tuned in order to get better result.

9 Conclusion

Swarm based techniques have been applied enormously in real life problems. Most of these techniques are inspired by nature and have a common aim to optimize. Technology advances have led to solve problems more efficiently and more effectively. To deal with the requirements of such advanced technology, associated problem has to solve intelligently. SI techniques have been one of the repository for such intelligence. To align with current technology, SI techniques have also been improved. Though these advanced variations shown improvement in quality of solutions, in parallel increases complexity of the approach and time requirement. Also, creates confusion regarding selection of one from these too many techniques along with their variants. Such application and SI techniques related issues have been addressed in this chapter. Generalization of SI techniques can be one of the solution to such issues, but due to constraints present at variety of applications this solution can not be materialized fully. Another solution being noticed to this problem is the utilization of simpler and relatively effective variant to applications, which not only meets quality requirements but also can easily deal with implementation related issues. Despite of numerous difficulties SI techniques gaining popularity and also consuming enormously in various applications.

References

Al Rashidi, M. R., & El-Hawary, M. E. (2009). A survey of particle swarm optimization applications in electric power systems. *IEEE Transactions on Evolutionary Computation, 13* (4), 913–918.

Alzalg, B., Anghel, C., Gan, W., Huang, Q., Rahman, M., & Shum, A. (2011). A computational analysis of the optimal power problem. In *Institute of Mathematics and its Application*. IMA Preprint Series 2396. University of Minnesota.

Amit, Y. (2002). *2D object detection and recognition: Models, algorithms, and networks*. Cambridge: MIT Press.

Borwein, J. M. & Lewis, A. S. (2010). *Convex analysis and nonlinear optimization: Theory and examples* (2nd ed.). Berlin, Springer.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1997). A new rank based version of the ant system. A computational study. *SFB Adaptive Information Systems and Modelling in Economics and Management Science, 7*, 25–38.

Chakraborty, B., & Chakraborty, G. (2013). Fuzzy consistency measure with particle swarm optimization for feature selection. In *2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 4311–4315), October 13–16, 2013, Manchester. IEEE. doi:10.1109/SMC.2013.735.

Chandra Mohan, B., & Baskaran, R. (2012). A survey: Ant colony optimization based recent research and implementation on several engineering domain. *Expert Systems with Applications, 39*(4), 4618–4627.

Chandrasekhar, U., & Naga, P. (2011). Recent trends in ant colony optimization and data clustering: A brief survey. In *2011 2nd International Conference on Intelligent Agent and Multi-agent Systems (IAMA)* (pp. 32–36), September 7–9, 2011, Chennai. IEEE. doi:10.1109/IAMA.2011.6048999.

Chu, S.-C., Roddick, J. F., & Pan, J.-S. (2003). Parallel particle swarm optimization algorithm with communication strategies. *Journal of Information Science and Engineering, 21*(4), 809–818.

Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*(1), 58–73.

Das, G., Pattnaik, P. K., & Padhy, S. K. (2014). Artificial neural network trained by particle swarm optimization for non-linear channel equalization. *Expert Systems with Applications, 41*(7), 3491–3496.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Dorigo, M. (1992). Optimization, learning and natural algorithms. *Ph.D. Thesis*, Politecnico di Milano, Italy.

Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *Computational Intelligence Magazine, IEEE, 1*(4), 28–39.

Eslami, M., Shareef, H., Khajehzadeh, M., & Mohamed, A. (2012). A survey of the state of the art in particle swarm optimization. *Research Journal of Applied Sciences, Engineering and Technology, 4*(9), 1181–1197.

Ganapathy, K., Vaidehi, V., Kannan, B., & Murugan, H. (2014). Hierarchical particle swarm optimization with ortho-cyclic circles. *Expert Systems with Applications, 41*(7), 3460–3476.

Gonzalez, R. C., & Woods, R. E. (2002). *Digital image processing*. New Jersey: Prentice Hall.

Graefe, V., & Efenberger, W. (1996). A novel approach for the detection of vehicles on freeways by real-time vision. In *Proceedings of the 1996 IEEE Intelligent Vehicles Symposium* (pp. 363–368), September 19–20, 1996, Tokyo. IEEE. doi:10.1109/IVS.1996.566407.

Hancer, E., Ozturk, C., & Karaboga, D. (2012). Artificial bee colony based image clustering method. In *2012 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1–5), June 10–15, 2012, Brisbane. IEEE. doi:10.1109/CEC.2012.6252919.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. Michigan: University Michigan Press.

Honghao, C., Zuren, F., & Zhigang, R. (2013). Community detection using ant colony optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC)* (pp. 3072–3078), June 20–23, 2013, Cancun. IEEE. doi:10.1109/CEC.2013.6557944.

Itti, L. (2000). Models of bottom-up and top-down visual attention. *PhD thesis*, California Institute of Technology.

Janacik, P., Orfanus, D., & Wilke, A. (2013). A survey of ant colony optimization-based approaches to routing in computer networks. In *2013 4th International Conference on Intelligent Systems Modelling and Simulation (ISMS)* (pp. 427–432), January 29–31, 2013, Bangkok. IEEE. doi:10.1109/ISMS.2013.20.

Kameyama, K. (2009). Particle swarm optimization-a survey. *IEICE Transactions on Information and Systems, 92*(7), 1354–1361.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report, (Technical report-tr06), Erciyes university, Engineering Faculty, Computer Engineering Department.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks*, (Vol. 4, pp. 1942–1948), 27 Nov 1995–2001 Dec 1995, Perth. IEEE. doi:10.1109/ICNN.1995.488968.

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on*, Vol. 5, October 12–15, 1997, Orlando, IEEE (pp. 4104–4108). doi:10.1109/ICSMC.1997.637339.

Kennedy, J. F., Kennedy, J., & Eberhart, R. C. (2001). *Swarm intelligence*. Los Altos: Morgan Kaufmann.

Kothari, V., Anuradha, J., Shah, S., & Mittal, P. (2012). A survey on particle swarm optimization in feature selection. In *Global Trends in Information Systems and Software Applications* (pp. 192–201). Berlin: Springer

Kulkarni, R. V., & Venayagamoorthy, G. K. (2011). Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, 41*(2), 262–267.

Kumar, G. K., & Jayaraman, V. (2013). Clustering of complex networks and community detection using group search optimization. *CoRR*, (abs/1307.1372).

Matoušek, J., & Gärtner, B. (2007). *Understanding and using linear programming*. 7th edition. Berlin: Springer.

Mendes, A. (2004). *Building generating functions brick by brick*. San Diego: University of California.

Monteiro, M. S., Fontes, D. B., & Fontes, F. A. (2012). Ant colony optimization: a literature survey. Technical report, Universidade do Porto, Faculdade de Economia do Porto.

Nguyen, T. T., Li, Z., Zhang, S., & Truong, T. K. (2014). A hybrid algorithm based on particle swarm and chemical reaction optimization. *Expert Systems with Applications, 41*(5), 2134–2143.

Ranaee, V., Ebrahimzadeh, A., & Ghaderi, R. (2010). Application of the pso–svm model for recognition of control chart patterns. *ISA Transactions, 49*(4), 577–586.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). Gsa: A gravitational search algorithm. *Information Sciences, 179*(13), 2232–2248.

Ratnaweera, A., Halgamuge, S., & Watson, H. C. (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation, 8*(3), 240–255.

Rechenberg, I. (1994). Evolution strategy *Computational intelligence: Imitating life*, (pp. 147–159). Piscataway: IEEE Press.

Reyes-Sierra, M., & Coello, C. C. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research, 2*(3), 287–308.

Ruszczyński, A. P. (2006). *Nonlinear optimization* (Vol. 13). NJ: Princeton University Press.

Schutte, J. F., Reinbolt, J. A., Fregly, B. J., Haftka, R. T., & George, A. D. (2004). Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering, 61*(13), 2296–2315.

Schwefel, H.-P. (1994). On the evolution of evolutionary computation, *Computational intelligence: Imitating life*, (pp. 116–124). IEEE Press: Piscataway.

Selvaraj, G., & Janakiraman, S. (2013). Improved feature selection based on particle swarm optimization for liver disease diagnosis. In *Swarm, Evolutionary, and Memetic Computing* (pp. 214–225). Berlin: Springer.

Shah-Hosseini, H. (2008). Intelligent water drops algorithm: A new optimization method for solving the multiple knapsack problem. *International Journal of Intelligent Computing and Cybernetics, 1*(2), 193–212.

Shah-Hosseini, H. (2009). The intelligent water drops algorithm: A nature-inspired swarm-based optimization algorithm. *International Journal of Bio-Inspired Computation, 1*(1), 71–79.

Shi, Y., & Eberhart, R. (1998a). A modified particle swarm optimizer. In *The 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence* (pp. 69–73), May 4-9, 1998, Anchorage. IEEE. doi:10.1109/ICEC.1998.699146.

Shi, Y., & Eberhart, R. C. (1998b). Parameter selection in particle swarm optimization. In *Evolutionary Programming VII*, March 25-27, 1998, San Diego, California, USA, Springer (pp. 591–600). doi:10.1007/BFb0040810.

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, Vol. 3, July 6-9, 1999, Washington, IEEE. doi:10.1109/CEC.1999.785511.

Singh, N., Arya, R., & Agrawal, R. (2014). A novel approach to combine features for salient object detection using constrained particle swarm optimization. *Pattern Recognition, 47*(4), 1731–1739.

Stützle, T., & Hoos, H. H. (2000). Max–min ant system. *Future generation computer systems, 16* (8), 889–914.

Todd, M. J. (2002). The many facets of linear programming. *Mathematical Programming, 91*(3), 417–436.

Vanneschi, L., Codecasa, D., & Mauri, G. (2012). An empirical study of parallel and distributed particle swarm optimization. In *Parallel Architectures and Bioinspired Algorithms* (pp. 125–150). Berlin: Springer.

Wiki (2014). Mathematical optimization. http://en.wikipedia.org/wiki/Mathematical_optimization. Accessed 2014-02-30.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 67–82.

Yan, X.-S., Li, C., Cai, Z.-H., & Kang, L.-S. (2005). A fast evolutionary algorithm for combinatorial optimization problems. In *Proceedings of 2005 International Conference on Machine Learning and Cybernetics* (Vol. 6, pp. 3288–3292) August 18-21, 2005, Guangzhou. IEEE. doi:10.1109/ICMLC.2005.1527510.

Yang, B., Chen, Y., & Zhao, Z. (2007). Survey on applications of particle swarm optimization in electric power systems. In *IEEE International Conference on Control and Automation (ICCA 2007)* (pp. 481–486), May 30 2007–June 1 2007, Guangzhou. IEEE. doi:10.1109/ICCA.2007.4376403.

Zhan, Z.-H., Zhang, J., Li, Y., & Shi, Y.-H. (2011). Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation, 15*(6), 832–847.