Swarm Intelligence in Multiple and Many Objectives Optimization: A Survey and Topical Study on EEG Signal Analysis

B.S.P. Mishra, Satchidanand Dehuri and Sung-Bae Cho

Abstract This paper systematically presents the Swarm Intelligence (SI) methods for optimization of multiple and many objective problems. The fundamental difference of Multiple and Many Objective Optimization problems have been studied very rigorously. The three forefront swarm intelligence methods, i.e., Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), and Artificial Bee Colony Optimization (ABC) has been deeply studied to understand their ways of solving multiple and many objective problems distinctly. A pragmatic topical study on the behavior of real ants, bird flocks, and honey bees in solving EEG signal analysis completes the survey followed by discussion and extensive number of relevant references.

1 Introduction

Multiple and many objective optimization problems are alarming of high importance in recent scientific and the industrial world [1]. Some of the practical optimization problems include flight rescheduling [2], shape optimization [3], mobile network design [4], and minimization of shooting failure probability in the weapon target assignment [5], etc. In practice, the number of objectives in a multiple optimization problem are restricted with ≤ 3. However, number of objectives are at least 4 in the case of many objective optimization. Irrespective of the number of objects i.e.,

B.S.P. Mishra (✉)
School of Computer Engineering, KIIT University, Bhubaneswar-24, Odisha, India
e-mail: mishra.bsp@gmail.com

S. Dehuri
Department of Systems Engineering, Ajou University, San 5, Woncheon-dong, Yeongtong-gu, Suwon 443-749, South Korea
e-mail: satchi@ajou.ac.kr

S. Cho
Soft Computing Laboratory, Department of Computer Science, Yonsei University, 262 Seongsanno, Seodaemun-gu 120-749, South Korea
e-mail: sbcho@cs.yonsei.ac.kr

S. Dehuri et al. (eds.), *Multi-objective Swarm Intelligence*,
Studies in Computational Intelligence 592, DOI 10.1007/978-3-662-46309-3_2

whether multi or many the objectives are conflicting in nature. Therefore, in current intelligent computational era several evolutionary [6] and swarm based meta heuristic algorithms [7] (simulating natural phenomenon) have been proposed to give a Pareto optimal solution. These Pareto optimal solutions achieve a tradeoff. They are solutions for which any improvement in one objective results in worsting of at least other objectives. A decision vector $\overrightarrow{\mathrm{dv}} \in S$, where S is decision space, is said to be Pareto optimal when there is no other $\overrightarrow{\mathrm{dv}'} \in S$, that dominates $\overrightarrow{\mathrm{dv}} \in S$. An objective vector $\overrightarrow{\mathrm{ov}} \in \overrightarrow{\mathrm{OV}}$ is said to be Pareto optimal when there is no other $\overrightarrow{\mathrm{ov}'} \in \overrightarrow{\mathrm{OV}}$, that dominates $\overrightarrow{\mathrm{ov}'}$. The Pareto optimal solution consists of a set of solutions known as a Pareto optimal set. It presents a complete set of solutions for a multi objective problem. A plot of entire Pareto sets in the design space by considering design objectives as axis it gives a Pareto front. All the Pareto optimal points are non-dominated, so they are also termed as non dominated points. In case of many objective optimization problems, the set of non dominated solutions increases, so the Pareto front becomes very hazy, hence it becomes very critical for handling such huge number of solutions. Hence, recently there is a growing interest in the community of many objective optimization researchers for designing of efficient methods to handle large number of non-dominated solutions during optimization.

On the other hand, swarm based techniques for optimization problems are leaping forward by modeling the swarming behavior of animals in the nature. Their self-adaptive, self-organized, and full of vigor with vitality attracts researchers of many disciplines [8, 9] to apply diversified domain. In this paper, we discuss on the swarm based approaches to handle the multi and many objective problems.

Now-a-days high density EEG systems are available at an affordable costs, with which large number of attributes involve and create opportunities for effective analysis. Most of the methods of analysis explicitly/implicitly follows a pattern recognition tasks. These analysis have important applications in Brain Computer Interface (BCI), epileptic seizure identification, monitoring of sleep disorder and patients in critical condition in the ICUs, etc. However, automated analysis of EEG signal in different situations is a great challenge because of the volume of dataset and dynamic nature of the signals with high temporal resolutions. Further, the automated analysis of EEG signal can be viewed as a multiple objective problems. In contrast to non-population and non-stochastic based approaches, swarm intelligence strives to achieve better compromised results by balancing both exploration and exploitation of large search space in a tolerable amount of time. This paper puts a light on the use of swarm based techniques for analyzing EEG signals which is one of the BCI agents.

1.1 Outline of Swarm Intelligence

On the set of probabilistic meta-heuristic approaches, swarm intelligence is attracting lot of attentions [6, 7]. The swarm intelligence (SI) algorithm is functioning by the collective behavior of decentralized, and self organized agents. The term swarm is used in a general manner to refer to any restrained collection of interacting agents or individuals. The classical example of swarm is ants, birds flocks and honey bees swarming around their hive. Swarm intelligence works on two basic principles: self organization and stigmergy (e.g., Fig. 1).

i. **Self organization**: This can be characterized by three parameters like structure, multi stability and state transitions. In swarms, Bonabeau et al. [1], interpreted the self-organization through four characteristics [10]: (i) positive feedback, (ii) negative feedback, (iii) fluctuations, and (iv) multiple interactions.
ii. **Stigmergy**: It means *stimulation by work*. Stigmergy is based on three principles: (i) work as a behavioral response to the environmental state; (ii) an environment that serves as a work state memory; and (iii) work that does not depend on specific agents.

According to Millonas in 1994 [9] five principles have to be satisfied by a swarm in order to be useful in any optimization problem.

i. **The proximity principle**: The swarm should be able to do simple space and time computations, to understand and conceptualize information more quickly.
ii. **The quality principle**: The swarm should be able to respond to quality factors in the environment such as the quality of foodstuffs or safety of the location.
iii. **The principle of diverse response**: The swarm should not allocate all of its resources along excessively narrow channels and it should distribute resources into many nodes.
iv. **The principle of stability**: The swarm should not change its mode of behavior upon every fluctuation of the environment.
v. **The principle of adaptability**: The swarm must be able to change behavior mode when the investment in energy is worth the computational price.

However, it is to be noted that the above principles are applicable for single, multiple, and many-criteria optimization. The description of different swarm inspired techniques like Wasp Colony Optimization (WCO), and Termite Colony Optimization (TCO), Bacteria Foraging Optimization (BFO) can be obtained in [11–13]. Though there are several numbers of swarm based algorithm [14], are developed, but the first three algorithms enlisted in Table 1 is quite mature, so we concentrate our study on those algorithms.

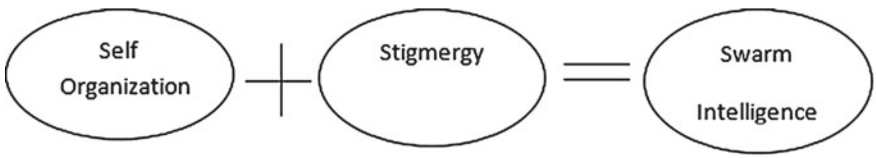

Fig. 1 Basic principles of swarm intelligence

Table 1 Swarm inspired algorithms

Name of the algorithm	Pioneer	Year of development	Motivation
ACO	M. Dorigo	1992	Ant colonies
PSO	J. Kennedy	1995	Group of birds
ABC	D. Karaboga	2005	Honey bees
WASPCO	P. Pinto	2005	Wasp
TCO	M. Roth and S.Wicker	2003	Termite
BATCO	Xin-She Yang	2010	Bat
BFO	K. M. Passino	2002	E. Coli and M. Xanthus

1.2 Multi-objective versus Many-objective Optimization

An optimization problem can be classified as a single, multi and many objective optimization problems based on the number of objectives. Again, we can classify the optimization problem in to two categories like constraint based and unconstrained. A single objective optimization problem can be stated as:

$$
\begin{aligned}
&\textbf{Problem with Constraint}\\
&\text{Minimize/Maximize } \mathrm{f}(\overrightarrow{\mathrm{x}})\\
&\text{Subject to,}\\
&\quad g_j(\overrightarrow{\mathrm{x}}) \geq 0,\ j = 1, 2, 3, \ldots, p\\
&\quad h_k(\overrightarrow{\mathrm{x}}) = 0,\ j = 1, 2, 3, \ldots, q\\
&\quad x_i^l \leq x_i \leq x_i^u;\ i = 1, 2, 3, \ldots, n
\end{aligned}
\tag{1}
$$

$$
\begin{aligned}
&\textbf{Problem without Constraint}\\
&\text{Minimize/Maximize } \mathrm{f}(\overrightarrow{\mathrm{x}})\\
&\quad x_i^l \leq x_i \leq x_i^u;\ i = 1, 2, 3, \ldots, n
\end{aligned}
\tag{2}
$$

A solution x is a vector of n decision variables: $\overrightarrow{\mathrm{x}} = (x_1, x_2, \ldots, x_n)$. In the case of problem with constraint, (e.g., Eq. 1) there are p inequality and q equality constraints associated and the terms $g_j(x)$ and $h_k(x)$ are called constraint functions. Although the inequality are treated as $\geq$ types, the $\leq$ constraints can also be considered in the above formulation by converting those to $\geq$ types simply by multiplying each constraint function by -1. However, there are problems, e.g., Eq. 2 which makes free from constraints but sometimes difficult to optimize, because of multi-modality, concavity, discrete, etc.

A multi-objective optimization problem can also be stated as constraint and unconstrained. (e.g., the general form is presented in Eqs. 3 and 4.

Problem with Constraint

$$\begin{aligned}
&\text{Minimize/Maximize } f(\vec{x}) = \{f_1(x), f_2(x), f_3(x)\} \\
&\text{Subject to,} \\
&\quad g_j(\vec{x}) \geq 0,\ j = 1, 2, 3, \ldots, p \\
&\quad h_k(\vec{x}) = 0,\ j = 1, 2, 3, \ldots, q \\
&\quad x_i^l \leq x_i \leq x_i^u;\ i = 1, 2, 3, \ldots, n
\end{aligned} \tag{3}$$

Problem without Constraint

$$\begin{aligned}
&\text{Minimize/Maximize } f(\vec{x}) = \{f_1(x), f_2(x), f_3(x)\} \\
&\quad x_i^l \leq x_i \leq x_i^u;\ i = 1, 2, 3, \ldots, n
\end{aligned} \tag{4}$$

In multi-objective optimization, the three objective functions of $F(\vec{x}) = \{f_1(x), f_2(x), f_3(x)\}$ can be either minimized or maximized or both. Like single objective optimization, the problem may have constraints or free from constraints. However, in this case, there is no single global optimal solution, rather a set of Pareto optimal solutions.

A multi-objective problem can be addressed broadly in three approaches, the first approach is weighted aggregated approach. In this case, we convert multi-objective problem into single objective problem, moreover, the most important concern is assignment of weights to the objectives, which is purely subjective. The second approach is known as a lexicographic approach. In this approach, specification of tolerance threshold and the corresponding degree of confidence for each objective is subjective and arbitrary. A third approach used is known as Pareto based approach, in this a trade-off of solutions known as non-dominated solutions are present on a front known as Pareto front. In this case, the user has his utmost autonomy to select the solution.

Most multi-objective optimization algorithms use the concept of domination to obtain a set of non-dominated solutions. The concept of domination is described in the following definitions (assuming, without loss of generality, the objective functions to be minimized).

Definition 1 Given two decisions or solution vectors $\vec{x}$ and $\vec{y}$, we say that decision vector $\vec{x}$ weakly dominates (or simply dominates) the decision vector $\vec{y}$ (denoted by $\vec{x} \leq \vec{y}$) if and only if $f_i(\vec{x}) \leq f_i(\vec{y})$ for $\forall i = 1, 2, 3$ (i.e., the solution $\vec{x}$ is no worse than $\vec{y}$ in all objectives) and $f_i(\vec{x}) < f_i(\vec{y})$ for at least one $i \in 1, 2, 3$ (i.e., the solution x is strictly better than y in at least one objective).

Definition 2 A solution $\vec{x}$ strongly dominates a solution $\vec{y}$ (denoted by $\vec{x} < \vec{y}$), if solution $\vec{x}$ is strictly better than $\vec{y}$ in all 3 objectives. However, if a solution $\vec{x}$ strongly dominates a solution, $\vec{y}$, the solution $\vec{x}$ also weakly dominates solution $\vec{y}$, but not vice versa.

Definition 3 The decision vector $\overrightarrow{x} \in P$ (where P is the set of solutions or decision vectors) is non-dominated with respect to set P, if there does not exist another $\overrightarrow{x^1} \in P$ such that $f_i^1(\overrightarrow{x}) \leq f_i(\overrightarrow{x})$.

Definition 4 Among a set of solution or decision vectors P, the non-dominated set of solutions or decision vectors P′ are those that are not dominated by any member of the set P.

Definition 5 A decision variable vector $\overrightarrow{x} \in P$ where P is the entire feasible region or simply the search space is Pareto optimal if it is non-dominated with respect to P.

Definition 6 When the set P is the entire search space, the resulting non-dominated set P′ is called the Pareto optimal set. In other words, $P' = \{\overrightarrow{x} \in P | \overrightarrow{x} \text{ is } \textit{Pareto optimal}\}$. The non-dominated set P′ of the entire feasible search space P is the global Pareto optimal set.

Definition 7 All Pareto optimal solutions in a search space can be joined with a curve (in two-objective space) or with a surface (more than two objective space), or with a pattern (with more than three objectives). This curve or surface is termed as a Pareto optimal front or simply Pareto front. In other words, $PF = \{f(\overrightarrow{x}) | \overrightarrow{x} \in P'\}$.

Now-a-days, many of the optimization problems are associated with more than three number of objectives. Any optimization problem which involves more than three number of objectives is known as many objective optimization problems. These objectives are also conflicting in nature. It states that many objectives need to be addressed simultaneously in an optimization problem.

Generally, a many-objective optimization problem is written as follows:

Problem with Constraint

Minimize/Maximize $f(\overrightarrow{x}) = \{f_1(x), f_2(x), f_3(x), \ldots, f_m(x)\}$

where, $m > 3$.

Subject to,

$$g_j(\overrightarrow{x}) \geq 0,\ j = 1, 2, 3, \ldots, p$$
$$h_k(\overrightarrow{x}) = 0,\ j = 1, 2, 3, \ldots, q$$
$$x_i^l \leq x_i \leq x_i^u;\ i = 1, 2, 3, \ldots, n \qquad (5)$$

Problem without Constraint

Minimize/Maximize $f(\overrightarrow{x}) = \{f_1(x), f_2(x), f_3(x), \ldots, f_m(x)\}$

$$x_i^l \leq x_i \leq x_i^u;\ i = 1, 2, 3, \ldots, n \qquad (6)$$

where $F(\overrightarrow{x})$ is the m-dimensional objective vector, $f_i(\overrightarrow{x})$ is the ith objective to be minimized, $\overrightarrow{x}$ is the decision vector and Ω is the feasible search space. A many-objective based problem can be solved broadly in two approaches.

Preference ordering approach: This approach addresses the approximation/optimization of many objective problems by including the techniques like reducing the number of non-dominated points and assigning different ranks to non-dominated points, without avoiding user preference information.

Objective Reduction Approach: In this approach by using linear or nonlinear algorithms, the objectives are reduced from many to multi-objective. This leads to higher search efficiency and lower computational cost.

As preference ordering approach is widely popular among the research community, so our survey mostly concentrated on this approach only. This approach can be broadly classified into two categories Pareto dominance based approach and non-Pareto dominance based approach. The Pareto dominance based approach includes the techniques like:

Modification of Pareto dominance: In this approach the selection pressure is increased over the Pareto front by using the epsilon-dominance approach. The detail procedure can be obtained from [15–18].

Introduction of different ranking method: In this method different ranks are assigned to the non-dominated solutions by establishing different relations among them. In this technique, different approaches like favour relation, average ranking method, epsilon-dominance and fuzzy pareto dominance methods are widely used [19–23].

Non-Pareto dominance based approach include techniques like indicator function [24–26, 28–30, 175], scalarizing function (a kind of weighted sum approach) [31–37], and preference information [27, 38–44]. Out of the above three approaches indicator function approach is widely used. There are a number of performance indicator available to measure the quality of approximation to true Pareto front, such as hyper volume, Δp, and R2. Hypervolume [45, 176] and Δp indicators are associated with the drawback of high computational cost as the number of objective increases. R2 indicator is associated with less computational cost. Therefore, our main concern is to focus on this operator [46, 47]. R2 [48] indicator is defined as:

$$R2(ND, U) = -\frac{1}{|U|} \sum_{u \in U \, \vec{n_l} \in ND}^{max} \{u(\vec{n_l})\} \tag{7}$$

where, ND contains set of non-dominated solutions, U is set of utility functions. As utility function is defined over the weight vector, hence further we can replace U with W.

$$\begin{aligned} R2(ND, W) &= -\tfrac{1}{|W|}\left(\sum_{\vec{w} \in W}\left(\max_{\vec{n_l} \in ND}\left\{-\max_{i\in\{1,\ldots,m\}} w_i \left|\frac{n_i - z_i^*}{z_i^{nad} - z_i^*}\right|\right\}\right)\right) \\ &= \tfrac{1}{|W|}\left(\sum_{\vec{w} \in W}\left(-\max_{\vec{n_l} \in ND}\left\{-\max_{i\in\{1,\ldots,m\}} w_i \left|\frac{n_i - z_i^*}{z_i^{nad} - z_i^*}\right|\right\}\right)\right) \\ &\qquad (\because min(\vec{w}) = -max(-\vec{w})) \\ &= -\tfrac{1}{|W|}\left(\sum_{\vec{w} \in W}\left(\min_{\vec{n_l} \in ND}\left\{\max_{i\in\{1,\ldots,m\}} w_i \left|\frac{n_i - z_i^*}{z_i^{nad} - z_i^*}\right|\right\}\right)\right) \end{aligned} \tag{8}$$

where, z_i^* is a lower bound of all the objective functions, z_i^{nad} is an upper bound of each objective function. The size of a weight vector and reference vector is equal to number of objectives.

Now the non-dominated sorting can be done based on the utility function adopted. The main objective is to group solutions that optimize the set of chosen utility functions and place such solutions as top rank.

Such points then be removed and the 2nd rank will be formed. In this case no-Pareto dominance is used. Now, the resultant rank equation is:

$$Rank_d = U_{\vec{w} \in W} \min_{\vec{n} \in ND/B_d} \left\{ \max_{i \in \{1,\dots,m\}} w_i \left| \frac{n_i - z_i^*}{z_i^{nad} - z_i^*} \right| \right\} \tag{9}$$

where, $B_d = \{U_x Rank_x | d \geq 2, i < x < d\}$ is the union of solutions with the lowest ranks.

The computational steps which are used to measure the quality of a Pareto front is given below.

Steps:

1. Archive Pareto optimal set.
2. Define Utility function by defining weight vector.
3. Call Tchebycheff(); // It calculates Tchebycheff value
4. Compare(); // Comapair Techbycheff value of each optimal solution with weight vector.
5. Rank solutions.

A small numerical example is illustrated to understand its efficiency and efficacy.

Problem (Weapon Target Assignment): In this problem the objective is to minimize simultaneously the probability of shooting failure and the number of weapons (or resources).

Figure 2, illustrates a typical Pareto front obtained to satisfy both the objectives. Let us see the influence of R_2 indicators for ranking solutions.

The non-dominated solutions of the Pareto front are enumerated as $ND = \{(1, 25), (2, 19), (3, 17), (4, 12), (5, 9.5), (6, 8), (7, 7.45), (8, 6.35)\}$. Let take weight vector $W = \{(0.1, 0.9), (0.4, 0.6), (0.6, 0.4), (0.9, 0.1)\}$. The value of z_i^* and z_i^{nad} derived from the ND as (1, 6.35), and (8, 25) respectively. The optimum Tchebycheff value of each solution calculated as: {0.1, 0.1285, 0.2284, 0.1817, 0.1520, 0.0795, 0.0857, 0.1} (reported in Table 2).

Now the ranking of the Pareto optimal solutions is made by comparing the Tchebycheff value and the weight vector. Since, solution n_1 and n_8 has the same value, therefore, the ranks of these two solutions can be computed using Manhattan norm. For example n_1 and n_8 has the same Tchebycheff value as 0.1, so by calculating the Manhattan norm, we find that n8 has a lower value than n_1. So, n_8 is better than n_1. If we see the Tchebycheff value from the Table 2, we can assign Rank 1 to n_6, n_7 as they are near to the weight vector value.

Table 2 Tchebycheff value of Pareto optimal set

Pareto optimal set (ND)	Number of weapons (F_1)	The probability of shooting failure (F_2)	Optimum Tchebycheff value
n_1	1	25.00	0.1000
n_2	2	19.00	0.1285
n_3	3	17.00	0.2284
n_4	4	12.00	0.1817
n_5	5	09.50	0.1520
n_6	6	08.00	0.0795
n_7	7	07.45	0.0857
n_8	8	06.35	0.1000

The rest of the paper is set out as follows: Sect. 2 discusses the optimization process of swarm intelligence in multiple conflicting objectives. A frame-work for optimizing many objectives based on swarm intelligence is discussed in Sect. 3. The study of swarm intelligence in EEG signal has been made in Sect. 4. Discussion and future research direction are made in Sect. 5.

2 Swarm Intelligence for Multi Objective Problems

In this section, we discuss the effectiveness of ACO, PSO, and ABC for optimizing multi-objective problems in Sect. 2.1, 2.2, and 2.3 respectively.

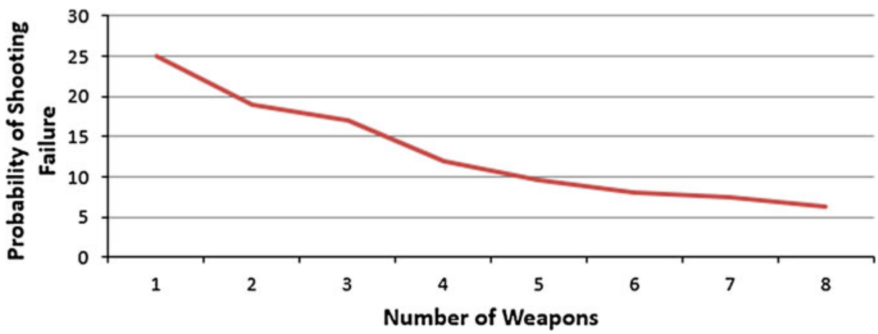

Fig. 2 Pareto front of weapon-target assignment

2.1 ACO for Multiple Objective Problems

Ant Colony Optimization [75] is a probabilistic metaheuristic technique in which a colony of artificial ants cooperates in finding better solutions to any optimization problem. Cooperation is the key element in the ACO. ACO algorithm can be used to solve both static and dynamic combinatorial optimization problems. Static problems are those problems in which the characteristics of the problems remain unchanged throughout its solution procedure. Dynamic problems in which the instance data, such as objective function values, decision parameters, or constraints, may change while solving the problems. ACO algorithm can be used to solve a multi-objective problem, in that case instead of one objective function two or three objectives functions evaluates competing criteria of solution quality. The steps to be followed while solving a problem using ACO can be enumerated as below:

Computational Steps:

1. The problem should be realized in the form of a weighted graph or sets of components and transitions on which ant can build the solutions.
2. Pheromone trails should be defined.
3. Heuristic preference for the ant should be defined while constructing the solution.
4. Choose a specific ACO algorithm and apply to the problem being solved.
5. Tune the parameter of the ACO algorithm.

The quality of the solution explored by ACO depends on the probability of proportion to the concentration of the pheromone. Where proportion of the pheromones leads to the length of the route. Concentration of the pheromones shows the number of ants used the route. The overall length of the route and the number of ants used the route influence the amount of pheromones accumulated on the route. Table 3 enumerates some of the variants of basic ACO algorithms.

Solving multiple-objective optimization problem (i.e., the number of objectives $\leq$3) using ACO is known as a Multi Objective Ant Colony Optimization (MOACO). On the other hand, solving problem with more than objectives (i.e., number of objectives $>$3) is popularly known as many objective optimization problems. Let us discuss some of the variants of MOACO.

In the last few years, many variants of algorithms are developed for handling the multi-objective problem by using ACO. While going through the literature it has noted that most of the multi objective algorithms are using the concept of Ant Colony System, Max-Min ant system, AQ ant system, etc. A generalized framework for MOACO is presented below:

Framework of MOACO:

1. Initialize ‘n’ pheromone matrices by using Initialize Pheromone trails ().
2. Make the archive of non-dominated solution empty at the beginning i.e., A $=\varphi$.
3. Execute Build solution once for each ‘m’ number of ants.
4. ‘m’ solution obtained at ‘t’ iteration is stored in s (t).
5. Update the Archive by considering s (t) and A and executing Archive update().

Table 3 Variants of ACO algorithms

Name of the ACO variant	Authors and year	Pheromone update rule	Technique adopted	Application area
Ant colony systems (ACS)	Dorigo et al. 1991 [201]	$\tau_{ij} = (1-\varphi)\tau_{ij} + \varphi\tau_0$	1. Local pheromone update technique. 2. Diversity in the search performed by subsequent ants during iterations. 3. Pseudo random proportional rule	Travelling salesman problem (TSP)
Max-min ant systems	Stutzle [179, 180]	$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}^{best}$	1. The best ant only update the pheromone trials. 2. Value of the pheromone is bounded. Zmax and Zmin that is lower and upper bounds are decided on the basics of analytical consideration	Quadratic assignment problem
Elitist ant system	Colorni et al. 1992 [202]	$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} + \Delta\tau_{ij}^{*}$	Global solution deposits pheromone on every iteration along with all other ants	Quadratic assignment problem
Ant Q	Gambardella et al. 1999 [54]	$\tau_0 = \gamma max_{j \in N_i^h}\{\tau_{ij}\}$	It involves multiple agents. These agents communicate, exchange information in the form of AQ-values	Irregation
Rank based ant system	Bullnheimer et al. 1997 [203]	$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij} + \Delta\tau_{ij}^{*}$	Solutions are ranked according to the density of the pheromone deposited for each solution	TSP
Best-worst ant system	Cordon et al. 2000 [204]	$\tau_{ij} = \tau_{ij} + \Delta\tau_{ij}$	It uses pheromone evaporation and best-worst pheromone updating	TSP

6. As per the new set ‘A’ of non-dominated solution, all the pheromone matrices are updated by considering objective values. Finally, after max iterations the archived set of non-dominated solution returned and can be found on the Pareto front.

In ACO, a MOP can be handled broadly in two different ways in ACO approach, by using multiple colonies or by using single ant colony [49, 152, 171, 173, 174, 177]. In later case, the pheromone matrix can be a single matrix for all objectives or multiple matrices. If they use multiple matrixes then an aggregation operation has to be carried out by weighted sum approach or weighted average or random approach. On the other hand if a multi-objective problem is handled using multiple ant colony then all the colonies can have a common pheromone information or according to the number of objectives each colony will have to be provided with that number of

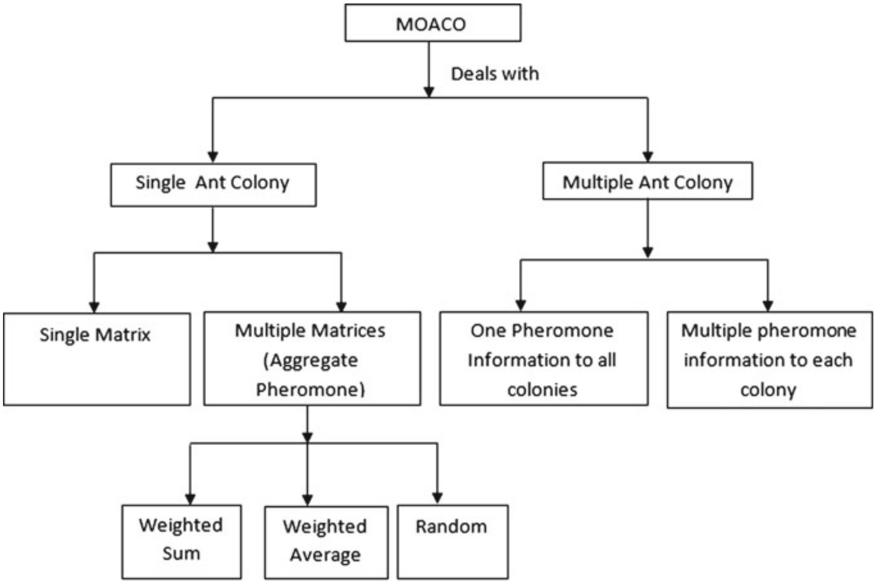

Fig. 3 Pictorial representation of different ways to handle MOP in ACO

pheromone information. A pictorial representation of handling a MOP in ACO is given in Fig. 3.

Most of the MOACO algorithms differ w.r.t. techniques adopting for pheromone trails updating, and definition of heuristic factor. Chaharsooghi and Kermani [50], have proposed a MOACO for handling multi objective resource allocation problem [181], in this algorithm they have increased the learning of ants and updated pheromone rules by simplifying probability calculations. Mariano and Morales [51, 178], have developed a MOACO by incorporating the reward mechanism in AQ ant algorithm for design of water distribution irrigation network. By, considering the pheromone updating technique Iredi et al. [52], have proposed a MOACO known as Bicriterion ANT for handling vehicle routing problem. Doerner et al. [53] proposed a Pareto based ACO algorithm for handling MOP known as P-ACO, he handled the MOP under the category of Pheromone trail matrix. Gambardella et al. [54] proposed a MOACO for handling vehicle routing problem with time window, in this they used the rank based approach with respect to the objectives, the MOACO is known as MACS-VRPTW. To handle multi-objective issues in a network MONACO is also proposed. Doerner et al. [55] proposed a COMPET ants MOACO to handle transportation problem. Gravel et al. [56], have suggested Multiple Objectives ACO Metaheuristics (MOACOM) for handling scheduling problems. Ali et al. [57], have proposed a MOACO for load balancing of distributed systems based on multiple ant colony optimization. Lopez-Ibanez and Stutzle [58, 59], have proposed a general framework by considering some MOACOs, which facilitates automatic configuration of algorithms for handling multi-objective problems. Few frequently used MOACO algorithms are analyzed and summarized in Table 4.

While discussing several MOACO algorithms in the literature, it has found that there are some key parameter exists which can take different values at different situations. One of such parameter is heuristic matrix this can take a single or multiple value in an algorithm. If the MOACO uses an aggregation method for pheromone update then it can take the form of weighted product, weighted sum or random. The setting of the weights can be done in a dynamic manner while solving the problem or it can be taken fixed value throughout the process. Evaluation of the solution can be Pareto based or non Pareto based. The archive Pareto can be maintained offline, online, or elitist. Some algorithms also don't maintain an archive. The pheromone matrix used by the ant can be a single matrix or multiple matrices. Individual pheromone updating can be done for each ant or global updating procedure can be adopted.

2.2 PSO for Multiple Objective Problem

Particle Swarm Optimization [172, 197] is a population-based search algorithm based on the simulation of the social behavior of birds within a flock. Kennedy and Eberhart [60] originally proposed the PSO algorithm for single objective optimization. As a basic principle, in PSO, a set of randomly generated particles in the initial swarm are flown (have their parameters adjusted) through the hyper-dimensional search space (problem space) according to their previous flying experience. Changes to the position of the particles within the search space are based on the social, psychological tendency of individuals to emulate the success of other individuals. Each particle represents a potential solution, to the problem being solved. The position of a particle is determined by the solution, it currently represents. The position of each particle is changed according to its own experience and that of its neighbors. These particles propagate towards the optimal solution over a number of generations (moves) based on larger amount of information about the problem space that is assimilated and shared by all members of the swarm. The PSO algorithm finds the global best solution by simply adjusting the trajectory of each individual toward its own best location (pbest) and the best particle of the entire swarm (gbest) at each time step (generation). In this algorithm, the trajectory of each individual in the search space is adjusted by dynamically altering the velocity of each particle according to its own flying experience and the flying experience of the other particles in the search space.

The position and velocity vector of the ith particle in the d-dimensional search space can be expressed as $\overrightarrow{\mathrm{x_i}} = \langle x_{i1}, x_{i2}, \ldots, x_{id}\rangle$ and $\overrightarrow{\mathrm{v_i}} = \langle v_{i1}, v_{i2}, \ldots, v_{id}\rangle$ respectively. According to a user defined fitness function, the best position of each particle (which corresponds to the best fitness value obtained by that particle at time t) is $\overrightarrow{\mathrm{p_i}} = \langle p_{i1}, p_{i2}, \ldots, p_{id}\rangle$, denoted as *pbest* and the fittest particle found so far in the entire swarm is $\overrightarrow{\mathrm{p_g}} = \langle p_{g1}, p_{g2}, \ldots, p_{gd}\rangle$, denoted as *gbest*. Then the new velocities and the new positions of the particles for the next fitness evaluation are calculated at time $t + 1$ using the following two self-updating equations:

Table 4 Summary of MOACO algorithms

Algorithm	Application area	Type of colony	Pheromone update technique	Aggregation technique	Number of solutions	MOO approach used	Archive
MORAP	Resource allocation	Single	ND	–	Single	Pareto	No
MOAQ	Network	Multiple	ND	–	Multiple	Pareto	No
Bicriterian ANT	Vehicle routing	Multiple	ND	Weighted product	Multiple	Pareto	No
P-ACO	Portfolio selection problem	Single	BO	Weighted product	Multiple	Pareto	Yes
MACS-VRPTW	Vehicle routing	Multiple	ND	Weighted product	Multiple	Pareto	Yes
MOACS	Vehicle routing	Multiple	ND	–	Multiple	Pareto	Yes
MONACO	Networking	Single	BOW	Weighted product	Single	No	No
COMPET ants	Transportation	Multiple	BO	Weighted sum	Multiple	Yes	No
MOACOM	Scheduling	Multiple	BO	Weighted product	Single	No	No
ACOAMO	Scheduling	Multiple	ND	Weighted product	Single	No	No
MOACO	Load balancing	Multiple	BO	Random	Single	Yes	Yes

$$v_{id}(t+1) = wv_{id}(t) + c_1 rand_1()(p_{id}(t) - x_{id}(t) + c_2 rand_2()(p_{gd}(t) - x_{id}(t)) \tag{10}$$

and,

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \tag{11}$$

The pseudo-code for a basic PSO algorithm for single objective optimization is illustrated below:

```
Begin
Parameter settings and initialization of swarm.
Evaluate fitness and locate the leader (i.e., initialize pbest and gbest).
I=0          // I = Iteration count
While (the stopping criterion is not met, say, I < Imax)
    Do
        For each particle
            Update velocity (flight) and position as per Eqs. 10 and 11.
            Evaluate fitness
            Update pbest
        End For
    Update leader (i.e., gbest)
    I++
End While
End
```

First, the swarm is initialized. This initialization includes both positions and velocities. The corresponding p_{best} of each particle is initialized and the leader is located (the g_{best} solution is selected as the leader). Then, for a maximum number of iterations, each particle flies through the search space updating its position (using equations) and its p_{best} and, finally, the leader is updated too. Based on the nature of the problem (Discrete/Continious), influence of parameters (towards exploration and exploitation), number of swarms, and combination with other heuristic methods, the PSO algorithms are classified as follows:

Discrete/Binary *PSO*: Kennedy and Eberhart [60, 154] immediately after proposing the framework for particle swarm optimization have given an alternative algorithm to operate on discrete binary variables. In this binary version, the trajectories are changes in the probability that a coordinate will take on a zero or one value. The binary PSO can handle the discrete optimization problems. There are different types of Binary PSO available in the literature like PSOLS, DPSO, CPSO, PBPSO (Table 5).

Adaptive PSO: Shi and Eberhart [61] after doing an empirical study over PSO, by using four different benchmark functions suggested a self-adapting strategy for adjusting the inertia weight to overcome the disadvantage that PSO may fail to find the required optima in cases when the problem to be solved is too complicated and complex. There are different types of adoptive PSO's like APSO, MPSO, AsCFPSO, APSO, ACPSO, etc. are reported in the literature (Table 6).

Table 5 List of discrete/binary PSO

Authors	Year	Algorithm	Application area
Lio et al. [182]	2005	Particle swarm optimization using local search (PSO-LS)	Flow shop scheduling
Correa et al. [183]	2006	Discrete PSO (DPSO)	Attribute selection
Wang [184]	2007	Discrete PSO (DPSO)	Knapsack problem
Jarboui et al. [185]	2007	Combinatorial PSO	Resource constraint project scheduling
Zhen et al. [186]	2008	Probability based binary PSO (PBPSO)	WFGD problem

Table 6 List of adoptive PSO

Authors	Year	Algorithm	Technique
Carlsie and Dozier [187]	2000	APSO	Each particle to reset its record of its best position as the environment changes
Zhang and Liu [188]	2005	APSO	Adjusts the parameters automatically, based on the fitness values of particles during the optimization process.
Zhen et al. [189]	2007	Modified PSO (MPSO)	Present a new adaptive mutation particle swarm optimizer, which is based on the variance of the population's fitness
Chunxia and Youhong [190]	2008	Adaptive simple PSO with constriction factor (AsCFPSO)	Used the concept of chaotic map
Zhan et al. [191]	2009	APSO	Used the concept of elitist learning
Hongwu [192]	2009	Adaptive chaotic PSO (ACPSO)	Applies a short term chaotic search to the best particle in the iteration

Multi-swarm PSO: Bergh and Engelbrecht [62], have proposed a variation on the traditional PSO algorithm, called the cooperative particle swarm optimizer, or CPSO, employing cooperative behavior, to significantly improve the performance of the original algorithm. This is achieved by using multiple swarms to optimize the different components of the solution vector cooperatively. Different Multi-swarm PSO are CPSO, Dual PSO, and MSBPSO (Table 7).

Hybrid PSO: This method combines different techniques to make the PSO algorithm more smatter. Some the hybrid PSO algorithms are: DPSO, SCEPSO, PSOwSA, OPSO, NF-PSO, ESCA (Table 8).

Table 7 List of multi-swarm PSO

Authors	Year	Algorithm	Technique
Bergh and Engelbrecht [62]	2004	Cooperative particle swarm optimizer (CPSO)	Potter's technique
Jian et al. [193]	2008	Dual PSO	The stochastic ranking algorithm is employed
Li and Xiao [194]	2008	Multi-swarm and multi-best particle swarm optimization (MSBPSO)	Aggregation

Table 8 List of hybrid PSO

Authors	Year	Algorithm	Basic components	Application area
Ling et al. [195]	2005	DPSO	DPSO + SA	Vehicle routing problem
Pan et al. [196]	2006	SCEPSO & PSOwSA	EA + PSO PSO + SA	Bench mark functions
Tian and Li [198]	2009	NFPSO	FLC + PSO	System performance
Lung and Dumitrescu [199]	2009	ESCA	EA + PSO	Moving peaks benchmark (MPB)
Mousa et al. [200]	2012	Hybrid PSO	LS + GA + PSO	Bench mark functions

The relative simplicity of PSO and the population-based technique as well as the information sharing mechanisms associated with PSO, makes it a natural candidate for solving multiple and many objective problems by redefining the notion of the guide. Let us discuss PSO for multi-objective optimization problem.

The two basic PSO equations restrict additional heuristics related to the real-world problem to be incorporated into the algorithm. Thus, PSO in its basic form will not perform well in searching complex multi-objective solution spaces, which are the case for many complex real world scenarios. Changing a PSO to a MOPSO requires a redefinition of guide, in order to obtain a front of optimal solutions (Pareto front). In MOPSO, the Pareto optimal solutions are used to determine the guide for each particle. In order to apply the PSO strategy for solving multi-objective optimization problems, the original scheme has to be modified. The algorithm needs to search a set of different solutions (the so-called Pareto front) instead of a single solution (as in single objective optimization). We need to apply (MOPSO) to search towards the true Pareto front (non-dominated solutions). Unlike the single objective particle swarm optimization, the algorithm must have a solution pool to store non-dominated solutions found by searching upto stopping criterion (say, upto iteration *Imax*). Any of the solutions in the pool can be used as the *gbest* particle to guide other particles in the swarm during the iterated process. The plot of the objective functions whose non-dominated solutions are in the solution pool would make up for the Pareto front. The basic MOPSO algorithm can be stated as below:

```
Begin
Parameter settings and initialize swarm
Evaluate fitness and initialize leaders in a leader pool or external archive
Archive the top best leader from the external archive through evaluation of some sort
of quality measure for all leaders.
I = 0          // I = Iteration count
While (the stopping criterion is not met, say, I < Imax)
    Do
        For each particle
            Select leader in the external archive
            Update velocity
            Update position
            Mutate periodically /*optional */
            Evaluate fitness
            Update pbest
        End For
    Crowding of the leaders
    Update the top best into the external archive
    I = I + 1.
End While
Report results in the external archive
End
```

In the above general MOPSO algorithm, first the swarm is initialized. Then, a set of leaders is also initialized with the non-dominated particles from the swarm. This set of leaders is stored in an external archive. Later on, some sort of quality measure is calculated for all the leaders in order to select, usually one leader for each particle of the swarm. At each generation, for each particle, a leader is selected and the flight is performed. Most of the existing MOPSOs apply some sort of mutation operator after performing the flight. Then, the particle is evaluated and its corresponding p_{best} is updated. A new particle replaces its pbest particle usually when this particle is dominated or if both are incomparable (i.e., they are both non-dominated with respect to each other). After all the particles have been updated, the set of leaders is updated, too. Finally, the quality measure of the set of leaders is re-calculated. This process is repeated for a certain fixed number of iterations.

Coello and Lechuga [63] extended PSO to deal with multi-objective optimization problems using the similar approach of Pareto dominance to determine the flight direction of a particle and their MOPSO algorithm maintains previously found non-dominated vectors in a global repository (secondary memory) that is later used by other particles to guide their own flight. Their approach is population based as well as geographically based to maintain diversity. Fieldsend and Singh [64] utilize the dominated tree data structure to enable the selection of an appropriate Pareto archive member to act as the global best for any given particle and also maintains a local set of best solutions for each swarm member. They have demonstrated that this approach is significantly better than the method used by Coello and Lechuga [63] and also

PAES derived from the unified model proposed by Laumanns et al. [65]. They have demonstrated that by including a stochastic turbulence variable within MOPSO, its performance has been significantly increased. Even though Coello and Lechuga [63] maintain an archive of global best solutions, Fieldsend and Singh [64] pointed out that there is a better way to select from this archive than by simple density based selection. Thus, they have included a new data structure called dominated tree, as this data structure facilitates rapid selection of an appropriate archive member for their new MOPSO method. Hu and Eberhart [66] attempt to optimize MOPSO having two objectives through the a priori knowledge of the test function properties. Parsopoulos and Vrahatis [67] introduced two methods to optimize MOP having two objectives. One uses a weighted aggregate approach and another is loosely based on Schaffer's MOEA [68] i.e. vector evaluated PSO (VEPSO) method. The second method—the vector evaluated particle swarm optimizer (VEPSO) of Parsopoulos and Vrahatis [67] uses one swarm for each objective. According to them, the best particle of the second swarm is used to determine the velocities of the first swarm (act as its global best), and vice versa. Mostaghim and Teich [69] introduce a new method called sigma method for finding best local guides for each particle of the population from a set of Pareto optimal solutions. According to them, such a technique has a great impact on the convergence and diversity of solutions, especially when optimizing problems with a high number of objectives. The sigma method which uses clustering techniques for fixing the archive size has a better computational time, diversity and covergence than the dominated tree method of Fieldsend and Singh [64] and strength pareto evolutionary algorithm (SPEA). Coello et al. [70] proposed an improved version of the MOPSO algorithm in which they have added a constraint handling mechanism and a mutation operator that considerably improves the exploratory capabilities of their original algorithm. Their MOPSO is validated using several standard test functions reported in the specialized literature. They have compared this improved MOPSO algorithm against three highly competitive evolutionary multi-objective (EMO) algorithms, NSGA-II and PAES. Fieldsend [71] compare a number of selection regimes for the choosing of global best (*gbest*) and personal best (*pbest*) for swarm members in MOPSO. He has shown two distinct *gbest* selection techniques, one that does not restrict the selection of archive members and the other with distance based *gbest* selection techniques. According to him, these two methods promote two types of search. He has also described the potential problem of particle clumping in MOPSO. Ray and Liew [72] propose an algorithm which uses Pareto dominance and combines concepts of evolutionary techniques with the particle swarm. Pulido and Coello [73] use the concept of Pareto dominance to determine the flight direction of a particle. The authors adopt clustering techniques to divide the population of particles into several swarms. This aims to provide a better distribution of solutions in decision variable space. This approach does not use an external archive since elitism in this case is an emergent process derived from the migration of leaders.

Mostaghim and Teich [74] propose particle swarm inspired evolutionary algorithm (PSEA) which is a hybrid between PSO and an evolutionary algorithm. The main aim is to use EA operator (mutation, for example) to emulate the workings of PSO mechanisms. Different methods for selecting and deleting particles (leaders) from the archive are analyzed to generate a satisfactory approximation of the Pareto front. The authors provide some statistical analysis in order to access the impact of each of the parameters used by their approach. Li [76] proposes an approach which incorporates the main mechanisms of the NSGA-II of Deb et al. [77] to the PSO algorithm. In this approach, once a particle has updated its position, instead of comparing the new position only against the *pbest* position of the particle, all the *pbest* positions of the swarm and all the new positions recently obtained are combined in just one set (given a total of 2N solutions, where N is the size of the swarm). Then, the approach selects the best solutions among them to conform the next swarm (by means of a non-dominated sorting). The author doesn't specify which values are assigned to the velocity of pbest positions, in order to consider them as particles. This approach also selects the leaders randomly from the leaders set (stored in an external archive) among the best of them, based on two different mechanisms: a niche count and a nearest neighbor density estimator. This approach uses a mutation operator that is applied at each iteration step only to the particle with the smallest density estimator value (or the largest niche count). Sierra and Coello [78] propose an approach which is based on Pareto dominance and the use of a nearest neighbor density estimator for the selection of leaders (by means of a binary tournament). This proposal uses two external archives: one for storing the leaders currently used for performing the flight and another for storing the final solutions.

Ho et al. [79] propose a novel formula for updating velocity and position of particles, based on three main modifications to the known flight formula. The authors introduce a craziness operator in order to promote diversity within the swarm. The craziness operator is applied (with certain probability) to the velocity vector before updating the position of a particle. Finally, the authors introduce one external archive for each particle and one global external archive for the whole swarm. The archive of each particle stores the latest Pareto solutions found by the particle and the global archive stores the current Pareto optimal set. Every time a particle updates its position, it selects its personal best from its own archive and the global best from the global archive. In both cases, the authors use a roulette selection mechanism based on the fitness values of the particles and on an age variable that the authors introduce and that is increased at each generation. Villalobos-Anias et al. [80] propose a new mechanism to promote diversity in multi-objective optimization problems. Although the approach is independent of the search engine adopted, they incorporate it into the MOPSO proposed in Coello et al. [70]. Salazar-Lechuga and Rowe [81] propose an approach whose main idea is to use PSO to guide the search with the help of niche counts to spread the particles along the Pareto front. The approach uses an external archive to store the best particles (non- dominated particles) found by the algorithm. Since this external archive helps to guide the search, the niche count is calculated for each of the particles in the archive and the leaders are chosen from this set by means of a stochastic sampling method (roulette wheel). Also, the niche count is used as a criterion to update the external archive. Each time the archive is full and a new particle wants to set in, its niche count is compared with the niche count of the worst solution of the archive. If the new particle is better than the worst particle, then the new particle enters into the archive and the worst particle is deleted. Niche counts are updated when inserting or deleting a particle from the archive. Janson and Merkle [82] proposed a hybrid particle swarm optimization algorithm for multi objective optimization, called ClustMPSO. ClustMPSO combines the PSO algorithm with clustering techniques to divide all particles into several sub-swarms. For this aim, the authors use the K-means algorithm. Each sub-swarm has its own non-dominated front and the total non-dominated front is obtained from the union of the front of all the sub-swarms. Lewis [83] has proposed a novel MOPSO algorithm called LoCost algorithm through some modification in the velocity updating equation of the conventional PSO algorithm based on an extension of the concepts of spatial social networks using a model of the behavior of particular types of swarms known as crickets and locusts. He observes that the proposed algorithm has performed quite comparably to a conventional MOPSO algorithm in terms of convergence, and has achieved appreciably greater coverage of the approximation to the Pareto-front. Leong and Yen [84] present the improvement of two design components (swarm growing strategy and objective space compression and expansion strategies) from the existing multiple swarm MOPSO, namely dynamic swarm in multi-objective particle swarm optimization (DSMOPSO). The multiple-swarm concept has been incorporated into PSO to yield more efficient and effective designs, especially in enhancing the population diversity, and to counter PSO's tendency in undesirable premature convergence. Lewis and Ireland [85] have proposed an approach of hybridizing a multi-objective optimization method and subsequent single-objective search has been proposed as a means to automate the process of solution selection from the set of Pareto-optimal solutions typically delivered.

Cagnina et al. [86] have proposed a hybrid particle swarm approach called simple multi-objective particle swarm optimizer (SMOPSO) which incorporates Pareto dominance, an elitist policy, and two techniques to maintain diversity: a mutation operator and a grid which is used as a geographical location over objective function space. Laura and Mihai [87] have proposed a hybrid technique that combines a genetic algorithm (GA) and a PSO algorithm. Each GA chromosome is an array encoding a meaning for updating the particles of the PSO algorithm. The evolved PSO algorithm is compared to a human-designed PSO algorithm by using ten artificially constructed functions and one real-world problem. The model proposed in this paper is divided into two levels: a macro level and a micro level. The macro level is a GA algorithm that evolves the structure of a PSO algorithm. For this purpose, a particular function is used as a training problem. The micro level is a PSO algorithm used for computing the quality of a GA chromosome from the macro level. The array of integers encoded into a GA chromosome represents the order of update for particles used by a PSO algorithm that solves a particular problem. Goldberg et al. [88] present a particle swarm optimization algorithm for the multicriteria constrained minimum spanning tree problem. The operators for the particle's velocity are based upon local search and path-relinking approaches. In path-relinking approach, a velocity operator is developed and utilized when a particle goes toward the position of another particle. For the iterations where a particle follows its own way, a local search procedure is used. Ho et al. [89] proposes a novel intelligent multi-objective particle swarm optimization (IMOPSO) to solve multi-objective optimization problems.

Koppen and Veenhuis [90] introduce a new approach to multi-objective particle swarm optimization. The approach is based on the recently proposed Fuzzy-Pareto-Dominance (FPD) relation. Chiu et al. [91] present a local guide assignment strategy for MOPSO called cross searching strategy (CSS) which will distribute suitable local guides for particles to lead them toward the Pareto front and also keeping a diversity of solutions. A disturbance operation is also introduced to enhance the particle's searching ability to avoid local search. Peng and Zhang [92] have studied the application of PSO techniques to multiobjective optimization using decomposition methods. A new decomposition-based multi-objective PSO algorithm is proposed, called MOPSOID. It integrates PSO into a multi-obejective evolutionary algorithm based on decomposition (MOEAID). Like MOEAlD, each particle in MOPSOID carries one unique weight vector. Therefore, each particle has an unique search direction defined by its weight vector. Padhye et al. [93], have reviewed the several proposals for guide selection in multiobjective particle swarm optimization (MOPSO) and compare them with each other in terms of convergence, diversity and computational times. The new proposals made for guide selection, both *pbest* and *gbest*, are found to be extremely effective and perform well compared to the already existing methods. The combination of various selection methods is also studied and it turns out that there exist certain combinations which yield an overall superior performance outperforming the others. Cabrera and Coello [94] present a multi-objective particle swarm optimizer (MOPSO) which is characterized for using a very small population size so that it requires a very low number of objective function evaluations (only 3000 per run) to produce reasonably good approximations of the Pareto front of problems of moderate dimensionality.

The proposed approach first selects the leader and then selects the neighborhood for integrating the swarm. The leader selection scheme adopted is based on Pareto dominance and uses a neighbors' density estimator. Additionally, the proposed approach performs a re-initialization process for preserving diversity and uses two external archives: one for storing the solutions that the algorithm finds during the search process and another for storing the final solutions obtained. Furthermore, a mutation operator is incorporated to improve the exploratory capabilities of the algorithm. Wang and Yang [95] extend the NSGA-II-MOPSO algorithm, which is based on the combination of NSGA-II and multi-objective particle swarm optimizer (MOPSO) for unconstrained multi-objective optimization problems, to accommodate constraints and mixed variables. In order to utilize the valuable information from the objective function values of infeasible solutions, a method called M+1 non-dominated sorting is proposed to check the non-domination levels of all infeasible solutions. Integer and discrete variables are dealt with using a method called stochastic approximation.

Goh et al. [96] propose a competitive and cooperative co-evolutionary approach to be adapted for multi-objective particle swarm optimization algorithm design. It appears to have considerable potential for solving complex optimization problems by explicitly modeling the co-evolution of competing and cooperating species. The competitive and cooperative co-evolution model helps to produce the reasonable problem decompositions by exploiting any correlation and interdependency among the components of the problem. Each sub-swarm is assigned a probability of representing a particular variable and only two sub-swarms, the current sub-swarm and competing sub-swarm compete for the right to represent any variable at any one time. Tsai et al. [97] propose an improved multi-objective particle swarm optimizer with proportional distribution and jump improved operation, named PDJI-MOPSO, for dealing with multiobjective problems. PDJI-MOPSO maintains diversity of new found non dominated solutions via proportional distribution, and combines the advantages of wide ranged exploration and extensive exploitation of PSO in the external repository with the jump improved operation to enhance the solution searching abilities of particles.

Zheng and Liu [98] propose a hybrid vertical mutation and self-adaptation based MOPSO (VMAPSO) to overcome the disadvantages of existing MOPSOs. Wang and Yang [99] use a new optimality criterion based on preference order (PO) scheme to identify the best compromise in multi-objective particle swarm optimization (MOPSO). Preference order is a generalization of Pareto optimality. It provides a way to designate some Pareto solutions superior to others when the size of the non-dominated solutions set is very large. To find the “best compromise”, the non-dominated solutions are ranked according to PO. The ranking procedure can be summarized in three steps: (i) identify the combinations of all subsets to m objectives; (ii) assign the order to all non-dominated solutions for each combination of all subsets based on PO; and (iii) identify the “best compromise” in all non-dominated solutions according to their order. The proposed algorithm is quite effective in maintaining the diversity of the solutions.

2.3 ABC for Multiple Objective Problem

ABC is one of the swarm based meta-heuristic algorithm introduced by Karaboga in 2005. It is based on the model proposed by Tereshko and Loengarov [100], Mishra et al. [14]. It is motivated by the intelligent behavior of honey bees. Honey bees are one of the interesting swarm in nature. They have the skills like photographic memories, space-age sensory, and navigation systems. Honey bees are social insects who live in colonies. There are three kinds of bees in a colony: queen, drones, and workers. Queen bee has the largest living years. She is the only egg laying female who is the mother of all the members of the colony. When the colony is lack of food sources, the queen produces new eggs. If the colony becomes too crowded, the queen stops laying. Drones are the father of the colony. They are produced from the unfertilized eggs; their life span is about six months. The main task of the drones is to fertilize with the queen. Workers have the task of collecting food, storing it, removing dead bees, ventilate the hive and give protection to the hive. The task of a worker bee is based on its age and the needs of the colony (Table 9).

At the initial stage the algorithm was used for numerical optimization, but later on it is widely used for combinatorial optimization and also for constraint and unconstrained function optimization problems. In the ABC location of food source represents a possible solution to the problem and the nectar amount of a food source

Table 9 Application of MOPSO on different discipline

Author	Number of objectives	Problem solved
Chauhan et al. [101]	2	Power system
Falcon et al. [102]	2	Clustering
de Carvalho et al. [103]	3	Fault prediction
Martin et al. [104]	2	Ultra wide band
Pang and Chen [105]	2	Signal processing
Hazra and Sinha [106]	2	Congestion management in power system
Qasem and Shamsuddin [107]	2	Medical
Pindoriya et al. [108]	2	Portfolio management
Sha and Lin [109]	2	Job scheduling
Wang and Singh [110]	2	Power sector
Montalvo et al. [111]	2	Water distribution
Liu [112]	2	Calibaration of rain fall
Zhang and Liu [113]	2	Power system
Cai et al. [114]	2	Environment & Economics
Ganguly et al. [115]	2	Electrical distribution system
Sankaran and Manne [116]	2	Channel equalization

corresponds to the quality of the solution. A general framework of ABC is presented below.

Framework of ABC:

Initialize the population of solution x_{ij}.
Evaluate the population
Cycle = 1
Repeat

Produce new solutions (food source positions) v_{ij}, in the neighborhood of x_{ij} for the employed bees $v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj})$ and evaluate

Apply the greedy selection process between x_i and v_i

Calculate probability values P_i for the solutions x_i by means of their fitness value as

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i}$$

Calculate the fitness value of solutions

$$fit_i = \begin{cases} \frac{1}{1+f_i} & \text{if } f_i \geq 0 \\ 1 + abs(f_i) & \text{if } f_i < 0 \end{cases}$$

Table 10 Control parameter in ABC

Parameter	Meaning
Population size	Number of solutions
Termination condition	Maximum number of cycles/maximum CPU time/if the onlooker bees converge faster
Number of employed bees	Number of food sources in the hive

Normalize P_i values into [0, 1]
Produce the new solutions v_i for the onlookers from the solutions x_i selected depending on P_i and evaluate them
Apply the greedy selection process for the onlookers between x_i and v_i
Determine the abandoned solution, if exists and replace it with a new randomly produced solution x_i for the scout using the equation

$$x_{ij} = min_j + rand(0, 1) * (max_j - min_j)$$

Memorize the best food source position (solution) achieved so far
Cycle = cycle + 1
Until cycle = Maximum Cycle Number (MCN)

The ABC algorithm is simple, easy to implement, has few control parameters and mostly robust in nature so, it is widely used for a single objective optimization problem. Due to the aforesaid qualities of ABC, like other multi-objective heuristic algorithms, it can be extended to multi and many objective domains [14, 129]. Let us discuss some of the variants of MOABC (Tables 10 and 11).

In the last few years ABC is being used widely for solving multi criterion problems. It has been the point of attraction for the researchers due to its characteristics like use of less number of control parameter and its simplicity. The basic MOABC algorithm can be stated as below:

The basic MOABC algorithm can be stated as below:

1. Cycle = 1
2. Initialize the food source positions (solutions) x_i, $i = 1, \ldots, SN$
3. Evaluate the nectar amount (fitness fit_i) of food sources
4. The initialized solutions are sorted based on nondomination
5. Store nondominated solutions in the external archive (EA)
6. Repeat
7. Onlooker Bees' Phase
    For each onlooker bee
        Randomly chooses a solution from EA
        Produce new solution vi by using comprehensive learning strategy
        Calculate the value fiti

Table 11 Popular ABC algorithms

Author	Year	Application area
Karaboga [117]	2005	Numerical optimization
Teodorovic and Dell [118]	2005	Ride matching problem/transportation problem
Wedde and Farooq [119]	2005	Mobile adhoc N/W
Drias and Yahi [120]	2005	MAX-W-SAT problem
Chong et al. [121]	2006	Job shop scheduling
Quijano and Passino [122]	2007	Resource allocation problem
Karaboga and Akay [123]	2009	Engineering design problem
Karaboga and Ozturk [124]	2009	Neural network training
Karaboga [125]	2009	Image processing
Karaboga and Ozturk [126]	2011	Data clustering
Xu and Duan [127]	2010	Image processing
Yu et al. [128]	2013	Test functions

Apply greedy selection mechanism to decide which solution enters EA
End For
8. The solutions in the EA are sorted based on nondomination
9. Keep the nondomination solutions of them staying in the EA
10. If the number of nondominated solutions exceeds the allocated size of the EA, then use crowding distance to remove the crowded members
11. Cycle = cycle + 1.
12. Until (cycle = Maximum Cycle Number)

In 2011 Zou et al. [130] proposed a Multi-objective Artificial Bee Colony (MOABC) which allows the ABC algorithm to deal with multi-objective optimization problems. It is based on non-dominated sorting strategy [131] and used the concept of Pareto dominance [132, 133] to determine which solution vector is better. In addition, it uses an external archive to maintain non-dominated solution vectors. Akbari et al. [133] proposed a MOABC (encouraged from their earlier work [133, 134]), which utilizes different types of bees (i.e., employed bees, onlookers, and scouts) and a fixed-sized archive to maintain the good solutions. To maintain the archive they have used an ε-dominance method. The MOABC method is constituted of five parts: Initialization, Send Employed Bees, Send Onlooker Bees, Send Scout Bees and Update the archive. They tested MOABC over unconstrained and constraint based test functions and concluded that MOABC successfully solves the functions and obtains first rank among all other optimization algorithms (Table 12).

Akbari et al. [133] concluded that the MOABC can obtain better performance by increasing the number of individuals in the population and the number of iterations. The effectiveness of the MOABC depends on two factors: a population of different bee types with an efficient decision making process, and a grid for controlling the diversity over the external archive. In 2011, Omkar et al. [129] employed the concept of Vector Evaluated Artificial Bee Colony (VEABC), a variant of the classic ABC for multi-objective design optimization of composite structure, which proves to be very

Table 12 Enlist of MOABC algorithms

Name of MOABC	Authors and year	Technique	Application domain	Number of objectives
MOABC	Zou et al. 2011	Pareto based	Test functions	2
MOABC	Akbari et al. 2012	Pareto based, grid approach	Test functions	2
VEABC	Omkar et al. 2011	Pareto based	Composite structure	2
EPABC	Wang et al. 2012	Pareto based	Job shop scheduling	3
BCMO	Xinyi et al. 2012	Pareto based	Test functions	2
A-MOABC/PD	Bahriye 2013	Non Pareto	Test function	2
A-MOABC/NS	Bahriye 2013	Non Pareto	Test function	2
S-MOABC/NS	Bahriye 2013	Non Pareto	Test function	2
ICABCMOA	Zhou et al. 2013	Pareto based	Test function	2

appropriate for structural problems. In 2012, Xinyl et al. [135] proposed an artificial Bee Colony algorithm for Multi-objective Optimization problems (BCMO) by introducing the concept of Pareto sorting operation and Pareto dominance approach. By using the effective decoding scheme, hybrid initialization strategy and by using the exploration and exploitation ability of ABC Zhou et al. [136] proposed an MOABC for handling Multi-objective job shop scheduling problem. Wang et al. [137] proposed an enhanced Pareto based artificial bee colony algorithm EPABC for solving multi-objective flexible job shop scheduling problem. Based upon synchronous and asynchronous models using Pareto dominance and non dominated sorting Akay [138] proposed three MOABC named as asynchronous multi-objective ABC using only Pareto dominance rule (A-MOABC/PD), asynchronous multi-objective ABC using non-dominated sorting procedure (A-MOABC/NS) and synchronous multi-objective ABC using non- dominated sorting procedure (S-MOABC/NS). By considering several parameters like inverted general distance, spread performance metrics and running time, etc. and came to a conclusion that S-MOABC/NS is more scalable and efficient in comparison to other two algorithms. To increase the search efficiency in ABC Zhou et al. [139] proposed Immune based Chaotic Artificial Bee Colony Multi-objective Optimization Algorithm (ICABCMOA). In this approach in order to meet the requirements of Pareto based approaches they defines a new fitness based function based on the dominated number. They have used a high dimension chaotic method based on tent map to increase the search efficiency.

From the above discussion, it is derived that the MOABC with properties such as an effective trajectory adjusting strategy, and an efficient way for maintaining the diversity over the Pareto front, can be used as an alternate way for optimizing multiple and many objective problems.

3 Swarm Intelligence for Many Objective Optimization

Recall that population based swarm intelligence techniques are one of the active research areas in the field of multi-objective problems. However, their search ability severely decreases when it is mapped to a many objective problem. For example, the swarm intelligence techniques like ACO, PSO, and ABC, which uses a Pareto dominance approach faces a number of difficulties like:

i. Decrease in convergence property: As the number of objective increases most of the solutions in the population become non-dominated, this decreases the Pareto dominance based selection pressure towards the Pareto front.
ii. As the number of objective increases, thousands of non-dominated solutions are generated, a subset of the nondominated solutions has to be selected to be approximated to entire Pareto front, which is a tedious task.
iii. In a multi-objective problem few number of non-dominated solutions appear on the Pareto front, so the user/the decision maker can choose a solution by visualizing the results. But in case of a many objective as the number of non-dominated solutions are more so it becomes difficult for the user/decision maker to visualize the results and to take a decision.

The problems raised by the swarm intelligent algorithms in a many objective field can be handled by using any of the following techniques.

i. By changing the standard Pareto dominance Method: Rather than using the common Pareto dominance method, the angle of dominating region should be adjusted with the number of objectives, i.e., increase in number of objectives requires a large angle of dominated region. So, that number of non-dominated solutions in each population are decreased and the selection pressure towards the Pareto front can be strengthened.
ii. By ranking the non-dominated solutions: By using favour relation proposed by Drechsler et al. [140] the non-dominated solutions can be ranked. The relation is based on a number of objectives.
iii. By using the indicator function [141]: A number of performance indicators have been proposed to measure the quality of the non-dominated solution sets. They can be applied over to find out the solutions. For example one of the performance indicator used is the hyper volume indicator. Hyper volume indicator measures convergence to the Pareto front and diversity of the obtained fronts. The hyper volume computes the volume in the objective function space covered by members of a non dominated set of solution ND. For each non-dominated solution of Q a hyper cube v_i is constructed, with a reference point 'w' and the solution 'i' as the diagonal corners of the hypercube. The reference point 'w' can be found by constructing a vector with the worst objective function values. After that the union of all hyper cubes are found and then its hyper volume 'HV' is calculated as:

$$\mathrm{HV} = \mathrm{Volume}(U_{i=1}^{n} v_i) \tag{12}$$

Higher values of the hyper volume performance measure imply more desirable solution.

iv. Use of scalarizing function [142]: In this technique weighted sum of multiple objectives are calculated even though the number of objectives is large. There are many scalarizing functions available in the literature like weighted sum, reference vector etc. to reduce number of objectives. For example: Let us consider F(z) is a function which is to be maximized number of objectives are 'n' number. F(z) is the 'n' dimensional objective vector, & Z is the decision vector. So,

$$Maximize\, F(z) = \{F_1(Z), F_2(Z), \ldots, F_n(Z)\}$$

By applying frequently used weighted sum scalarizing function $w = \{w_1, w_2, \ldots, w_n\}$

$$fitness(z) = w_1 * F_1(z) + w_2 * F_2(z) + \cdots + w_n * F_n(z)$$

This is how the weighted sum approach works. The weight vector is a user input or decision makers input. Distance from the reference vector can be used as a scalarizing function. When a reference vector $RV = \{RV_1, RV_2, \ldots, RV_n\}$ is given as a desired point in the objective space, the distance from RV can be calculated as a scalaring function.

$$fitness(z) = distance(RV, F(z))$$

v. Use of preference Information: SI algorithms generally designed to search for a set of non-dominated solutions that approximates the Pareto front. As the number of solutions for a good approximation exponentially increases with the number of objectives, one can focus on a specific region of the Pareto front using decision maker's preference.

General Framework for solving a Many-objective Problem using SI

Step 1 Parameter setting of MOACO/MOPSO/MOABC.
Step 2 Generate non-dominated solution by MOACO/MOPSO/MOABC.
Step 3 Collect preference/weight information from the decision maker.
Step 4 Rank non-dominated solution using indicator based approach R2 analysis/TOPSIS
Step 5 Get the highest rank solution as the selected one.
Step 6 If the solution is satisfactory stop else go to Step 4.

There are a number of problems available which can be addressed as a many-objective problem. We are enlisting few of them, which will help the researcher to test their designed algorithm over many objective problems (Table 13).

Table 13 List of many objective test problem

Problems	Objectives
Knapsack	4, 5
Heuristic learning	7, 8
Nurse scheduling	25
DTLZ	8, 15
TSP	5, 10, 15, 20
Job shop scheduling	5, 10, 15, 20
DTLZ	10, 15, 20
Flight control system	8
Modified DTLZ	30, 5, 7, 9
Supersonic wing design	4
UF1, UF2, UF3	5
Inverted DTLZ1	5
C2-DTLZ2	4
Crash-worthiness in design of vehicle	4
Car side impact problem	4

4 Study of Swarm Intelligence for EEG Signal

Brain Computer Interface is a system that connects a human brain and machine. The main objective is to establish a new augmentative communication system that translates human intensions reflected by suitable brain signals [143]. Therefore, it is anticipated that the BCI system will become a technology not only for the average persons but also for the disabled persons. All types of activities start in a human being with the help of neurons, which is the basic element of the neural system of the human being. The actions or signals generated by the brain are electrical in nature and represent not only the brain function, but also the status of the whole body. Commonly three methods are used to know the changes within the brain with high temporal resolution of neuron interactions at the network level. They are EEG (Electroencephalography), MEG (Magneto encephalography), and fMRI (functional Magnetic Resonance Imaging) [144, 145]. Among them EEG signals are widely used and is the focus of this section/article. The EEG signal processing starts with acquisition of the signal and ends with a post processing task.

EEG signals of brain are very non-linear, random in nature, and dynamic. Human brain consists of number of cells they communicate with each other with the help of electrical impulses. The impulse can be measured by placing the electrode on the scalp of the agent. The EEG signals are generated through the cortical nerve cell inhibitory and excitatory post synaptic potential. These postsynaptic potential summate in the cortex and extend to the scalp surface where they are recorded as EEG. A typical EEG signal has the amplitude of about 10–100 v and the frequency in the range of 1 Hz to about 100 Hz. During recording of the signal, their may be lots of chances that, different types of artifacts get added to the signal.

EEG signal can be recorded in two different ways. It can be recorded by finding the voltage difference between an active electrode on the scalp and a reference electrode on the ear lobe. In the other way the recording can be done by measuring the voltage difference between two scalp electrodes. The first method is known as the Monopolar whereas the latter is known as the Bipolar technique. EEG signals are associated with several descriptors like: (i) frequency or wave length, (ii) voltage, (iii) waveform, (iv) regulation, (v) manner of occurrence, (serial, continuous, random), (vi) locus, (vii) reactivity, and (viii) inter hemispheric coherence (Symmetry, Synchrony). Any change in the EEG signal pattern leads to abnormality [146, 147]. There are several methods used for measuring the dissimilarity among the patterns they are autocorrelation, high order statics, spectral error, and auto regressive modeling. EEG signals are characterized by delta waves: its frequency of oscillation is 1–4 Hz. It is most prominent at deep stage sleep, this wave characterize depth of sleep. Theta waves: its frequency of oscillation is 4–7 Hz. It is most prominent at dreaming sleep, this wave characterize drowsiness and sleep. Alpha waves: its frequency of oscillation is 8–15 Hz. It is most prominent at relaxation with closed eyes and it is characterized when the body is physically at rest. Beta waves: its frequency of oscillation is 25–100 HZ. This is most prominent at normal waking consciousness. There are various events which affect the EEG signals like sleep, epilepsy, reflexology, drug/anesthesia, diabetes, meditation, music and artifacts.

In a nutshell, while doing any type of analysis starting from pre to post processing operation in an EEG signal it becomes tedious to handle such a real life complex and huge amount of data [148]. On the other hand, it is also difficult for the analyst by using the non-population, non-parallel, and non-heuristic methods to handle such a situation. Hence, in this article, we turn our attention towards swarm based algorithm and its effectiveness in the domain of EEG analysis. Additionally, we discuss the implicit and explicit connections of multi-objective optimization in EEG analysis. The subsections are organized as follows: Sects. 4.1, 4.2, and 4.3 presents the application of ACO, PSO, and ABC for EEG signal analysis. In Sect. 4.4, we discuss the connectivity of multi and or many-objective optimization in EEG signal analysis (Fig. 4).

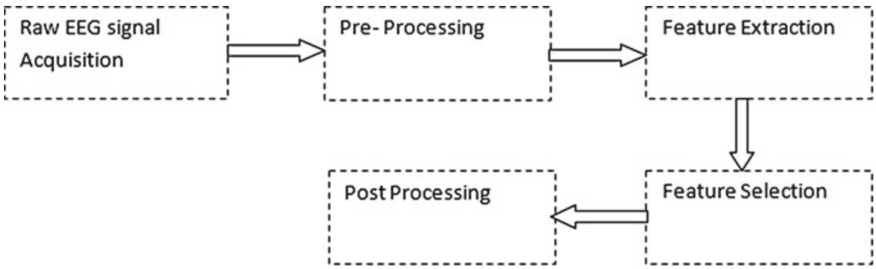

Fig. 4 Block diagram of EEG signal processing

4.1 ACO in EEG Signal Analysis

Bursa and Lhotska [150] and Dries and Peterson [149], have addressed the issue of ant-inspired clustering in the process of long-term electrocardiogram and electroencephalogram processing. They developed an ACO_DTree method which is based on the auto-catalytic collective behavior of real insect colonies. They have used their proposed method in EEG and compared it with WEKA Random Tree and found that their method provides more robust and improve performance. Bursa and Lhotska [150] also addressed automated classification of newborn sleep electroencephalogram using inductive classification methods. They have performed automated classification through ant colony approach (ACO_DTree algorithm) and the Group of Adaptive Models Evolution inductive models. Khushaba et al. [151] proposed a new feature extraction method which utilizes ant colony optimization in the selection of wavelet packet transform (WPT) and adopted in classifying bio-medical signals. The new algorithm named as intelligent artificial ants (AAI), which searches the wavelet packet tree for subsets of features that best interact together to produce high classification accuracies. The AAI method is the mixture of filter and wrapper approaches in the feature subset selection. The significance of the subsets selected by the ants is measured using linear discriminant analysis (LDA) classifier. The proposed method AAI is then tested on bio-signal driven applications, which is the brain computer interface (BCI) problem with 56 EEG channels. Results show that the proposed method achieves a maximum accuracy of 83 %. Again, Khushaba et al. [151] investigated the use of a combination of ACO and differential evolution called ANTDE for feature selection. They compare ANTDE, GA and BPSO, and reported that ANTDE's outperforms GA and BPSO due to the use of a mutual information based heuristic measure. In 2008, Khushaba et al. [152] have reported the application of clustering method inspired by the behavior of real ants in the nature in bio-medical signal processing. The ants cooperatively maintain and evolve a pheromone matrix which is used to select features. Their main aim was to design and develop a combination of feature extraction and classification methods, for automatic recognition of significant structure in biological signal recordings. The method is targeted towards speeding up and increasing objectivity of identification of important classes and may be used for online classification. The method can also be used in expert classification process. They have obtained significant results in EEG signals. Bursa and Lhotska [153] have presented a paper that describe the improved ant colony algorithm. They used the improved ant colony algorithm in emotion clustering of EEG signal to raise the efficiency of the image retrieval. Result shows that the improved algorithm has better clustering and accuracy rate.

4.2 PSO in EEG Signal Analysis

Qiu et al. in 2005 [155] did the feasibility study of EEG dipole source localization using PSO. Dipoles are widely used for approximating the sources of electrical activity in our brain. They showed that PSO is much more efficient than other evolutionary algorithms for EEG Dipole source localization. Qiu et al. [155] proposed an algorithm for EEG classification using radial basis PSO neural network for brain machine interfaces in 2007. In this, they proposed a mental task classification algorithm using PSO for a radial basis neural network. Features, were extracted from EEG signals that were recorded during five mental tasks like resting, mathematical multiplication, geometric figure rotation, letter composing and visual counting. In 2010 Nakamura et al. [157] and Pulraj et al. [156] proposed a method for evaluating the degree of human's preference based on EEG analysis. It is said that sense of touch is an important factor to decide what we like. They proposed a method for extracting the information on the sense of touch based on EEG analysis. They analyzed the EEG signals at the time of touching objects. They often used the frequency analysis for data analysis of the EEG. They applied PSO for selection of the significant component. For separating, support vector machine (SVM) was also used. Their proposed method has the capability of separating human's preference. Alp et al. [158] in 2009 proposed a PSO based technique on Dipole source reconstruction of brain signals. Their proposed method uses PSO for optimally choosing the dipole parameters. Simulation on synthetic data sets showed that their proposed method localized the dipoles into the actual location as well. In the real data sets, the actual dipole parameters are unknown. Due to this the fit error between the measured data and the reconstructed data is minimized. It is observed that their method reduces this error to the noise level by localizing only a few dipoles in the brain. In 2009, Satti et al. [159] worked on Spatio-spectral & temporal parameter searching using class correlation analysis and PSO for BCI. Distinct features play a vital role in enabling a computer to associate different EEG signals to different brain states. To ease the workload on the feature extractor and enhance separability between different brain states, numerous parameters, such as separable frequency bands, data acquisition channels and time point of maximum separability are chosen explicitly to each subject. Earlier research had shown that using subject specific parameters for the extraction of invariant characteristics specific to each brain state can significantly improve the performance and accuracy of a BCI. They developed a fast autonomous user-specific tuned BCI system using PSO to search for an optimal parameter combination based on the analysis of the correlation between different classes i.e. the R-Squared (R2) correlation coefficient rather than assessing overall system performance via performance measure such as classification accuracy. Lin and Hsieh [160] in 2009 proposed a neural classifier based on improved particle swarm optimization (IPSO) to classify an EEG of mental tasks for left-hand movement imagination, right-hand movement imagination, and word generation. First, the EEG patterns utilize principle component analysis (PCA) in order to reduce the feature dimensions. Then a three-layer neural network trained using PSO is used to realize a classifier.

The proposed IPSO method consists of the modified evolutionary direction operator (MEDO) and the traditional PSO. Their proposed MEDO combines the evolutionary direction operator (EDO) and the migration. The MEDO can strengthen the searching global solution. The IPSO algorithm can prevent premature convergence and outperform the other existing methods. Nasser Omer Sahel Ba-Karait et al. [161] proposed detection system of epileptic seizure in EEG signals which is based on Discrete Wavelet Transform (DWT) and Swarm Negative Selection (SNS) algorithm. DWT was used to analyze EEG signals at different frequency bands and statistics over the set of the wavelet coefficients were calculated to introduce the feature vector for SNS classifier. The SNS classification model uses negative selection and PSO algorithms to form a set of memory Artificial Lymphocytes (ALCs) that have the ability to distinguish between normal and epileptic EEG patterns. Thus, adapted negative selection is employed to create a set of self-tolerance ALCs. Whereas, PSO is used to evolve these ALCs away from self patterns towards non-self space and to maintain diversity and generality among the ALCs. The technique was approved to be robust and effective in detecting and localizing epileptic seizure in EEG recording. Wei and Wang [162] used Binary Multi-Objective Particle Swarm Optimization for Channel Selection in Motor Imagery Based Brain-Computer Interfaces in 2011. With the increase of channel numbers, multi-channel EEG signals need inconvenient recording preparation and complex calculation, this is time-consuming and lead to lower classification accuracy. To address this problem, they proposed a novel method, named binary multi-objective particle swarm optimization (BMOPSO) for channel reduction. In 2011, zbeyaz et al. [163] used PSO for Regularization and Kernel Parameters Optimization in EEG Signals Classification with SVM. In this study, firstly power spectrum was obtained by applying Auto-Regressive Burg (AR-Burg) method to the EEG signals. The data obtained from the analysis of AR-Burg was classified with Support Vector Machines (SVM). Classification achievements were investigated for some kernel functions used in SVM. Regularization parameter and kernel parameter that increase the success of classification were calculated with a novel approach of PSO, such that, global best results for classification were searched by investigating optimum values. As a result of this study a new algorithmic approach presented for the diagnosis of epilepsy patients. Kim et al. [164] proposed "A Binary PSO-Based Optimal EEG Channel Selection Method for a Motor Imagery Based BCI System". Brain-computer interface based on motor imagery is a system that transforms a subject's intention into a control signal by classifying EEG signals obtained from the imagination of movement of a subject's limbs. Using many channels cause other problems. When applying a common spatial pattern (CSP), which is an EEG extraction method, many channels cause an over fitting problem, in addition there is difficult using this technique for medical analysis. To overcome these problems, they suggested a PSO applied to CSP. In 2012, Arslan et al. [165] have used a hybrid Structure of ANN and PSO for EEG Signals Classification. ANN and PSO techniques designed in the form of a hybrid structure are used for diagnosis of epilepsy patients via EEG signals. Attributes of EEG signals are needed to be determined by employing EEG signals which are recorded using EEG. From this data, four characteristics are extracted for the classification process. 20 % of available data is reserved for testing while 80 % of available data is being reserved for training. These actions were repeated five times by performing cross-validation process. PSO is used for updating the weights during training ANN and a program is constituted for classification of EEG signals. Atyabi et al. [166] proposed on Adapting Subject-Independent Task-Specific EEG Feature Masks using PSO in 2012. It is reported that dimension reduction is an important step toward asynchronous EEG based BCI systems, with EA based Feature/Electrode Reduction (FR/ER) methods showed significant potential for this purpose. A PSO based approach can reduce 99 % of the EEG data in this manner while demonstrating generalizability through the use of 3 new subsets of features/electrodes that were selected based on the best performing sub-set on the validation set, the best performing sub-set on the testing set, and the most commonly used features/electrodes in the swarm. Their study focused on applying the subsets generated from 4 subjects on a 5th one. Two schemes for this are implemented based on (i) extracting separate subsets of feature/electrodes for each subject (out of 4 subjects) and combining the final products together for use with the 5th subject, and (ii) concatenating the pre processed EEG data of 4 subjects together and extracting the desired subset with PSO for use with the 5th subject. In 2013, Shirvany et al. [167], have proposed a method for solving an inverse problem EEG-based source localization. To determine the location of the brain sources that are responsible for the measured potentials at the scalp electrodes. They proposed a new global optimization method based on PSO to solve the epileptic spike EEG source localization inverse problem. In a forward problem a modified subtraction method is proposed to reduce the computational time.

4.3 ABC in EEG Signal Analysis

In 2013, Ahirwal et al. [148] have used ABC to construct Adaptive Noise Canceller (ANC) for EEG filtering with modified range selection, described as bounded range ABC. They have also implemented ANC with RLS and LMS. They have performed the comparative study of conventional methods like LMS, RLS with ABC, and they found that ABC performs well than conventional methods. They also proposed a new form of controlled search space to stabilize the randomness of swarm intelligence, especially for the EEG signal. The proposed controlled search space technique was tested on each of the swarm intelligence techniques and found to be more accurate and powerful.

In 2013, Rakshit et al. [168] have used EEG based BCI to decode the various movements related data generated from the motor areas of the brain. One the issues in BCI research is the presence of redundant data in the features of a given data set, they have used an ABC cluster algorithm to reduce the features, and acquired their corresponding values. The result shows that it has the highest accuracy of 64.29 % and it also reduced the problem feature. From this study, we have concluded that there are lots of tasks which can be carried out in future by using ABC and its variants.

4.4 Towards Multiple and Many Objectives of EEG Signal

Now-a-days multiple and many objective optimization has been applied in many fields of science ranging from health to engineering sciences, where optimal decisions need to be taken in the presence of trade-offs between three or more objectives. In this section, our main concerned is to revealing conflicting objectives if any during the EEG signal analysis.

Recall that the non-invasive BCI uses EEG signals to capture the brain signal associated with predefined mental tasks. The number of channels used by an EEG system can vary according to the experiment held and the hardware design. It usually ranges between 16 and 256 channels. According to Event Related Desynchronization/Synchronization (ERD/ERS) research, motor imagery experiments can use only the channels at the contralateral hemispheres, which can be as few as 3–5 channels. Using a lot of channels for recording can be useful for medical and diagnostic purposes. For BCI system and especially when building online system, the number of channels should be as minimum as possible. In order to avoid a large number of channels one can choose several electrode positions that are known from neuroscience and psychology studies. Although this approach can be very useful, it ignores the fact that different subjects respond differently and the optimal positioning of the electrodes may vary. The other way around this problem is to use a large number of channels and use some methods to reduce the dimensionality of the input features or to select the best set of channels for each subject. Hence, this problem can be realized as a multi-objective optimization problems by minimizing the number of channels and maximize the classification accuracy. Sleep disorders, epilepsy identification problem, etc. can also be viewed as a multi-objective problem.

In the normal adult, there are two stages of sleep that alternate at about 90-min intervals. Rapid eye movement sleep can be described as a period when the brain is active and the body is paralyzed (except for eye movements, middle ear ossicles, and respiration). In non-rapid eye movement sleep, the brain is less active but the body can move. Non rapid eye movement sleep is composed of four stages that are differentiated on the basis of EEG characteristics. When normal individuals first fall asleep, they enter stage 1 (sleep drowsiness) and then progress through stages 2, 3, and 4 of non-rapid eye movement. Stages 3 and 4 (deep sleep) are often called slow wave sleep or delta sleep because they are characterized by high amplitude, slow waves (also called delta waves) on EEG. Slow wave sleep may last from a few minutes to an hour, depending on the person's age, before reversion back to stage 2 sleep. Shortly after this, the first REM sleep period begins, lasting about 15–20 min and is followed by another non-REM cycle. This alternating pattern continues throughout the night, but as the night progresses stages 3 and 4 are less apparent and the periods of REM sleep grow longer. These different phases of sleep can be analyzed by extracting the relevant features from the EEG signal. This helps in finding the sleep disorder.

Epilepsy is a neuron disorder from which the general function of the brain is affected. There are three typical stages of epilepsy like interictal, preictal and ictal. Therefore, a model can be designed to identify the epileptic and can be classified by considering the objectives like maximization of classification accuracy, minimization of number of features, and minimization of number of instances. Hence, it can be inferred that many problems through EEG signal can be viewed as multi objective problems, therefore, suitable multi-objective swarm intelligence techniques can be used as a tool.

5 Discussion and Future Research

Over the years many swarm intelligence techniques like ant colony optimization, particle swarm optimization, artificial bee colony, Bat algorithm [169], Wasp colony optimization [170], etc. have been developed to solve many intractable/complex single, multiple, and many objective optimization problems. However, this paper restrict its discussion with ant colony optimization, particle swarm optimization, and artificial bee colony to solve multiple and many objective optimization problems. We have started our work with an intensive study on multi-objective optimization problems and how it differs from many objective problems. The exploration and exploitation capability of ACO, PSO, and ABC has been studied along with the diversity mechanism. Next part of the study, a basic framework of these swarm intelligent techniques is presented to explain how they can be used for handling many objective problems. This framework is a common one for all the three categories of swarm optimization technique. We have also discussed about some of the indicators which are used for handling many objective problem. This part presents a clear view on the many objective problems and how they can be handled.

At the last part of this work, we have studied the EEG signal analysis as it is now one of the promising area of BCI. This paper puts a light on the basic of EEG signal and in one section also we discuss multi objective issues associated with EEG signal analysis. We focus our study on how the swarm intelligence techniques are used for EEG signal analysis. In a nutshell, our paper presents a clear view on swarm intelligent techniques and their usages in handling multiple and many objective problems commendable in the field of EEG signal analysis.

References

1. Bonabeau, E., Dorigo, M., Thraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems, vol. 23. Oxford University Press, Oxford (1999)
2. Rafie, F.M., Manzari, S.M., Khashei, M.: Scheduling flight perturbations with ant colony optimization approach. Int. J. Comput. Sci. Artif. Intell. **2**(2), 1–9 (2012)
3. Sashikumar, G.N., Mahendra, A.K., Gouthaman, G.: Multi-objective shape optimization using ant colony coupled computational fluid dynamics solver. Comput. Fluids **46**(1), 298–305 (2011)
4. Prasad, S., Zaheeruddin, Lobiyal, D.K.: Multi-objective multicast routing in wireless ad-hoc networks: an ANT colony approach. In: Proceedings of IEEE 3rd International Conference

in Advance Computing, pp. 511–514 (2013)
5. Goa, S.: Solving weapon-target assignment problems by a new ANT colony algorithm. International Symposium on Computational Intelligence & Design, pp. 221–224. IEEE Press (2008)
6. Yang, X.S.: Nature-inspired metaheuristic algorithms. Luniver Press, Beckington, UK (2008)
7. Lim, C.P., Jain, L.C., Dehuri, S. (eds.): Innovations in Swarm Intelligence. Springer, Berlin (2009)
8. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: a short review. In: Proceedings of 2008 IEEE Congress on Evolutionary Computation, pp. 2424–2431. Hong Kong (2008)
9. Millonas, M.M.: Swarms, phase transition and collective intelligence. In: Langton, C.G. (ed.) Artificial Life III. Addison Wesley, Reading (1994)
10. Karaboga, D.: Artificial be colony algorithm. Scholarpedia **5**(3), 6915 (2010)
11. Dehuri, S., Cho, S.-B., Ghosh, S. (eds.): Integration of Swarm Intelligence and Artificial Neural Networks. World Scientific Press, New Jersey (2011)
12. Dehuri, S., Cho, S.-B. (eds.): Knowledge Mining Using Intelligent Agents. Imperial College Press, London (2010)
13. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. Mag. **22**(3), 52–67 (2002)
14. Mishra, B.S.P., Dehuri, S., Wang, G.-N.: A state-of-the-art review of artificial bee colony in the optimization of single and multi-criteria. Int. J. Appl. Metaheuristics Comput. **4**(4), 23–45 (2013)
15. Sato, H., Aguirre, H.E., Tanaka, K.: Controlling dominance area of solutions and its impact on the performance of MOEAs. Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization- EMO 2007. Springer, Berlin (2007)
16. Ikeda, K., Kita, H., Kobayashi, S.: Failure of pareto-based MOEAs: does non dominated really mean near to optimal? In: Proceedings of 2001 IEEE Congress on Evolutionary Computation, pp. 957–962. Seoul (2001)
17. Branke, J., Kauler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization. Adv. Eng. Softw. **32**(6), 499–507 (2001)
18. Branke, J., Deb, K.: Integrating user preferences into evolutionary multi-objective optimization. In: Jin, Y. (ed.) Knowledge Incorporation in Evolutionary Computation, pp. 461–477. Springer, Heidelberg (2005)
19. Drechsler, N., Drechsler, R., Becker, B.: Multi-objective optimization based on relation. Lecture Notes in Computer Science 1993: Evolutionary Multi-Criterion Optimization - EMO 2001. pp. 154–166. Springer, Berlin (2001)
20. Slflow, A., Drechsler, N., Drechsler, R.: Robust multi-objective optimization in high dimensional spaces. Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007, pp. 715–726. Springer, Berlin (2007)
21. Kppen, M., Yoshida, K.: Substitute distance assignments in NSGA-II for handling many-objective optimization problems. Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007, pp.727–741. Springer, Berlin (2007)
22. Corne, D., Knowles, J.: Techniques for highly multi-objective optimization: some non-dominated points are better than others. In: Proceedings of 2007 Genetic and Evolutionary Computation Conference, pp. 773–780. London (2007)
23. Kukkonen, S., Lampinen, J.: Ranking-dominance and many objective optimization. In: Proceedings of 2007 IEEE Congress on Evolutionary Computation, pp. 3983–3990. Singapore (2007)
24. Wagner, T., Beume, N., Naujoks, B.: Pareto, aggregation and indicator-based methods in many-objective optimization. Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007, pp. 742–756. Springer, Berlin (2007)
25. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Iterative approach to indicator-based multi-objective optimization. In: Proceedings of 2007 IEEE Congress on Evolutionary Computation, pp. 3697–3704. Singapore (2007)

26. Brockhoff, D., Zitzler, E.: Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. Lecture Notes in Computer Science 4193: Parallel Problem Solving from Nature - PPSN IX, pp. 533–542. Springer, Berlin (2006)
27. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer, Boston (1998)
28. Knowles, J.D., Corne, D.W.: On metrics for comparing non dominated sets. In: Proceedings of 2002 IEEE Congress on Evolutionary Computation, pp. 711–716. Honolulu (2002)
29. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance assessment of multi-objective optimizers: ananalysis and review. IEEE Trans. Evol. Comput. **7**(2), 117–132 (2003)
30. Okabe, T., Jin, Y., Sendhoff, B.: A Critical Survey of Performance Indices for Multi-objective Optimization. In: Proceedings of 2003 IEEE Congress on Evolutionary Computation, pp. 878–885. Canberra (2003)
31. Ishibuchi, H., Doi, T., Nojima, Y.: Incorporation of scalarizing fitness functions into evolutionary multi-objective optimization algorithms. Lecture Notes in Computer Science 4193: Parallel Problem Solving from Nature - PPSN IX, pp. 493–502. Springer, Berlin (2006)
32. Ishibuchi, H., Nojima, Y.: Optimization of scalarizing functions through evolutionary multi-objective optimization. Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007, pp. 51–65. Springer, Berlin (2007)
33. Hughes, E.J.: MSOPS-II: a general-purpose many-objective optimizer. In: Proceedings of 2007 IEEE Congress on Evolutionary Computation, pp. 3944–3951. Singapore, pp. 25–28 (2007)
34. Hughes, E.J.: Multiple single objective pareto sampling. In: Proceedings of 2003 IEEE Congress on Evolutionary Computation, pp. 2678–2684. Canberra (2003)
35. Ishibuchi, H., Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling. IEEE Trans. Syst. Man Cybern. - Part C **28**(3), 392–403 (1998)
36. Jaszkiewicz, A.: Genetic local search for multi-objective combinatorial optimization. Eur. J. Oper. Res. **137**(1), 50–71 (2002)
37. Ishibuchi, H., Yoshida, T., Murata, T.: Balance between genetic search and local search in Memetic algorithms for multi-objective permutation flowshop scheduling. IEEE Trans. Evol. Comput. **7**(2), 204–223 (2003)
38. Fleming, P.J., Purshouse, R.C., Lygoe, R.J.: Many-objective optimization: an engineering design perspective. Lecture Notes in Computer Science 3410: Evolutionary Multi-Criterion Optimization - EMO 2005, pp. 14–32. Springer, Berlin (2005)
39. Deb, K., Sundar, J.: Preference point based multi-objective optimization using evolutionary algorithms. In: Proceedings of 2006 Genetic and Evolutionary Computation Conference, pp. 635–642. Seattle (2006)
40. Thiele, L., Miettinen, K., Korhonen, P.J., Molina, J.: A Preference Based Interactive Evolutionary Algorithm for Multiobjective Optimization. Helsinki School of Economics, Helsinki (2007). Working Paper, W-412
41. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: on the design of pareto-compliant indicators via weighted integration. Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007, pp. 862–876. Springer, Berlin (2007)
42. Fonseca, C.M., Fleming, P.J.: Multi-objective optimization and multiple constraint handling with evolutionary algorithms - Part I: a unified formulation. IEEE Trans. Syst. Man Cybern. - Part A **28**(1), 38–47 (1998)
43. Coello, C.A.C.: Handling preferences in evolutionary multi-objective optimization: a survey. In: Proceedings of 2000 IEEE Congress on Evolutionary Computation, pp. 30–37. San Diego (2000)
44. Cvetkovic, D., Parmee, P.: Preferences and their application in evolutionary multi-objective optimization. IEEE Trans. Evol. Comput. **6**(1), 42–57 (2002)
45. Zitzler, E., Thiele, L.: Multi-objective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)

46. Brockhoff, D., Wagner, T., Trautman, H.: On the properties of the R2 indicator. In: Proceedings of 2012 Genetic Evolutionary Computation Conference (GECCO2012), pp. 465–472. ACM Press, Philadelphia (2012)
47. Gomez, R.H., Coello, C.A.C.: MOMBI: A new Metaheuristics for many-objective optimization based on the R2 indicator. IEEE Congr. Evol. Comput. (CEC-2013) **1**, 2488–2495 (2013)
48. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multi-objective selection based on dominated hypervolume. Eur. J. Oper. Res. **180**(3), 1653–1669 (2007)
49. Garca-Martnez, C., Cordon, O., Herrera, F.: Discrete optimization a taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP. Eur. J. Oper. Res. **180**, 116–148 (2007)
50. Chaharsooghi, S.K., Kermani, A.H.M.: An effective Ant Colony Optimization Algorithm (ACO) for Multi-objective Resource Allocation Problem (MORAP). Appl. Math. Comput. **200**(1), 167–177 (2008)
51. Mariano, C.E., Morales, E.: MOAQ: an ant-Q algorithm for multiple objective optimization problems. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Hnavar, V., Jakiela, M., Smith, R.E. (eds.) In: Proceedings of the Genetic and Evolutionary Computing Conference (GECCO 99), pp. 894–901. San Francisco (1999)
52. Iredi, S., Merkle, D., Middendorf, M.: Bi-criterion optimization with multi colony ant algorithms. In: Proceedings First International Conference on Evolutionary Multi-criterion Optimization (EMO01). Lecture Notes in Computer Science 1993, pp. 359–372 (2001)
53. Doerner, K., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C.: Pareto ant colony optimization: a metaheuristics approach to multi-objective portfolio selection. Ann. Oper. Res. **131**(1–4), 79–99 (2004)
54. Gambardella, L., Taillard, E., Agazzi, G.: MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. In: Corne, D., Dorigo, M., Glover, F. (eds.) New Ideas in Optimization, pp. 73–76. McGraw-Hill (1999)
55. Doerner, K., Hartl, R.F., Teimann, M.: Are COMPETants more competent for problem solving? the case of full Truckload transportation. Cent. Eur. J. Oper. Res. **11**(2), 115–141 (2003)
56. Gravel, M., Price, W.L., Gagne, C.: Scheduling continuous casting of aluminium using a multiple objective ant colony optimization Metaheuristics. Eur. J. Oper. Res. **143**(1), 218–229 (2002)
57. Ali, A.-D., Belal, M.A., Al-Zoubi, M.B.: Load balancing of distributed systems based on multiple ant colonies optimization. Am. J. Appl. Sci. **7**(3), 428–433 (2010)
58. Lopez-Ibanez, M., Stutzle, T.: The automatic design of multiobjective ant colony optimization algorithms. IEEE Trans. Evol. Comput. **16**(6), 861–875 (2012)
59. Lopez-Ibanez, M., Stutzle, T.: Automatic configuration of multi-objective ant colony optimization algorithms. In: Dorigo, M., et al., (eds.) Ants. Lecture Notes in Computer Science, vol. 6234, pp. 95–106. Springer, Heidelberg (2010)
60. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1942–1948. Perth (1995)
61. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1945–1950 (1999)
62. Bergh, F.V.D., Engelbrecht, A.P.: A cooperative approach to particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 225–239 (2004)
63. Coello, C.A.C., Lechuga, M.: MOPSO: A proposal for multi-objective particle swarm optimization. In: Proceedings of the 9th IEEE World Congress on Computational Intelligence, pp. 1051–1056. Honolulu (2002)
64. Fieldsend, J.E., Singh, S.: A multi-objective algorithm based upon particle swarm optimization, an efficient data structure and turbulence. In: Proceedings of the Workshop on Computational Intelligence, pp. 37–44. Brimingham (2002)
65. Laumanns, M., Zitzler, E., Thiele, L.: A unified model for multi-objective evolutionary algorithm with Elitism. In: Proceedings of the IEEE World Congress on Evolutionary Computation, pp. 46–53. Piscataway (2000)

66. Hu, X., Eberhart, R.C.: Multi-objective optimization using dynamic neighborhood particle swarm optimization. In: Proceedings of the IEEE World Congress on Evolutionary Computation, pp. 1677–1681. Honolulu (2002)
67. Parsopoulos, K.E., Vrahatis, M.N.: Particle swarm optimization method in multi-objective problem. In: Proceedings of the ACM, Symposium on Applied Computing, pp. 603–607. Madrid (2002)
68. Schaffer, J.D.: Multi-objective optimization with vector evaluated genetic algorithms. In: Proceedings of the First International Conference on Genetic Algorithms, pp. 93–100 (1985)
69. Mostaghim, S., Teich, J.: Strategies for finding good local guides in MultiObjective Particle Swarm Optimization (MOPSO). In: Proceedings of the IEEE Symposium onSwarm Intelligence, pp. 26–33 (2003)
70. Coello, C.A.C., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. IEEE Trans. Evol. Comput. **8**(3), 256–279 (2004)
71. Fieldsend, J.E.: Multi-objective particle swarm optimization methods, Technical Report No. 419. Department of Computer Science, University of Exeter (2004)
72. Ray, T., Liew, K.M.: A swarm metaphor for multi-objective design optimization. Eng. Optim. **34**(2), 141–153 (2002)
73. Pulido, G.T., C. A. C., Coello: Using clustering techniques to improve the performance of a particle swarm optimizer. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 225–237. Seattle (2004)
74. Mostaghim, S., Teich, J.: Particle swarm inspired evolutionary algorithm PSEA for multi-objective optimization problem. In: Proceedings of the IEEE World Congress on Evolutionary Computation, pp. 2292–2297. Canberra (2003)
75. Dorigo, M., Blum, C.: Ant colony optimization theory: a survey. Theor. Comput. Sci. **344**, 243–278 (2005)
76. Li, X.: A non-dominated sorting particle swarm optimizer for multi-objective optimization. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 37–38 (2003)
77. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and Elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002)
78. Sierra, M.R., Coella, C.A.C.: Improving PSO-based multi-objective optimization using crowding, mutation and dominance. In: Proceedings of the 3rd International Conference on Evolutionary Multi-criterion Optimization, pp. 505–519. Guanajuato (2005)
79. Ho, S.L., Shiyou, Y., Lo, E.W.C., Wong, H.C.: A particle swarm optimization b based method for multi-objective design optimization. IEEE Trans. Mag. **41**(5), 1756–1759 (2005)
80. Villalobos-Anias, M.A., Pulido, G.T., Coello, C.A.C.: A proposal to use stripes to maintain diversity in a multi-objective particle swarm optimizer. In: Proceedings of IEEE Swarm Intelligence Symposium, pp. 22–29. Pasadena (2005)
81. Salazar-Lechuga, M., Rowe, J.: Particle swarm optimization problems. In: Proceedings of IEEE World Congress on Evolutionary Computation, pp. 1204–1211. Edinburgh (2005)
82. Janson, S., Merkle, D.: A new multi-objective particle swarm optimization algorithms using clustering applied to automated docking. In: Hybrid Metaheuristics Second International Workshop, pp. 128–142. Barcelona (2005)
83. Lewis, A.: The effect of population density on the performance of a spatial social network algorithm for multi-objective optimization. In: Proceedings of IEEE International Symposium on Parallel & Ditributed Processing, pp. 1–6. Rome (2009)
84. Leong, W.F., Yen, G.G.: Impact of tuning parameters on dynamic swarms in PSO-based multi-objective optimization. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 1317–1324. Hong Kong (2008)
85. Lewis, A., Ireland, D.: Automated solution selection in multi-objective optimization. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 2163–2169 (2008)
86. Cagnina, L., Esquivel, S., Coello, C.A.C.: A Particle Swarm Optimizer for Multi-objective Optimization. J. Comput. Sci. Technol. **5**(4) (2005)
87. Laura, D., Mihai, O.: Evolving the structure of particle swarm optimization algorithm. Evolutionary Computation in Combinatorial Optimization, pp. 25–36. Springer, Berlin (2006)

88. Goldberg, E.F.G., de Souza, G.R., Goldberg, M.C.: Particle swarm optimization for the bi-objective degree constrained minimum spanning tree. IEEE Congress on Evolutionary Computation, pp. 16–21. Sheraton Vancouver Wall Centre Hotel, Vancouver (2006)
89. Ho, S., Ku, W., Jou, J., Hung, M., Ho, S.: Intelligent particle swarm optimization in multi-objective problems. PAKDD 2006. LNAI 3918, pp. 790–800. Springer, Berlin (2006)
90. Koppen, M., Veenhuis, C.: Multi-objective particle swarm optimization by fuzzy- Pareto-dominance meta-heuristic. Int. J. Hybrid Intell. Syst. **3**, 179–186 (2006)
91. Chiu, S., Sun, T., Hsieh, S.: Cross-searching strategy for multi-objective particle swarm optimization. Expert Syst. Appl. **37**(8), 5872–5886 (2010)
92. Peng, W., Zhang, Q.: A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In: IEEE International Conference on Granular Computing, pp. 534–537. Hangzhou (2008)
93. Padhye, N., Branke, J., Mostaghim, S.: Empirical comparison of MOPSO methods guide selection and diversity preservation. In: IEEE Congress on Evolutionary Computation, CEC09, pp. 2516–2523. Trondheim (2009)
94. Cabrera, J.C.F., Coello, C.A.C.: Micro-MOPSO: a multi-objective particle swarm optimizer that uses a very small population size. In: Proceedings of Studies in Computational Intelligence, vol. 261, pp. 83–104. Springer, Berlin (2010)
95. Wang, Y., Yang, Y.: Particle swarm optimization with preference order ranking for multi-objective optimization. Inf. Sci. **179**(12), 1944–1959 (2009)
96. Goh, C.K., Tan, K.C.B., Liu, D.S.B., Chiamb, S.C.: A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design. Eur. J. Oper. Res. **202**, 42–52 (2009)
97. Tsai, S., Sun, T., Liu, C., Hsieh, S., Wu, W., Chiu, S.: An improved multi-objective particle swarm optimizer for multi-objective problems. Expert Syst. Appl. **37**(8), 5872–5886 (2010)
98. Zheng, X., Liu, H.: A hybrid vertical mutation and self-adaptation based MOPSO. Comput. Math. Appl. **57**, 2030–2038 (2009)
99. Wang, Y., Yang, Y.: Particle swarm optimization with preference order ranking for multi-objective optimization. Inf. Sci. **179**(12), 1944–1959 (2009)
100. Tereshko, V., Loengarov, A.: Collective Decision-Making in Honey Bee Foraging Dynamics. Computing and Information Systems **9**(3), 1–7. University of the West of Scotland, UK (2005)
101. Chauhan, N.C., Kartikeyan, M.V., Mittal, A.: Design of RF window using multi-objective particle swarm optimization. In: Proceedings of International Conference on Recent Advances in Microwave Theory and Applications, pp. 34–37. Jaipur (2008)
102. Falcon, R., DePaire, B., Vanhoof, K., Abraham, A.: Towards a suitable reconciliation of the findings in collaborative fuzzy clustering. In: Proceedings of Eighth International Conference on Intelligent Systems Design and Applications, vol. 3, pp. 652–657. IEEE, USA (2008)
103. de Carvalho, A.B., Pozo, A., Vergilio, S., Lenz, A.: Predicting fault proneness of classes through a multi-objective particle swarm optimization algorithm. In: Proceedings of 20th IEEE International Conference on Tools with Artificial Intelligence, vol. 2, pp. 387–394 (2008)
104. Martin, J.E., Pantoja, M.F., Bretones, A.R., Garcia, S.G., de Jong Van Coevorden, C.M., Martin, R.G.: Exploration of multi-objective particle swarm optimization on the design of UWB antennas. In: Proceedings of 3rd European Conference on Antennas and Propagation, pp. 561–565. IEEE, Berlin (2009)
105. Pang, H., Chen, F.: An Optimization Approach for Intersection Signal Timing Based on Multi-Objective Particle Swarm Optimization. In: Proceeding of IEEE Conference on Cybernetics and Intelligent Systems. IEEE, Chengdu (2008)
106. Hazra, J., Sinha, A.K.: Congestion management using multi-objective particle swarm optimization. IEEE Trans. Power Syst. **22**(4), 1726–1734 (2007)
107. Qasem, S.N., Shamsuddin, S.M.: Radial basis function network based on time variant multi-objective particle swarm optimization for medical diseases diagnosis. Appl. Soft Comput. **11**(1), 1427–1438 (2011)

108. Pindoriya, M.N., Singh, S.N., Singh, S.K.: Multi-objective mean-variance skewness model for generation portfolio allocation in electricity markets. Electr. Power Syst. Res. **80**(10), 1314–1321 (2010)
109. Sha, D.Y., Lin, H.: A multi-objective PSO for job-shop scheduling problems. Expert Syst. Appl. **37**(2), 1065–1070 (2010)
110. Wang, L., Singh, C.: Stochastic combined heat and power dispatch based on multi-objective particle swarm optimization. Int. J. Electr. Power Energy Syst. **30**(3), 226–234 (2008)
111. Montalvo, I., Izquierdo, J., Schwarze, S., Prez-Garca, R.: Multi-objective particle swarm optimization applied to water distribution systems design: an approach with human interaction. Math. Comput. Model. (2010)
112. Liu, Y.: Automatic calibration of a rainfall-runoff model using a fast and Elitist multi-objective particle swarm algorithm. Expert Syst. Appl. **36**(5), 9533–9538 (2009)
113. Zhang, W., Liu, Y.: Multi-objective reactive power and voltage control based on fuzzy optimization strategy and fuzzy adaptive particle swarm. Int. J. Electr. Power Energy Syst. **30**(9), 525–532 (2008)
114. Cai, J., Ma, X., Li, Q., Li, L., Peng, H.: A multi-objective chaotic particle swarm optimization for environmental/economic dispatch. Energy Convers. Manag. **50**(5), 1235–1318 (2009)
115. Ganguly, S., Sahoo, N.C., Das, D.: Multi-objective particle swarm optimization based on fuzzy-pareto-dominance for possibilistic planning of electrical distribution systems incorporating distributed generation. Fuzzy Sets Syst. **213**, 47–73 (2013)
116. Sankaran, A., Manne, J.R.: Probabilistic multi-objective optimal design of composite channels using particle swarm optimization. J. Hydraul. Res. **51**(4) (2013)
117. Karaboga, D.: An idea based on honey bee swarm for numerical optimization, Technical Report-TR06. Erciyes University, Engineering Faculty, Computer Engineering Department (2005)
118. Teodorovic, D., Dell, M.O.: Bee colony optimization - a cooperative learning approach to complex transportation problems. In: Proceedings of 10th EWGT Meeting and 16th Mini EURO Conference, pp. 51–60 (2005)
119. Wedde, H., Farooq, M.: The wisdom of the hive applied to mobile ad-hoc networks. In: Proceedings of the Swarm Intelligence Symposium 2005, pp. 341–348. Pasadena (2005)
120. Drias, H.S.S., Yahi, S.: Cooperative bees swarm for solving the maximum weighted satisfiability problem. Computational Intelligence and Bioinspired Systems. LNCS, vol. 3512, pp. 318–325. Springer, Berlin (2005)
121. Chong, C.S., Sivakumar, A.I., Malcolm Low, Y.H., Gay, K.L.: A bee colony optimization algorithm to job shop scheduling. In: Proceedings of the 38th conference on Winter simulation WSC '06, pp. 1954–1961. California (2006)
122. Quijano, N., Passino, K.: Honey bee social foraging algorithms for resource allocation, Part-I: algorithm and theory. In: Proceedings of American Control Conference, ACC '07, pp. 3383–3388. IEEE, New York (2007)
123. Karaboga, D., Akay, B.: A comparative study of artificial bee colony algorithm. Appl. Math. Comput. **214**, 108–132 (2009)
124. Karaboga, D., Ozturk, C.: Neural networks training by artificial bee colony algorithm on pattern classification, pp. 279–292. Neural Network World, Institute of Computer Science AS CR v. v. i, Czech Republic (2009)
125. Karaboga, N.: A new design method based on artificial bee colony algorithm for digital IIR filters. J. Franklin Institute, **346**(4), 328–348 Elsevier, Netherlands (2009)
126. Karaboga, D., Ozturk, C.: A novel clustering approach: Artificial Bee Colony (ABC) algorithm. Appl. Soft Comput. **11**(1), 652–657 (2011)
127. Xu, C., Duan, H.: Artificial Bee Colony (ABC) optimized Edge Potential Function (EPF) approach to target recognition for low-altitude aircraft. Pattern Recognit. Lett. **31**(13), 1759–1772 (2010)
128. Yu, W.-J., Zhang, J., Chen, W.-N.: Adaptive artificial bee colony optimization, GECCO13. In: Proceedings of the Fifteenth Annual Conference on Evolutionary Computation Conference, pp. 153–158 (2013)

129. Omkar, S., Senthilnath, N., Khandelwal, J.R., Naik, G.N., Gopalakrishnan, S.: Artificial Bee Colony (ABC) for multi-objective design optimization of composite structures. J. Appl. Soft Comput. **11**(1), 489–499 (2011)
130. Zou, W., Zhu, Y., Chen, H., Zhang, B.: Solving multi-objective optimization problems using artificial bee colony algorithm. Dyn. Nat. Soc. (569784), 37. Hindawi Publishing Corporation (2011). doi:10.1155/2011/569784
131. Qu, B.Y., Suganthan, P.N.: Multi-objective evolutionary programming without non-domination sorting is up to twenty times faster. In: Proceedings of Congress on Evolutionary Computation, CEC09, pp. 2934–2939 (2009)
132. Li, H., Zhang, Q.: Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Trans. Evol. Comput. **13**(2), 284–302 (2009)
133. Akbari, R., Ziarati, R.H.K., Hassanizadeh, B.: A multi-objective artificial bee colony algorithm. Swarm Evol. Comput. **2**, 39–52 (2012)
134. Akbari, R., Mohammadi, A., Ziarati, K.: A novel bee swarm optimization algorithm for numerical function optimization, communications. Nonlinear Sci. Numer. Simul. **15**(9), 3142–3155 (2010)
135. Xinyi, L., Zunchao1, L., Liqiang, L.: An Artificial Bee Colony Algorithm for Multi-objective Optimization. In: IEEE Second International Conference on Intelligent System Design and Engineering Application, pp. 153–156 (2012)
136. Zhou, G., Wang, L., Xu, Y., Wang, S.: An effective artificial bee colony algorithm for multi-objective flexible job shop scheduling problem. In: Huang, D.-S., et al. (eds.) ICIC, pp. 1–8 (2012)
137. Wang, L., Zhou, G., Xu, Y., Liu, M.: An enhanced pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling. Int. J. Adv. Manuf. Technol. **60**, 1111–1123 (2012)
138. Akay, B.: Synchronous and asynchronous pareto-based multi-objective artificial bee colony algorithms. J. Glob. Optim. **57**, 415–445 (2013)
139. Zhou, X., Shen, J., Li, Y.: Immune based chaotic artificial bee colony multiobjective optimization algorithm. In: Tan, Y., Shi, Y., Mo, H. (eds.) ICSI 2013, Part I. LNCS 7928, pp. 387–395 (2013)
140. Drechsler, N., Drechsler, R., Becker, B.: Multi-objective Optimization Based on Relation Favour. Lecture Notes in Computer Science. 1993: Evolutionary Multi-Criterion Optimization - EMO 2001, pp. 154–166. Springer, Berlin (2001)
141. Zitzler, E., Thiele, L.: Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999)
142. Ishibuchi, H., Nojima, Y.: Optimization of scalarizing functions through evolutionary multi-objective optimization. Lecture Notes in Computer Science **4403**, 51–67 (2007)
143. Julee, T., Park, S.M., Sim, K.B.: Electro encephalography signal grouping and feature classification using harmony Sarch for BCI. J. Appl. Math. **2013**(154539), 1–9 (2013)
144. Stastny, J., Covaka, P., Stancak, A.: EEG signal classification introduction to the problem. Radio Eng. **12**(3), 51–55 (2003)
145. Gonzalez, A.R.: EEG signal processing for BCI application: human computer system interaction background and applications. J. Adv. Intell. Soft Comput. **98**(1) (2012)
146. Teplan, M.: Fundamentals of EEG measurement. Meas. Sci. Rev. **2**(2), 1–11 (2002)
147. Garrett, D., Peterson, D.A., Anderson, C.W., Thaut, M.H.: Comparison of linear, non linear, and feature selection methods for EEG signal classification. IEEE Trans. Neural Syst. Rehabil. Eng. **11**(2) (2003)
148. Ahirwala, M.K., Kumar, A., Singh, G.K.: Adaptive Filtering of EEG/ERP Through Bounded Range Artificial Bee Colony (BR-ABC) algorithm. Journal of Digital Signal Processing **25**, 164–172 (2013)
149. Dries, J.E., Peterson, L.G.: Scaling ant colony optimization with hierarchical reinforcement learning partitioning. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, GECCO ’08, pp. 25–32 (2008)

150. Bursa, M., Lhotska, L.: Modified ant colony clustering method in long term electrocardiogram processing. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2007 (EMBS- 2007), pp. 3249–3252 (2007)
151. Khushaba, R.N., Alsukker, A., Al-Ani, A., Al-jumaily, A.: Intelligent artificial ants based feature extraction from wavelet packet co-efficient for bio-medical signal classification. ISCCSP. Malta (2008)
152. Khushaba, R.N., Alsukker, A., Al-Ani, A., Al-jumaily, A.: A combined ant colony and differential evolution feature selection algorithm. In: Darigo, M., et al. (eds.) ANTS 2008. LNCS-5217, pp. 1–12 (2008)
153. Bursa, M., Lhotska, L.: Ant colony cooperative strategy in electrocardiogram and electroencephalogram data clustering. Stud. Comput. Intell. (SCI) **129**, 323–333 (2008)
154. Kennedy, J., Eberhart, R.C.: A discrete binary version of the particle swarm algorithm. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 4101–4108. IEEE Press, New York (1997)
155. Qiu, L., Li, Y., Yao, D.: A Feasibility Study of EEG Dipole Source Localization Using Particle Swarm Optimization. IEEE Congress on Evolutionary Computation, vol. 1, pp. 720–726 (2005)
156. Pulraj, M.P., Hema, C.R., Nagrajan, R., Yaacob, S., Adom, A.H.: EEG classification using radial basis PSO neural network for brain machine interfaces. In: The 5th Student Conference on Research and Development -SCOReD 2007. Malaysia (2007)
157. Nakamura, T., Tomita, Y., Ito, S., Mitsukura, Y.: A method of obtaining sense of touch by using EEG. 19th IEEE International Symposium on Robot and Human Interactive Communication Principe di Piemonte - Viareggio. Italy, Sept. 12–15 (2010)
158. Alp, Y., Arikan, O., Karakas, S.: Dipole source reconstruction of brain signals by using particle swarm optimization. In: IEEE 17th Conference on Signal Processing and Communication Applications, Antalya, pp. 9–12 (2009)
159. Satti, A.R., Coyle, D., Prasad, G.: Spatio-spectral & temporal parameter searching using class correlation analysis and particle swarm optimization for a brain computer interface. In: Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics. San Antonio (2009)
160. Lin, C.-J., Hsieh, M.-H.: Classification of mental task from EEG data using neural networks based on particle swarm optimization. J. Neurocomput. Elsevier (2009)
161. Ba-Karait, N.O.S., Shamsuddin, S.M., Sudirman, R.: Swarm negative selection algorithm for electroencephalogram signals classification. J. Comput. Sci. **5**(12), 995–1002 (2009)
162. Wei, Q., Wang, Y.: Binary multi-objective particle swarm optimization for channel selection in motor imagery based brain-computer interfaces. In: IEEE 4th International Conference on Biomedical Engineering and Informatics (BME I) (2011)
163. zbeyaz, A., Grsoy, M., oban, R.: Regularization and Kernel parameters optimization based on PSO algorithm in EEG signals classification with SVM. In: 2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU 2011) (2011)
164. Kim, J.-Y., Park, S.-M., Ko, K.-E., Sim, K.-B.: A binary PSO-based optimal EEG channel selection method for a motor imagery based BCI system, ICHIT. Springer, Berlin (2012)
165. Sem a Arslan, Gulay T ezel, Ha kan Islk: EEG signals classification using a hybrid structure of ANN and PSO. Int. J. Future Comput. Commun. **1**(2) (2012)
166. Atyabi, A., Luerssen, M., Fitzgibbon, S.P., Powers, D.M.W.: Adapting subject-independent task-specific EEG feature masks using PSO. In: 2012 IEEE World Congress on Computational Intelligence (CEC), pp. 1–7. Brisbane (2012)
167. Shirvany, Y., Edelvik, F., Jakobsson, S., Hedstrm, A., Persson, M.: Application of particle swarm optimization in Epileptic spike EEG source localization. J. Appl. Soft Comput. **13**(5), 2515–2525 (2013)
168. Rakshit, P., Bhattacharyya, S., Konar, A., Khasnobish, A., Tibarewala, D.N., Janarthanan, R.: Artificial bee colony based feature selection for motor imagery EEG data. In: Bansal, J.C., et al. (eds.) Proceedings of Seventh International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2012), pp. 127–138 (2013)

169. Gandomi, A.H., Yang, X.S., Talatahari, S., Alavi, A.H.: Bat algorithm for constrained optimization tasks. Neural Comput. Appl. **22**(6), 1239–1255 (2013)
170. Dehuri, S., Cho, S.-B. (eds.): Knowledge Mining Using Intelligent Agents, Imperial College Press (2011)
171. Knowles, J., Corne, D.: Quantifying the effects of objective space dimension in evolutionary multi-objective optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) Proceedings of Evolutionary Multi-Criterion Optimization, (EMO 2007). Lecture Notes in Computer Science, vol. 4403, pp. 757–771. Springer, Matshushima (2007)
172. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks. Piscataway, pp. 1942–1948 (1995)
173. Zitzler, E., Thiele, L.: Multi-objective optimization using evolutionary algorithms - a comparative case study. Parallel Problem Solving from Nature - PPSNV. Lecture Notes in Computer Science 1498, pp. 292–301. Springer, Berlin (1998)
174. Knowles, J.D., Corne, D.W., Fleischer, M.: Bounded archiving using the Lebesgue measure, In: Proceedings of 2003 Congress on Evolutionary Computation, pp. 2490–2497. Canberra (2003)
175. Zitzler, E., Knzli, S.: Indicator-based selection in multi-objective search. Parallel Problem Solving from Nature - PPSN VIII. Lecture Notes in Computer Science 3242, pp. 832–842. Springer, Berlin (2004)
176. Emmerich, M., Beume, N., Naujoks, B.: An EMO algorithm using the hypervolume measure as selection criterion. Evolutionary Multi-Criterion Optimization -EMO 2005. Lecture Notes in Computer Science 3410, pp. 62–76. Springer, Berlin (2005)
177. Ibanez M.L., Stutzle, T.: The automatic design of multi-objective ant colony optimization algorithms. IEEE Trans. Evol. Comput. **16**(6) (2012)
178. Mariano, C.E., Morales, E.: A Multiple Objective Ant-Q Algorithm for the Design of Water Distribution Irrigation Networks, Technical Report HC-9904. Institute Mexicano de Tecnologa del Agua, Mexico (1999)
179. Stutzle, T.: MAXMIN Ant System for Quadratic Assignment Problems, Technical Report AIDA-97-04. FG Intellektik, FB Informatik, TU Darmstadt, Germany (1997)
180. Stutzle, T., Hoos, H.H.: MAXMIN ant system. Future Gener. Comput. Syst. J. **16**(8), 889–914 (2000)
181. Baran, B., Schaerer, M.: A multi-objective ant colony system for vehicle routing problem with time windows. In: Proceedings of the 21st IASTED International Conference, pp. 97–102 (2003)
182. Liao, C.J., Tseng, C.T., Luarn, P.: A discrete version of particle swarm optimization for flowshop scheduling problems. J. Comput. Oper. Res. **34**, 3099–3111 (2005)
183. Correa, E.S., Freitas, A.A., Johnson, C.G.: A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 35–42 (2006)
184. Wang, J.: A novel discrete particle swarm optimization based on estimation of distribution. In: International Proceedings on Intelligent Computing, pp. 791–802 (2007)
185. Jarboui, B., Damak, N., Siarry, P., Rebai, A.: A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. J. Appl. Math. Comput. **195**, 299–308 (2007)
186. Zhen, Z., Wang, L., Huang, Z.: Probability-based binary particle swarm optimization algorithm and its application to WFGD control. Proceedings International Conference on Computer Science and Software Engineering **1**, 443–446 (2008)
187. Carlisle, A., Dozier, G.: Adapting particle swarm optimization to dynamic environments. In: Proceedings of International Conference on Artificial Intelligence, pp. 1958–1962 (2000)
188. Zhang, W., Liu, Y.: Adaptive particle swarm optimization for reactive power and voltage control in power systems. In: Proceedings of International Conference in Natural Computation. LNCS 3612, pp. 449–452 (2005)
189. Zhen, Z., Wang, Z., Liu, Y.: An adaptive particle swarm optimization for global optimization. In: Proceedings of Third International Conference on Natural Computation **4**, 8–12 (2007)

190. Chunxia, F., Youhong, W.: An adaptive simple particle swarm optimization algorithm. In: Proceedings of Chinese Control and Decision Conference International Conference on Computer Science and Software Engineering, pp. 3067–3072 (2008)
191. Zhan, Z.H., Zhang, J., Li, Y., Chung, H.S.H.: Adaptive particle swarm optimization. IEEE Trans. Syst. Man Cybern. **39**(6), 1362–1381 (2009)
192. Hongwu, L.: An adaptive chaotic particle swarm optimization. In: Proceedings of International Colloquium on Computing, Communication, Control, and Management **2**, 254–257 (2009)
193. Jian, L., Zhiming, L., Peng, C.: Solving constrained optimization via dual particle swarm optimization with stochastic ranking. In: Proceedings of International Conference on Computer Science and Software Engineering **1**, 1215–1218 (2008)
194. Li, J., Xiao, X.: Multi-swarm and multi-best particle swarm optimization algorithm. In: Proceedings of 7th World Congress on Intelligent Control and Automation, pp. 6281–6286 (2008)
195. Ling, C.A., Gen-ke, Y., Zhi-ming, W.: Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. J. Zhejiang Univ. **7**(4), 607–614 (2005)
196. Pan, G., Duo, Q., Liu, X.: Performance of two improved particle swarm optimization in dynamic optimization environments. In: Proceedings of Sixth International Conference on Intelligent Systems Design and Applications, vol. 2, pp. 1024–1028 (2006)
197. Meissner, M., Schmuker, M., Schneider, G.: Optimized Particle Swarm Optimization (OPSP) and its Application to Artificial Neural Network Training. J. BMC Bioinform. **7** (2006)
198. Tian, D.P., Li, N.Q.: Fuzzy particle swarm optimization algorithm. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 263–267 (2009)
199. Lung, R. I., Dumitrescu, D.: Evolutionary swarm cooperative optimization in dynamic environments. Nat. Comput. (2009)
200. Mousa, A.A., El-Shorbagy, M.A., Abed-El-Wahed, W.F.: Local search based hybrid particle swarm optimization algorithm for multi-objective optimization. Swarm Evol. Comput. **3**, 1–14 (2012)
201. Dorigo, M., Maniezzo, V., Colorni, A.: The ant system : an autocatalytic optimizing process, Technical Report TR91-016, Politecnicodi Milano (1991)
202. Colorni, A., Dorigo, M., Maniezzo, V.: An investigation of some properties of an ant algorithm. In: Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92), pp. 509–520. Elsevier, Belgium (1992)
203. Bullnheimer, B., Hartl, R.F., Straub, C.: A new rank based version of the ant system-A computational study. Cent. Eur. J. Oper. Res. Econ. **7**, 25–38 (1997)
204. Cordon, O., Viana, I.F., Herrera, F., Moreno, L.: A new ACO model interesting evolutionary computation concepts : the best-worst ant system. In: Proceedings of ANTS 2000, pp. 22–29. IRIDIA (2000)

http://www.springer.com/978-3-662-46308-6

Multi-objective Swarm Intelligence
Theoretical Advances and Applications
Dehuri, S.; Jagadev, A.K.; Panda, M. (Eds.)
2015, XIV, 201 p. 60 illus., 11 illus. in color., Hardcover
ISBN: 978-3-662-46308-6