ISSN: 2455 - 1597

ijergs Open Access

International Journal of Engineering Research and Generic Science (IJERGS)
Available online at: https://www.ijergs.in

Volume - 8, Issue - 6, November - December - 2022, Page No. 01 - 10

A Survey on Particle Swarm Optimization

[1]Sonam Rawat, M. Tech Scholar, Department of Computer Science Engineering, Mahila Engineering College, Ajmer

[2]Mukesh Kumar Khandelwal, Assistant Professor, Department of Computer Science Engineering, Mahila Engineering College, Ajmer

Abstract

Nature is great creation itself. Everything accomplishes itself without within required time bounds. Swarm intelligence computational algorithm are taking its inspiration from various natural phenomena, these phenomena can be natural, biological, physical, chemical and genetic kind of real occurrence. A algorithm which take its source of inspiration form collection of birds or fishes, by observing magical way of accomplish the journey for finding the roost or food, without any centralized leadership control, this is known as Particle swarm optimization (PSO). This algorithm is used to find the target solution and optimized the values for final success. This computational search method is well regarded in category of Swarm intelligence. It is most well regarded in field of swarm intelligence. This concept is invented in 1995 year by electrical engineer and social psychologist, namely Russell C. Eberhart and James Kennedy. It is most simplified methodology which gains researcher interest to use it, because there are very few parameters to tune according to problem. As per need of problem, there has been proposed many variants of particle optimization. By Many mathematicians, Theoretical analysis has been proposed for PSO. Further to improve performance, many parameters have been tuned with different setting to gain generalized set for parameter for standard PSO. Further in this direction, in the paper, a survey covering PSO properties has been done and proposed.

Keywords: Overview of PSO, PSO Variants, Inertia Weight, Parametric Study.

Introduction

Particle Swarm Optimization (PSO) is process of simulating the bird's behavior into computation methodologies. It is used for optimized the solution for meta-heuristic algorithms. Birds or Particle find their food or target by random taking information from other members of the group. Over the time, swarm should move forward towards better solution. Every new value taken by any swarm member must be better than previous; this will be known as personally achieved best values for particle itself. All member of swarm communicates with each other and take knowledge for globally achieved best solution. Process of PSO is inclined towards personally achieved best values and globally achieved best values. Swarm members those having better position inform to other members of flocks, and the other members have tendency to move immediately to that place, by global attraction. This happens repeatedly until the best conditions or a food source discovered. PSO is algorithm that belongs to community of Swarm intelligence, so some of SI principles need to follow by PSO to maintain the searching process flow in balance way to perform in adverse phase of application.

Swarm Intelligence Concepts: There are five swarming concepts are applicable to explain swarm intelligence.

Principle	**PSO adherence**
Proximity-principle(PP)	Include an n-dimensional calculation for every step
Quality principle(QP)	react to g best and p best factor to maintain the quality maintain the quality factor
Diverse Response principle (DRP)	Swarm response: p best and g best are given priority. Respond to priority factors.
Stability Principle(SP)	Swarm changes are done only with when g bests do. Stable with g best.
Adaptability Principle(AP)	Swarm changes always with g best changes. Swarm adjusts always g best changes. Adapt the new g best values.

Basics of PSO: Particle has two properties in PSO, that is velocity to move particle in search space and position which is achieved after movement of particles. The algorithm intimate a particle or birds or members flying in the search space and moving towards the global optimum. A particle or (agents) in PSO can be defined as $P_i \epsilon [a, b]$ where i= 1,2,3…D and a,b ϵR, D is represents dimensions and R is abbreviation for real numbers . At start particle position and velocity are randomly initialized in searching region. Best values found for particle position in hyper space is known as Perbest and a best value found in whole swarm is known as Globest.

The formulation of PSO is

1) Particles are updating in parallel as they have individual values.

2) New Updated values has dependency on own previous values and its neighbor's values.

3) Same velocity equation is applying to all particles.

$$V_j(t+1) = V_j(t) + C_1R_1\left(P_{best}(t) - X_j(t)\right) + C_2R_2\,(G_{best}(t) - X_j(t))$$

$$X_j(t+1) = X_j(t) + V_j(t+1)$$

T is total number of iteration; t is current iteration count; j is current particle count

Vj is particle's velocity, Xj is the Particle's position,

Pbest is best location in search space is found so far by particle

Gbest is best location found in search space among all particles

There are two uniform random variable R1 and R2 are used in the ranges [0, 1]

There are two coefficients C1 C2 is used namely leaning factors.

Parametric Analysis of PSO

There are few parameter in PSO still changes in PSO, have great reflection shows in converging towards solutions. Some parameter in PSO and their initialization plays roles in results. Number of swarm size, how particles are allocated in searching reason. Values for velocity is either kept zero or randomized in search space. Initialization for position of particle is uniform distribution or random distribution. These facts show dependency on results. The cognitive factor and social factor values shows impact on exploration and exploitation. The values for C1 and C2 are kept equal then it shows equal emphasis in local and global attractor as well. If the C1 is kept high values than C2 it shows location attraction and trap in local minima before convergence. The values for C2 is greater than C1, it shows the all the focus for global exploration, excessive wandering not able to do enough exploitation. C1 =0 shows the global best concept is implemented only. C2= 0 shows that particle are independent form global influence.

Process Convergence of Particle swarm is categorized three groups-

-all particle leads to convergence to certain point to refined the values for problem.

-all particle leads to convergence to a local optima by less diversity of exploration.

-Expected first hitting time is run time analysis does evaluation until the point is visited

Velocity Clamping

The particle's overall exploration will be under the management of velocity clamping. Velocities are limited to a maximum velocity, V_{max}, in each dimension. Particles may fly past viable solutions if $V_{max}$is set to a high value. Particles will set a certain value of velocity if its speed exceeds the maximum speed limit. Particles may not sufficiently explore beyond locally favorable regions if V_{max} is very modest. The velocity of the particles should be constrained to a suitable range. Here, a new constant called V_{max} is introduced to indicate the highest possible velocity.

If v>V_{max}, then v= V_{max}

If v < - V_{max}, then v = -V_{max}

Maximum velocity V_{max}diversify the global exploration, smaller V_{max}encourages local exploitation.

PSO Variants: There have been modifications made to PSO; we will explore these modifications in more detail in the section that follows. The fundamental modifications imply that the best mathematical model available that is appropriate for the current environment should be used.

Quantum-behaved PSO (QPSO), which was proposed by several researchers and was inspired by ideas from quantum mechanics. For instance, Jau et al. developed a modified QPSO that reduced the impact of observations with outliers by using a least-trimmed-squares approach and a high breakdown regression estimator. Additionally, elitist GA crossover and adaptive SA decay are employed to defeat prematureness and regulate search policy. For continuous nonlinear large-scale problems, Tang et al[3] .'s suggested an improved QPSO technique based on memetic algorithm and memory mechanism. The memory mechanism was used to create a "bird kingdom" with memory capacity, and the memetic algorithm was used to make all particles obtain a few experiences through a search strategy before participating in the evolutionary process. Both of these techniques can enhance the algorithm's ability to perform a global search.

Bare-Bones PSO

The velocity and position update rules are replaced by a process that samples a parametric probability density function in the bare-bones PSO (BBPSO) variant of the PSO algorithm. In order to update some particles in the population, Zhang et al. modified the original BBPSO using both the mutation and crossover concept of the DE method. The effectiveness of the developed algorithm was evaluated using 16 vapor-liquid equilibrium issues and 10 benchmark functions. An adaptive version of the cloud model was proposed by Zhang et al. [3] based on their analysis of the sampling distribution in BBPSO.

Chaotic PSO

PSO has been enhanced through the integration of chaos theory-related ideas. Chaotic PSO is the name given to this kind of PSO (CPSO). Chaotic maps were added to catfish PSO by Chuang et al. The chaos approach was used in the proposed strategy to boost search capacity. To train the weights/biases of a two-hidden-layer forward neural network and create a hybrid crop classifier for polar metric synthetic aperture radar images, Zhang and Wu [4] presented adaptive CPSO (ACPSO). In order to estimate wavelet parameters, Dai et al. [5] presented a unique adaptive chaotic embedded PSO (ACEPSO). In addition to nonlinearly and adaptively adjusting parameters, ACEPSO incorporated chaotic factors into traditional PSO. By assessing the population fitness variance of the particle swarm and the average distance between points, it also determined if the particles were focused or discrete.

Fuzzy PSO

The fuzzy sets theory was merged with PSO to increase its potency. Fuzzy PSO is the name given to this kind of PSO (FPSO). An adaptable FPSO (AFPSO) method was put out by Juang et al. The proposed AFPSO enhanced the precision and effectiveness of searches by using fuzzy set theory to alter the PSO acceleration coefficients. A new variant known as AFPSO-Q1 was created by combining this algorithm with quadratic interpolation and the crossover operator to further improve the global searching capacity.

PSO TVAC

To enhance the performance of ordinary PSO even more, PSO with time-varying acceleration coefficients (TVAC) was proposed. PSOTVAC was given to the new variation. According to Cai et al., the linear automation technique may not always be effective. The social and cognitive learning components were therefore updated in accordance with a previously determined predicted velocity index in a new form called predicted modified PSO with time-varying accelerator coefficients. In contrast to the small cognitive coefficient, which used a huge global search capability, the huge cognitive coefficient offered a large local search capability. PSOTVAC was used by Chaturvedi et al. [4] to resolve the practical economic dispatch problem (EDP). Here, the goal of TVAC was to effectively manage local and global search to prevent early convergence and to find global solutions.

Opposition-Based Learning PSO

When OBL theory was combined with PSO, a new form known as opposition-based PSO was created (OPSO). In order to produce a new population during the learning process, Dhahri and Alimi proposed the OPSO employing the opposite number notion. They integrated BBFNN and OPSO. The OPSO-BBFNN produced a greater generalization performance, according to the results. A more advanced PSO method, known as GOPSO, was developed by Wang et al. [4] that made use of Cauchy mutation and generalized OBL (GOBL). Faster convergence was made possible by GOBL, and imprisoned particles were able to escape from local optima due to the long-tailed Cauchy mutation.

Simplified PSO

In contrast, several researchers disapproved of studies that increased the complexity of PSO; as a result, they tended to simplify ordinary PSO without compromising its efficiency in order to speed up computation, enhance convergence, or make implementation simpler. According to the fitness values, Guochu [4] separated the swarm into three categories, better particles, ordinary particles, and the worst particles. According to three related forms of streamlined algorithm models, these three particle types dynamically evolved. The outcomes demonstrated that simplified PSO (SPSO) performed better in terms of optimization than other enhanced PSOs.

Guaranteed Convergence PSO

When a particle's personal position and the global best position are the same (i.e., x = p best= gbest), then updating velocity or position o will depend on only inertia, as demonstrated by Van den Bergh and Engelbrecht [5], who also demonstrated that the w-PSO does not guarantee convergence to a local optimum. This implies that in order for the particle to move, the inertia weight and previous velocity must both be non-zero. If not, all particles will soon stagnate, leading to an early convergence to a location that is only the best outcome the entire population searched thus far rather than a local or global optimum. This algorithms is created by Engelbrecht and Van den bergh and here is velocity equation

$$\mathbf{v}_{\tau}^{t+1} = \omega \mathbf{v}_{\tau}^{t} + \left(\mathbf{g}_{best}^{t} - \mathbf{x}_{\tau}^{t}\right) + \rho^{t}\left(1 - 2\mathbf{r}_{2_{\tau}}^{t}\right),$$

Random vector r is sample U(0, 1) and other term (rho) is define as:

$$\rho^{t+1} = \begin{cases} 2\rho^{t} & \text{if } CS > s_c, \\ \frac{1}{2}\rho^{t} & \text{if } CF > f_c, \end{cases}$$

Hybridization of PSO: In order to take use of the strengths of both approaches and balance out their drawbacks, PSO was integrated with various traditional and evolutionary optimization algorithms.PSO has been hybridized with other evolutionary algorithms, including GA, DE, and ACO. The following describes the hybridization of PSO with GA, DE, ACO, as well as other methods.

Hybrid GA with PSO

One of the first evolutionary algorithms, GA, was presented by John Holland. Due to the improved convergence performance when compared to the individual PSO and GA, combining PSO with GA is a well-known strategy that has been seriously explored. To address multimodal issues, a hybrid PSO another hybrid GA (GA-PSO) was presented.

Hybrid DE with PSO

R. Storn and K. Price [76] first introduced the population-based technique known as DE in 1995 to address optimization issues. Although they have different functions, the selection, mutation, and crossover operators of GA are also utilized in DE. DE has the benefit of maintaining diversity, but unlike PSO, it cannot keep track of the history of the process. A hybrid DE with PSO (DEPSO) algorithm was suggested in to address economic dispatch issues. This proposed algorithm's overall process is built on DE and allowing PSO produces a second mutant operator. DEPSO demonstrated its ability to produce solid conclusions and efficient computing. DEPSO demonstrated its ability to produce solid conclusions and efficient computing. Increased PSO and DE.

Hybrid PSO with other algorithms

Other algorithms, such as ACO , gravitational method (GSA), grey wolf optimizer (GWO), and simulated annealing (SA) , have been hybridised with PSO in addition to GA and DE. A hybrid of PSO and ACO was suggested in. The hybrid ant particle optimization algorithm is the name of the newly created hybrid algorithm (HAP). PSO and ACO are executed independently on each HAP iteration, yielding a fresh PSO solution and a fresh ACO solution. Out of these two options, the best one is picked to serve as the system's overall best option. The computed global best's parameters are used to update the placements of particles and ants. In comparison to SPSO and ACO, HAP has demonstrated that it is capable of achieving greater results. However, only straightforward and low-dimensional benchmarking functions were used to test HAP. Its effectiveness in difficult, high-dimensional optimization situations requires research.

Single Solution PSO

To find single solutions, many PSO variations can be found. These PSO implementations were created specifically to find a single solution to continuous-valued, unrestricted, static, single-objective optimization problems. The majority of these algorithms can also be used to solve different kinds of problems.

Niching with PSO

The term "niching algorithm" is used in the field of EC to describe algorithms that find many solutions. Generally speaking, speciation refers to the process of discovering multiple solutions or niches. Niching algorithms simulate another natural process in which many people vie for the use of scarce resources in the physical world. A species is a group of people (a particle in the context of PSO) that converge on a single niece. Nieces are partitions of an environment, whereas species are partitions of computational optimization. A niece represents one solution to the problem.

Constraint Optimization using PSO

The feasible area where a solution to the issue can be discovered is reduced by constraints. The goal of optimization algorithms is to find a workable solution. In other words, the optimization algorithm must discover a solution that maximizes the objective function and satisfies all requirements. The algorithm must strike a compromise between the optimal objective function value and the number of constraints violated if it is not possible to satisfy all constraints.

GCPSO with niching PSO (NichePSO) algorithm

The NichePSO first establishes sub-swarm leaders by training the main swarm using a cognition-only model, which is an approach taken from GAs. Then, a sub-swarm radius is determined. As optimization moves forward, particles are subsequently permitted to join sub-swarms, which are then permitted to combine. Particles have converged to their sub-optimal swarm's state once their velocity has been minimized. The method consistently succeeded in converging, but the authors acknowledge that the swarm's initialization was crucial to the outcome (using Faure sequences). [5]

Gaussian PSO

On both unimodal and multimodal functions, Gaussian mutation was tested along with velocity and position update rules. Results from the hybrid were superior to those from GA and PSO used separately. Included crossover, elitism, and mutation. The elites, or top half of the best performers, are seen as a swarm and are strengthened by PSO. In the new generation, the enhanced elites make up half of the population, and the other half is produced via crossover and mutation procedures on the enhanced elites. This method outperformed both PSO and GA in the study and mimics the maturation process in nature.

Cooperative PSO

Much more specialised a cooperative, evolutionary genetic algorithm By treating each dimension as a separate optimization problem, this seeks to reduce the exponential rise in difficulty that comes with increasing the problem's dimensions. A number of cooperative solutions have been created, wherein tiny swarms attack each dimension, and cross-dimension communication enables the overall solution to go in the right direction. Due to the serial nature of swarm evaluations and sub-swarms' ability to locate pseudo-minimas, this did nonetheless pose the risk of stagnation [6]. Combining single swarm and cooperative strategies helped to solve these issues. The significant rise in algorithmic complexity is the drawback of cooperative swarms. The writers stay away from this issue since performance is assessed objectively through function evaluations rather than execution time. One of the best applications of Wolpert and Macready's "No Free Lunch" theorems is problems with reduced dimensionality, where the constrained PSO still performs favourably (1997).

Fuzzy Adaptive TPSO (FATPSO)

The TPSO used the principle that PSO’s premature convergence is caused by particles stagnating The TPSO was based on the idea that particles stalling around a less-than-ideal location is what causes PSO's premature convergence. When a particle deviates from the minimal velocity, the velocity memory is replaced by a random turbulence operator. The velocity parameters were then adaptively regulated during an optimization run using the fuzzy logic extension, allowing coarse-grained exploratory searches to take place early on before being substituted by fine-grained exploitation later on. The method performed well against both the low and high dimensionality problems that it was tested on. Notably, the TPSO and FATPSO were mostly unchanged as dimensionality grew, whereas the performance of the classical PSO significantly declined.

PSO with Fast Local Search (FLS)

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

Discrete PSO

When particle's positions are discrete values**,** then Discrete PSO algorithm is applied over discrete-valued search space. The velocity and position equation are developed for problem defined over real values and updated in each iteration. Discrete PSO has a high success rate in solving integer programming problems as compare with other methods, such as branch-and-bound fail. It has a quick convergence and better performance results. [4]

IEPSO

In Immunity-enhanced particle swarm optimization IEPSO, a population of particles is sampled randomly in the feasible space. The population of particles is used to execute PSO or its variants having the updated values of position and velocity. After that, it executes receptor editing operator also known as non- uniform mutation according to a certain probability (pr), and vaccination operator according to probability (pv). The new generation is obtained by the selection operator after the flying of particles and two immune operators receptor editing and vaccination.

Quantum-Behaved PSO algorithm

Quantum-behaved particle swarm optimization algorithm in. Implicit space decomposition is adopted i.e. the whole swarm is divided into several sub-swarms which search the whole space respectively. Particles in different sub-swarms will locate in different areas and evolve in different directions which prevents rapid decline in diversity of the whole swarm effectively.

Multi-objective optimization (MPSO)

In current years, multi-objective optimization has been a very active research area to the researchers. In multi-objective optimization (MO) problems, objective function may be optimized separately from each other and the best solution may be found for each objective. This results in there being a group of alternative solutions. The relevance of each objective relative to the others are considered equivalent in the absence of concern information. The group of alternative solutions is known as Pareto optimal set or Pareto front. The selection of social and cognitive leaders (nBest and pBest) is the key point of MO-PSO algorithms.

Global Best Particle Swarm Optimization

Global best PSO (GBPSO) is one of the standard PSO variant. For finding velocity of the particle in this algorithm the equation is given as below

$$v_i(t+1) = v_i(t) + c_1 r_1 \left(p_{best}(t) - x_i(t)\right) + c_2 r_2 \left(g_{best}(t) - x_i(t)\right)$$

Step by step explanation of Global Best is as follows:

Step 1: Swarm initialization (Random initialization of positions and velocities in search space).

Step 2: Fitness value Calculation for each particle in the swarm.

Step 3: Compare the fitness value with personal best value of each particle, if current value is better, then update personal best value.

Step 4: Set best of all personal bests of all particles as a global best values.

Step 5: Update velocities and positions of all particles.

Step 6: Check for the required criterion met or not. If yes then terminate whole algorithm otherwise again start with step 2.

Decreasing Weight Particle Swarm Optimization

Decreasing weight PSO (DWPSO) is same as the Global best PSO in all manners except that inertial weight is decreased continuously with time [10]. DWPSO concentrates on diversity in initial iterations and on convergence in later iterations. This is the best strategy for getting better results. So, here the velocity equation from (4) changes to

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 \left(p_{best}(t) - x_i(t)\right) + c_2 r_2 \left(g_{best}(t) - x_i(t)\right)$$

Where inertia weight ω at every iteration n is determined by using following equation:

$wn = wf - (wf - wl)n / N$

Where wn is the inertia weight at iteration n,

wf stands for inertia weight for the first iteration and

wl represents inertia weight for the last iteration N.

The inertia weight w depends upon the value of N i.e. total number of iterations in this algorithm.

Behavior of this algorithm at every iteration can be changed by changing the total number of iterations N. To restart an algorithm from certain point is not possible, if N is changed

Time-Varying Acceleration Coefficient PSO

In this variant of PSO, all velocity weights which are inertia weight w, $c1$and $c2$ which are personal best and global best weights vary over time. In this TVACPSO algorithm also the main aim is to achieve a high diversity at starting iterations and a high convergence for ending iterations. The inertia weight changes as in DWPSO. Calculation of velocity is done by using following equation:

$$v_i(n+1) = \omega(n) v_i(n) + c_1(n) r_1 \left(p_{best}(n) - x_i(n)\right) + c_2(n) r_2 \left(g_{best}(n) - x_i(n)\right)$$

Personal best and global best weights are given as follows:

$c1(n) = c1f - (c1f - c1l)n/N$

$c2(n) = c2f - (c2f - c2l)n/N$

$c1(n)$ and $c2(n)$ are the personal best weight and global best weight at iteration n respectively. $c1f$ and $c2f$ are personal best and global best weights designed for first iteration. $c1l$ and $c2l$ are the personal best weight and global best weight designed for last iteration N respectively [5]

Conclusion

In the past several years, Particle Swarm Optimization (PSO) has been successfully applied in many research and application areas such as fuzzy system control, function optimization, artificial neural network training, wireless sensor network image segmentation. It is proved that PSO gets better results in faster, cheaper way as compared to other optimization method. PSO is very popular optimization technique because it has very few parameters to adjust. For wide variety of application, classical PSO can be modified to another version with slight variation in parameter.

Reference

1. Muhammad Imran, Rathiah Hashima and Noor ElaizaAbd Khalid, “An Overview of Particle Swarm Optimization Variants”, Malaysian Technical Universities Conference on Engineering & Technology 2012, MUCET 2012 Part 4 Information And Communication Technology, 2012.
2. Wang Shanli, Long Jun and Wei Zhiyi, "Application of quantum-behaved particle swarm optimization algorithm", IEEE The 26th Chinese Control and Decision Conference (2014 CCDC), 2014.
3. Diego L. Cavalca and Ricardo A. S. Fernandes, "Gradient-based Mechanism For PSO Algorithm: A Comparative Study On Numerical Benchmarks", IEEE 2018 IEEE Congress on Evolutionary Computation (CEC), 2018.
4. Dan Zhang and Bin Wei, "Comparison between differential evolution and particle swarm optimization algorithms", 2014 IEEE International Conference on Mechatronics and Automation, 2014.
5. Zhang, Yudong, Shuihua Wang, and GenlinJi. "A comprehensive survey on particle swarm optimization algorithm and its applications." Mathematical problems in engineering, 2015.
6. Shami, T. M., El-Saleh, A. A., Alswaitti, M., Al-Tashi, Q., Summakieh, M. A., &Mirjalili, S., “Particle swarm optimization: A comprehensive survey”, IEEE Access, 2022.
7. Ashok Kumar and Brajesh Kumar Singh, “Particle Swarm Optimization: A Study of Variants and Their Applications”, International Journal of Computer Applications, Volume 135, No.5, February 2016
8. Houssein, E. H., Gad, A. G., Hussain, K., &Suganthan, P. N., “Major advances in particle swarm optimization: theory, analysis, and application”, Swarm and Evolutionary Computation, Vol. 63, 2021.
9. Yarat, Serhat, SibelSenan, and Zeynep Orman, "A Comparative Study on PSO with Other Metaheuristic Methods", Applying Particle Swarm Optimization, pp. 49-72, 2021.
10. Yuhui Shi and R. Eberhart, "A Modified Particle Swarm Optimizer", Proc IEEE International confernce Evolutionary Computation. Anchorage, pp. 69-73, 1998.
11. Yamille del Valle et. al., "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", IEEE Transactions on Evolutionary Computation, Volume: 12, Issue: 2, pp. 171-195, April 2008.
12. Soodabeh Darzi, Tiong SiehKiong and Balasem Salem, “Overview of Particle Swarm Optimization (PSO) on its Applications and Methods”, Australian Journal of Basic and Applied Sciences, vol 7, issue. 2, pp. 490-499, 2013.
13. Soedibyo, Mohamad Ridwan, Andri Pradipta, Sjamsjul Anam and Mochamad Ashari, "Particle swarm optimization based maximum power point tracking for seawater battery application", IEEE International Conference on Innovative Research and Development (ICIRD), 2018.
14. Yahya Rahmat Samii, "Evolutionary Optimization Methods for Engineering: Part II - Particle Swarm Optimization", IEEE Course Fields, Waves & Electromagnetic, Feb 2011.