Cooperation and Learning the Selection of Parameters in the Particle Swarm Optimization Algorithm (PSO)

Adam Mrozek

University of Economics and University of Silesian, Katowice, Poland
adam.mrozek@ue.katowice.pl

Abstract. The paper introduces the subject and scope of planned and partly implemented works during the preparation of the dissertation. The topic of establishing parameters of heuristic methods through machine learning, undertaken in the article, assumes conducting simulation tests of continuous optimization of selected test problems (functions) called benchmarks. The Particle Swarm Optimization (PSO) method and several variants were selected for the study. The basic algorithms have been subjected to author's modifications introducing the principles of cooperation between the particles participating in the method. The research initiated by the author is to confirm or exclude the hypothesis that the rules governing a swarm of particles and aimed at improving the process of searching for optimal solutions to the problem, can be developed through science. This research, referred to above, will be implemented by means of a neural network. As the quality criterion of the solutions obtained, several parameters were proposed, among which the universality of applications and resistance to changes in initial conditions are to be decisive.

Keywords: PSO, Artificial Neural Networks, Tuning.

1 Introduction

Optimization of processes occurring in the continuous field is a fundamental problem in many branches of economy, economics and technology. A number of methods are known for determining optimal quantities for simple runs of the variables studied. Difficulties arise when there are more decision variables, the course of the objective function is not smooth, there are many (at least two) evaluation criteria. There are known methods for solving each of the above cases individually. When the course of the objective function is known in an analytical manner, it is possible to find the values of decision variables in the same way. Finding an analytical solution determining the extreme of any function can be troublesome when it is formulated in a complicated way. In many cases, there are no automatic methods for finding such formulas. Well-known methods lead to their complex forms, which then require reduction. A separate issue (in addition to the high cost of calculations) is to ensure that the results obtained are accurate and reliable. Often, the cost of calculations related to the verification of the admissibility of a given solution is much less than finding the optimal solution, but the difficulty is to find such a solution that would meet the admissibility condition. Assuming the acceptable quality of the data held and their distribution, it is possible to use one of the numerical or approximation methods.When a given probability distribution can be assumed, it is reasonable to use probabilistic methods, otherwise random methods are useful. In such methods one has to assume either some accuracy or minimal probability of extreme occurrence within certain limits. The course of variability of the objective functions under study has a big influence on the effectiveness of the methods. Even for the smooth nature of these functions, the use of gradient methods based on partial derivatives of the first or higher degrees does not bring the desired effects in the absence or excessive variability of this function. The results stabilize in local extremes. These methods can not be used when the objective function is noisy or is not differentiable over the entire range of specificity. Many simple methods (referred to above) can not be applied directly to each case. It is usually necessary to pre-match them to a specific problem. The optimization process is divided into several stages. One of them is the initial transformation of the search space. It is necessary to know the transformation inverse to a given one, which is not always possible. A feature of some systems is the simplicity of the transforming function and the high complexity of the inverse function (and not when it is missing). At each stage of the calculation, separate objectives are realized. Calculations run in the so-called iterations each time getting closer to reaching the goal. Very big influence on the convergence of these methods has such parameters as determining the initial state and individual parameters of the applied method.

Therefore, the so-called construction methods and further approximations - correction methods. It is inherently necessary to carry out the parameter tuning process to match the general methods. This stage is indispensable especially for simple heuristic methods. More complex methods use hybrid techniques that combine many methods [1]. Applying appropriate strategies in relation to lower methods, so-called metaheuristics. Many such methods based on the laws of nature have been developed. The goal of imitating nature is first of all to rationalize costs in terms of results. Another approach would be to replace the nature of the optimization problem with the decision-making one, in which the method must decide which variant of the many available to choose for the later stages.

The rest of the article has the following structure: section 2 will provide an overview of the problem discussed, which is finding the best parameter values found in continuous optimization methods for waveforms with unknown characteristics. In particular, discussing the tuning of heuristic methods and determining their characteristic parameters. This section has three subsections. The first presentation presented by the author hypothesis and the proposed method of its verification. Another one explains the heuristics method used in the study separately. The third one is devoted to two selected methods and their cooperation: PSO and machine learning methods found in hybrids. The third section is the proposed work plan. the fourth section is a summary and conclusions drawn during literature studies and initial own research. The last section contains a list of literature.

2 Problem Statement

The overarching problem raised in many works is the adjustment of parameters of heuristic methods to the problem which is solved by these methods. Many concepts have been developed, whose main goal was to reduce the algorithm's workload and at the same time increase its versatility. These two concepts are in conflict with each other.

The basic PSO algorithm is readily used during optimization of engineering processes [2] due to its simplicity. In the field of economics, its modifications are used adapted to discrete or mixed optimization, e.g. for flowshop scheduling problems [3] as well as transporting route problem (Traveling Salesman Problem) [4] [5] , and many others. The author, however, will limit his research on test functions for which characteristics and extreme places are known.

Due to the course of parameter setting methods, researchers distinguished their two groups called offline and online. The former are based on the initial selection of parameters before the actual calculations. Such parameters can be both qualitative and quantitative factors. For example, by converting one character to another in the hope of achieving a narrowing of the range of variability of these parameters. in quantitative methods for a given instance of a method, by means of preliminary calculations, the values of parameters are determined for which these calculations proved to be the most advantageous taking into account the selected criterion. this whole process would have to be repeated on a case-by-case basis. The disadvantage of offline methods is the high computational cost, especially used for each problem occurrence.
In general, online methods are divided into:

- deterministic - they require knowledge of the process,
- adaptive - adaptation rules imposed from above,
- self-adaptive - knowledge of the impact of other measurements on the current operation of the system.

The key factor here is the assessment of the results of the progress of the method as a whole, rather than individual components. Such correlation occurs in population methods [6]. What is characteristic is the degree of relationships between individual components: called individuals, particles, individuals representing parallel implementation of a single computational process. Earlier research conducted by the author in his master's thesis confirmed the supposition that the effectiveness of simple strategies drastically decreases with the increase of the dimension of problems.

2.1 The thesis and method of verification

Taking up issues related to the above problems, the author put the following thesis: "The neural network can perform the heuristic algorithm tuning process during calculations".

To prove the thesis the author proposes the following steps:

- Formulation of quality measures and confrontation of the proposed method with others known from the literature.

- Research apparatus and methods for collecting and analyzing results.
- Visualization of results and conclusions from the results achieved.
- Suggestions for the future.

After performing simulation experiments, the following results are expected:
- Reduction of the computing effort by reducing the size of the herd.
- Division into several smaller subgroups of particles with different properties depending on the characteristics of the local environment.
- Dynamic adjustment of the learning process and results control.
- A smaller spread of results.

The following are the envisioned obstacles:
- Incorrect wording of quality measures and lack of improvement in the algorithm's efficiency.
- The process will turn out to be divergent
- The algorithm will lose in general - it specializes in a single case.
- Change in the nature of the tested substrate will take place before the learning cycle ends.
- The choice of the method will turn out to be irrelevant.
- The simpler methods will be more effective.

The author has not encountered successful implementations of his method, but there are many alternative solutions for optimization problems using various modifications of the PSO algorithm. The quality and length of the learning method may not be affected by factors that are part of the assessment function. In particular, there can be many such measures. It is also known that the results of the operation of neural networks largely depend on the topology, the number of individual layers and the neurons themselves, as well as on the learning time. Just as simple shapes are learned faster and complex ones are slower, so in the case of complexity, the functions are subject to optimization. In extreme cases, there may be a phenomenon of overfitting or ignorance. Moreover, the functions are not uniform throughout the space, let alone dynamic problems in which they change during calculations. Recent fears point to the act that the method is quite laborious and could rather be used as an offline method and as an online method to be of low efficiency in relation to simpler heuristics.

2.2 Metaheuristics applied

The method of interest - Particle Swarm Optimization (PSO) - combines many desirable features. Compared to other methods, it is characterized by the relative simplicity of the structure. There is a random element in the form of a generation of initial particle positions and motion parameters, as well as aspect ratios between the components. In contrast to genetic methods [7], there is no crossing in it, although other effects (selected particles) have been introduced on the parameters of the others. This impact can be global - in relation to the entire population of particles or local - to selected ones.

The canonical form of the method is based on two formulas: calculating the speed components (v) and the new position (x):

$$\begin{cases} v_{i,d}(t+1) = v_{i,d}(t) + U(0,c_1)\left(p_{i,d}(t) - x_{i,d}(t)\right) + U(0,c_2)\left(l_{i,d}(t) - x_{i,d}(t)\right) \\ x_{i,d}(t+1) = v_{i,d}(t) + x_{i,d}(t) \end{cases} \tag{1}$$

where:
U is unimodal distribution, c_1 & c_2 are and social parameter, p and l are the best own position and the best known by particle i position in direction d.

The concept of neighborhood can be variously defined, fixed or variable over time with a regular topology or not. There are many variations of the canonical form of this method adapted to different categories of problems: continuous, combinatorial or discrete [8][9]. However, there is no one universal form of this method that guarantees satisfactory results in any general case. It proved the convergence of the method for selected specific variants of it. In this form, the speed of the particles increased too quickly and the particles exceeded the limits. The way was to be a constant inertia ω:

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + c_1 r_{1,d}(t) \cdot \left(p_{i,d}(t) - x_{i,d}(t)\right) + c_2 r_{2,d}(t) \cdot \left(l_{i,d}(t) - x_{i,d}(t)\right) \tag{2}$$

The theoretical studies of the convergence of the basic PSO algorithm were conducted by Maurice Clerc [10] and Ender Ozcan and Chilukuri K. Mohan [11]. Theoretical work was carried out on a simplified model. Such simplifications included on eliminating a random factor, adopting certain assumptions as to the form of the objective function, limiting the number of parameters to be tested, or adopting extreme values. Such treatments enabled the use of analytical methods and deriving general convergence conditions of the method. During the course of many studies, the scope and form of the main parameters of the method as well as the indicators of their effectiveness assessment were proposed. This made it possible to compare them with each other as well as, among other things, optimization methods.

The comparisons mainly concern the number of starts of the objective function with the assumed level of acceptable results or the level of this level achieved after the assumed number of runs. The study carried out by the author for the basic variant of this method showed that for a small number of dimensions (up to 6) and for non-sophisticated target functions, the additional overhead on tuning does not compensate for the improvement of results. The use of such techniques is, however, already justified with the more sophisticated form of the objective function or more decision variables (10). The two most popular topologies are so-called "local best" and "global best". The first is to seek a "better partner" in the nearest geographical area and the second - among all those present. While studying the properties of the PSO method, the author limited himself to one of the versions (the simplest) of global best. This version assumes the existence of one dominant individual (alpha) and the next movement of each of the other particles is heading towards it to a greater or lesser extent. The simplest variant in implementation, here in the selection of the dominant entity, the other particles are guided only by a two-step scale of values: is the own assessment of well-being greater than all others. Another version (the best place) of this algorithm allows many local leaders using the same scale but with respect to the limited neighborhood [12] [13].

Although the first one works for simple unimodal and smooth functions, in the case of noisy and polymodal function, it is exposed to many negative phenomena. One of the main is getting stuck in the local extreme. The advice in this case is to stop the process, isolate the leader (or move to a random location) and resume calculations. However, when most well-situated particles are in the field of attraction of this extreme, the above method becomes ineffective, because usually at this stage the majority of the best particles are focused around one extremum and the new leader becomes close to the previous one. Situations can be changed by repeating the calculations from the beginning by re-initialization. To avoid this, a second independent colony with its own policy and leader was used. It is important that the positions of the leaders of each colony are remote from each other. Such a policy was called the best place. The partial co-dependence of individual particles into several colonies is also not excluded. Another idea would be to apply the penalty function imposed both in relation to the boundary conditions and the extremes achieved. This method is successfully used in genetic algorithms. With this method, the proposed algorithm suggests to use the so-called niche. This treatment is aimed at reducing the attractiveness of selected areas of exploitation, for example, to limit the convergence of all particles to one solution. It is an element regulating exploitation and exploration. Because the PSO algorithm has the characteristics of a stochastic algorithm and is additionally non-deterministic, there is no guarantee of finding the absolute optimum. The results may vary on each launch. For this reason, it is important to be convergent. This means that successive iterations are characterized by a smaller and smaller spread of results and they end up in a single value in infinity. As demonstrated in many studies and theoretically - the general form of the algorithm does not meet this requirement. The behavior depends to a large extent on the parameters used. This convergence assurance becomes one of the tasks of controlling the parameters of the algorithm. As the polarization of particles occurs and their trajectory is oriented, another phenomenon is the migration of particles between their colonies. The factor contributing to this phenomenon can be the distance parameter introduced into the algorithm and the measured increase in the quality (welfare) of each of the particles separately. It is this increase in quality that stimulates particles to activity. In the proposed algorithm it influences the order of service.

A different parameter is synchronous or asynchronous updating. By making an analogy to the natural conditions of development, a mass synchronous update is put forward, similar to the parallel evolution of individuals in each population. Such a solution is like searching graphs in a wider way. Due to the lack of parallel mass computing machines architectures, there is a need for interchangeable concurrent processing with a limited number of threads.

The competitive model - asynchronous - assumes an independent course of each thread, including all particles. A species competition of particles appears here. The result is the existence of particles of varying degrees of development. It is necessary to introduce a priority queuing system. In a special case, this situation may lead to starving weaker particles and promote the creation of so-called "cliques" or privileged dominant individuals involved in the calculation. Another negative effect may be the disappearance of diversity, that is, focusing on one best individual. When the quality level is compensated, the so-called flickering, or alternating domination of several individuals.

What causes that the remaining particles alternately follow different leaders (regardless of the distance between them). This makes it difficult to penetrate the search space, especially in the initial phase of the method and finding alternative extremes, as well as the exit from stagnation. These observations prompted the author to research on the influence of links between individual molecules on their development abilities. The proposed modification of the canonical algorithm presupposes the occurrence of particle clusters on the model of an island evolutionary algorithm.

In this algorithm, neighboring molecules merge into clusters, there is also the phenomenon of particle migration between clusters. Cooperation occurs when the leader loses development capability, i.e. the group does not achieve improvement for a certain period of time. It is assumed that the local extreme was achieved then. The factor that is reinitivating is the appearance of another "alpha" (out of group) in the group. The best position (of the current leader) is remembered, and this individual is transferred to another group (a new position is drawn), his role is taken by the leader of that group. In his current group, a new leader is chosen and the whole group does not participate in the competition for several cycles, which allows faster growth of the others, similarly to the destruction in genetic algorithms. The algorithm has one more modification - grace period. An individual who remains in the leader's place for a certain period of time has the right to reelection, i.e. to continue the leadership despite the fall in the quotation - he gets a second or third chance. This treatment is to protect against flickering (temporary extremes). All saved results are output. Calculations are interrupted after reaching a certain level of satisfaction - finding a set number that the objective functions used for each group do not have to be the same. In this case, it is possible to perform multi-criteria optimization. It would be good if the number of groups was a multiple of the number of criteria.

2.3 PSO and machine learning

The relationship between the PSO method and machine learning can be presented in two ways. First, try to teach the herp's PSO algorithm by influencing the entire set of parameters, and secondly to optimize the operation of the neural network using one of the herd intelligence algorithms. The literature proposes many solutions setting parameters for the PSO method.[14] [15] [16] [17] There is a division into methods that determine the optimal parameter values before solving the problem (offline methods) and those in which the values of these parameters change dynamically during the actual calculation - (online). Knowing the optimal desired values of the optimized task, it is possible to assess the quality of the process. When we do not know this size, we should at least use the best known result obtained by this or another method. Both theoretical research and numerous experiments on the so-called the benchmark functions did not determine either the best form of the algorithm or the optimal, universal values of individual parameters. Many authors (Clerc [10], Kennedy [18] , Trelea, Van Den Bergh, F. Engelbrecht, A. P. [19], Carlisle [20]) have proposed their values for these parameters. In most cases, they oscillate within a narrow range of values. As the research conducted by the author has shown - for a small number of parameters and simple test functions, standard parameters and other published proposals are equally valuable. Differences appear only when there are more (than 10) parameters and more complex functions. Both known offline methods and online methods are based on the assumption of knowledge of the variability characteristics of the examined surfaces. The above-mentioned methods introduce changes globally to all particles involved in the calculations. The problem arises in a situation in which there are rapid changes in the value of the function examined (wolf pits) or changes are negligible (flat space). Each of the above situations would require a different action. In such a situation it would be desirable that each particle could independently assess the environmental characteristics and decide on the next action depending on its own assessment. The introduction to each individual of his own "intelligence" makes him the so-called program agent. Each such agent should be guided by both individual and social considerations. Taking into account individual predispositions, he has to submit long-term benefits over short-term ones. In practice, this means abandoning the best first strategy, promoting the best solution in the next step and allowing temporary deterioration of the results. On the other hand, also accept errors and make adjustments. In assessing the situation, it is helpful to use communication between other individuals. The variant of the PSO method called full connected introduces some freedom in the choice and influence of the neighborhood. It assumes the influence of each of the particles on all the others, taking into account the established weights. These scales can be adjusted depending on the policies adopted, as well as connections between neurons in the neural nework.

The application of the PSO method to adjust parameters of deep neural networks used for the machine learning process would be based on the fact that the PSO algorithm would control the parameters of neural networks [21][22] - the number of neurons on each layer and replaced the back propagation method commonly used for network learning. As you know, the backward propagation method is very time-consuming and requires a lot of examples. There is also the risk of overtraining the network. The optimization of such a structure belongs to the category of mixed models - continuous and discrete. It seems reasonable to separate the role into two criteria: with the given architecture, we optimize the weights of connections and with fixed weights of connections we optimize the number of neurons in particular layers of this network. The question that the author wants to get the answer is: will the Pareto multi-criteria optimization method be applied here? And will the use of the co-evolutionary method be effective? Sun and Shiyuan and Li, Jianwei attempted to build a method based on coevolving two swarms of particles [23], the introduction of cultural elements into the strategy of PSO were proposed by Y. Huang, Y. Xu, and G. Chen [24]. Whereas the use of a complex ecosystem in many swarms of particles to solve dynamic optimization tasks was introduced in their work by J. J. Liang and P. N. Suganthan [25].

Is there any method of learning in which a swarm of particles will optimize its operation in terms of maximizing the number of successes with the least amount of effort? The set task is similar to the issue of investment diversification known from the economy. It is about minimizing the price-to-profit ratio by assuming a maximum acceptable price and a minimum satisfying profit.

The initial scenario would consist in remembering the values of the algorithm's parameters and the effect in the form of meeting the final condition. Since the aim of the research is not to find a specific solution to the chosen problem but to find such sets of parameters that allow good results on a group of problems - the factor proving quality is average and worst behavior as well as features such as resistance to change and interference.

How to investigate the susceptibility to changing the characteristics of problems? First, we use a set of known test functions [26], and secondly we use a function generator. The function generator is some other random or deterministic function, the argument of which is the function being examined and the result given the function after transformation: a certain mutation of a given function. To know the characteristics of the chosen method, the following statistical values should be used: median, standard deviation, variance

3 Evaluation Plan

Research conducted to refute or confirm the thesis is carried out in the form of computer simulations. Of the many heuristic methods, the Particle Swarm Optimization (PSO) method was chosen as being easy to implement and with promising performance. This method has many applications both in the field of continuous optimization and (after appropriate modifications) in discrete optimization. It also has many varieties. There are also some theoretical considerations giving criteria for the convergence of the method. There are also many test results of the method effectiveness available for functions with different characteristics as well as variable (dynamic) calculations. The study conducted by the author takes place in several stages:

1. Literature analysis and collection of test materials. - That's part of the job done. Many methods of tuning the PSO algorithm have been recognized. And also the results and research methodology and results of other scientists. A review of the methods used to teach neural networks was also made in the context of interaction with other optimization algorithms. A set of test functions with different levels of difficulty was selected.
2. Development of the simulation program - the basic PSO method. - A simple program was implemented to implement the basic PSO method, which was modified. Six simple test functions with a variable number of parameters have been pre-implemented. The functionality of the program has been extended to include counters and statistical meters. Restrictions have been introduced on the total time it takes to run a single algorithm, the stop criterion for no progress. The method was focused on the convergence criterion and the spread of results obtained in subsequent launches. The repeatability of acceptable results in relation to the number of all calls of the evaluation function was adopted as the superior criterion. In the pilot tests, successive changes of individual parameters were applied within theoretically considered sufficient to coincide with the PSO algorithm. The size of the examined functions was limited to 6 variables. The software of the superior algorithm of artificial neural networks, selection of appropriate architecture as well as methods of learning and verification of its results remain to be made.
3. Performing a series of pilot simulations for simple functions and with few variables.

a. The use of fixed parameters recommended in the literature.
b. Using a smooth change of parameters over time.
c. Use of adaptive methods.

4. Modification of the method taking into account the complete (full informal) or partial (partial informal) interdependence of particles.
   a. Regulation of group sizes (subpopulations) and their interaction.
   b. Implementing the strategy of belonging to a given group and migration between groups.
5. Introduction of methods of supervision over the evolution of the method.
   a. Determining the method quality criteria:
      i. Is the method convergent at the assumed time? Adoption of the allowable cost of calculations in which all tests must be successful.
      ii. Resistance to changing the specification of the problem. What is the spread of results for various test functions, including the change in their characteristics during calculations?
      iii. Resistance to changing parameter values. What is the sensitivity of the method, i.e. resistance to interference.
      iv. Impact of random factor. How does the calculation result change using different random number generators?
   b. Introduction of trial and error principles and return methods for herd strategy
      i. Memory mechanisms of previous states
      ii. Selection rules
      iii. Determining the output and output signals in the structure of the artificial neural network.
      iv. Choosing a topology of the neural network
      v. Submission to the learning process.
      vi. Preparation of research results.
   c. Summary of experiments and drawing conclusions in relation to the thesis presented.

4 Conclusions

Summarizing, the method chosen for the study combines both the features of simple heuristics and certain rules of their mutual relations with each other. Making many attempts with different sets of parameters, many authors tried to determine both the general form of this method and the way of changing the value of parameters depending on the solved basic problem. In many cases, the method turned out to be sensitive to the type of problem. The low effectiveness of the method throughout the space of optimized parameters is associated with a relatively long period before the parameter value is changed. The moment of making the decision about the change and the direction of such change is important here. The decision task can be treated as a classification. Such tasks are successfully managed by neural networks. Unfortunately, teaching them is long and requires a lot of data. The intellectual effort of the creators has been focused here on the parameters of neural networks. This may be more difficult than with the basic PSO method. Reviewing the literature on the subject, no realizations of hybrid constructions PSO-Neural Networks were found. However, there are works in which a swarm of particles controls the parameters of the neural network. This may be suggested by the conclusions: The studies conducted on such a hybrid did not confirm its effectiveness, therefore this path of development was abandoned or work on such a hybrid was not undertaken because other (less complex) methods with greater effectiveness than expected using this tuning method appeared. Both premises seem to be differently probable. However, the research task posed by the author is the answer to the question: whether and how the use of neuron networks will improve the operation of the PSO method and not to develop a method of the competitive method to another already known and studied.

5 References

1. Z. Michalewicz i D. B. Fogel: Jak to rozwiązać czyli nowoczesna heurystyka (How to solve it or modern heuristics. Scientific and Technical Publishing House). Wydawnictwo Naukowo-Techniczne, Warszawa (2006).
2. Korab, R., Owczarek, R., Połomski, M.: Optymalizacja nastaw przesuwników fazowych z wykorzystaniem algorytmu roju cząstek. (Optimization of phase shifter settings using a particle swarm algorithm. Electrotechnical Review). Prz. Elektrotechniczny. 93, 60–64 (2017). https://doi.org/10.15199/48.2017.03.15.
3. Liao, C.J., Chao-Tang Tseng, Luarn, P.: A discrete version of particle swarm optimization for flowshop scheduling problems. Comput. Oper. Res. 34, 3099–3111 (2007). https://doi.org/10.1016/j.cor.2005.11.017.
4. Shi, X.H., Liang, Y.C., Lee, H.P., Lu, C., Wang, Q.X.: Particle swarm optimization-based algorithms for TSP and generalized TSP. Inf. Process. Lett. 103, 169–176 (2007). https://doi.org/10.1016/j.ipl.2007.03.010.
5. Ribeiro, P., Schlansker, W.: A Hybrid Particle Swarm and Neural Network Approach for Reactive Power Control. IEE. (2003).
6. J. Kennedy i R. Mendes: Population Structure and Particle Swarm Performance. In: Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002. pp. 1671–1676. IEEE, Honolulu, HI, USA (2002). https://doi.org/10.1109/CEC.2002.1004493.
7. Hassan, R., Cohanim, B., De Weck, O., Venter, G.: A comparison of particle swarm optimization and the genetic algorithm. In: Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. pp. 1138–1150 (2005).
8. Poli, R., Kennedy, J., Blackwell, T.: Particle Swarm Optimization: An Overview. Swarm Intell. 1, 33–57 (2007). https://doi.org/10.1007/s11721-007-0002-0.
9. Imran, M., Hashim, R., Khalid, N.E.A.: An overview of particle swarm optimization variants. In: Procedia Engineering. pp. 491–496. Elsevier Ltd (2013). https://doi.org/10.1016/j.proeng.2013.02.063.
10. Clerc, M.: Standard Particle Swarm Optimisation. HAL Archives Ouvertes (2012).
11. Mohan, E.O. i C.K.: Analysis Of A Simple Particle Swarm Optimization System. ntelligent Eng. Syst. Through Artif. Neural Networks. 253–258 (1998).

12. Gao, Y., Du, W., Yan, G.: Selectively-informed particle swarm optimization. Sci. Rep. 5, (2015). https://doi.org/10.1038/srep09295.
13. Suganthan, P.N.: Particle swarm optimiser with neighbourhood operator. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999. pp. 1958–1962. IEEE Computer Society (1999). https://doi.org/10.1109/CEC.1999.785514.
14. Bonyadi, M.R., Michalewicz, Z.: Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review. Evol. Comput. 25, 1–54 (2017). https://doi.org/10.1162/EVCO_r_00180.
15. Francesca, G., Pellegrini, P., Stützle, T., Birattari, M.: Off-line and on-line tuning: A study on operator selection for a memetic algorithm applied to the QAP. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 203–214 (2011). https://doi.org/10.1007/978-3-642-20364-0_18.
16. Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T.: ParamILS: An automatic algorithm configuration framework. J. Artif. Intell. Res. 36, 267–306 (2009). https://doi.org/10.1613/jair.2808.
17. Ansótegui, C., Malitsky, Y., Samulowitz, H., Sellmann, M., Tierney, K.: Model-based genetic algorithms for algorithm configuration. In: IJCAI International Joint Conference on Artificial Intelligence. pp. 733–739. International Joint Conferences on Artificial Intelligence (2015).
18. Bansal, J.C., Singh, P.K., Saraswat, M., Verma, A., Jadon, S.S., Abraham, A.: Inertia weight strategies in particle swarm optimization. In: Proceedings of the 2011 3rd World Congress on Nature and Biologically Inspired Computing, NaBIC 2011. pp. 633–640. IEEE, Salamanca, Spain (2011). https://doi.org/10.1109/NaBIC.2011.6089659.
19. Van Den Bergh, F., Engelbrecht, A.P.: A study of particle swarm optimization particle trajectories. Inf. Sci. (Ny). 176, 937–971 (2006). https://doi.org/10.1016/j.ins.2005.02.003.
20. Carlisle, A.J.: Applying The Particle Swarm Optimizer To Non-Stationary Environments. 1–163 (2002).
21. Green, R.C., Wang, L., Alam, M.: Training neural networks using Central Force Optimization and Particle Swarm Optimization: Insights and comparisons. Expert Syst. Appl. 39, 555–563 (2012). https://doi.org/10.1016/j.eswa.2011.07.046.
22. Das, G., Pattnaik, P.K., Padhy, S.K.: Artificial Neural Network trained by Particle Swarm Optimization for non-linear channel equalization. Expert Syst. Appl. 41, 3491–3496 (2014). https://doi.org/10.1016/j.eswa.2013.10.053.
23. Sun, S., Li, J.: A two-swarm cooperative particle swarms optimization. Swarm Evol. Comput. 15, 1–18 (2014). https://doi.org/10.1016/j.swevo.2013.10.003.
24. Huang, Y., Xu, Y., Chen, G.: The culture-based particle swarm optimization algorithm. In: Proceedings - 4th International Conference on Natural Computation, ICNC 2008 (2008). https://doi.org/10.1109/ICNC.2008.239.
25. Liang, J.J., Suganthan, P.N.: Dynamic multi-swarm particle swarm optimizer. In: Proceedings - 2005 IEEE Swarm Intelligence Symposium, SIS 2005. pp. 127–132. , Pasadena, CA, USA (2005). https://doi.org/10.1109/SIS.2005.1501611.
26. J. Liang, B. Y. Qu, P.N.S.: Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Tech. Rep. 201311, Comput. Intell. Lab. 1–32 (2013).