International Journal of Swarm Intelligence Research
Volume 14 • Issue 1

A Review on Convergence Analysis of Particle Swarm Optimization

Dereje Tarekegn, Adama Science and Technology University, Ethiopia*
Surafel Tilahun, Addis Ababa Science and Technology University, Ethiopia
Tekle Gemechu, Adama Science and Technology University, Ethiopia

ABSTRACT

Particle swarm optimization (PSO) is one of the popular nature-inspired metaheuristic algorithms. It has been used in different applications. The convergence analysis is among the key theoretical studies in PSO. This paper discusses major contributions in the convergence analysis of PSO. A systematic classification will be used for the review purpose. Possible future works are also highlighted as to investigate the performance of PSO variants to deal with COPs through theoretical perspective and general discussions on experimental results on merits of the proposed approach.

KEYWORDS

In-Variance, Local Convergence, Parameter Selection, Particle Swarm Optimization, Stability Analysis, Topology

1 INTRODUCTION

Particle swarm optimization (PSO) algorithm is a stochastic optimization technique based on the behavior of swarm (Agrafiotis and Cedeno, 2002). Stochastic search algorithms are better suited for solving highly nonlinear problems as compared to deterministic algorithms. In fact, the main motive behind developing stochastic search algorithms is to solve larger problems at a faster rate while maintaining the robustness of the algorithms. The main design idea of the PSO algorithm is closely related to evolutionary algorithms and artificial life. Just like the evolutionary algorithm, PSO is also a population-based algorithm to simultaneously search large region in the solution space of the optimized objective function. It does not necessitate the use of optimized functions such as differential, derivative, and continuous; its convergence rate is fast; and the algorithm is simple and straightforward to implement through programming. Artificial life studies the artificial systems with life characteristics (Liu, 2015). It has been successfully applied to many problems such as artificial neural network training, function optimization, fuzzy control, and pattern classification (Gong et.al., 2017 and Xue et.al., 2019), to name a few. Because of its ease of implementation and fast convergence to acceptable solutions, PSO has received broad attention (*Gong* et.al., 2017*)*. Since 1995, different aspects of the original or basic version of PSO have been modified and many variants have been

DOI: 10.4018/IJSIR.328092 *Corresponding Author

proposed. In PSO, particles can update their positions and velocities according to the environment change, namely it meets the requirements of proximity (the swarm should be able to carry out simple space and time computations) and quality (the swarm should be able to sense the quality change in the environment and the response). The convergence and stochastic stability study of a number of PSO variants, differing from the classical PSO in the statistical distribution of the three PSO parameters: inertia weight, local and global acceleration factors [30]. Besides the robust variant of cuckoo search (CS) algorithm, two additional optimization algorithms, i.e., GA and a modified PSO in the form of repulsive PSO with local search and chaotic perturbation (RPSOLC) are also employed for solving the considered optimization problem. The relative optimization performance of these three algorithms is also evaluated. After validating the numerical optimization procedures of the algorithms under consideration, numerous test problems available in the literature having different boundary conditions, skew angles, and aspect ratios are finally solved, and the derived solutions are reported (Kalita et.al., 2021). They presented an analytical presentation for the top limit of the particle trajectories' second-order stability areas (the so-called USL curves), which is available for most PSO algorithms. Numerical experiments revealed that adjusting the PSO parameters near to the USL curve yielded the greatest algorithm performance. Although a few review articles on PSO and its convergence analysis (see section 2 for details) have been published already (Banks et.al., 2008 and Poli et.al., 2007b) an important reason for an additional review paper is that the latest comprehensive review paper on convergence analysis of PSO have been published since then. As the latest comprehensive review paper solely on convergence analysis of PSO was published in 2013(Dong ping Tian, 2013). The need for a new review paper seems justified.

The main aim of this survey is to review the presented ideas, categorize and link most recent high-quality studies, and provide a vision for directions that might be valuable for future research. However, we briefly discuss some of these methods from theoretical perspectives: convergence to local optima, transformation invariance, and the time complexity of the methods.

The remainder of this review paper is organized as follows. Section 2 provides some information about the standard form of PSO as well as reviews on the existing literature of convergence analyses of PSO algorithm. Section 3 reviews articles that have identified limitations in convergence analyses of PSO. Section 4 reviews articles which have modified convergence analyses of PSO to have better performance in solving Unconstrained Optimization Problems (UOPs). Finally, the paper ends with some important conclusions in Section 5.

2. STANDARD PSO

In this review paper, we consider a minimization problem defined as follows: find

$$\overrightarrow{x^*} \in S \subseteq \mathbb{R}^d \text{ such that } \forall \vec{x} \in S, f\left(\overrightarrow{x^*}\right) \leq f\left(\vec{x}\right) \tag{1}$$

where S is the search space defined by $\left\{\vec{x} : -\infty < l^i \leq x^i \leq u^i < \infty, i = 1,\ldots,d\right\}$, x^i is the i^{th} dimension of the vector $\vec{x}$, u^i and l^i are upper bound and lower bound of the i^{th} dimension, respectively, d is the number of dimensions, and $f\left(.\right)$ is the objective function. In (Bonyadi and Michalewicz, 2017) the endowed PSO is first established based on the behavior of bird flocks, referred to as swarm of size $n>1$; each particle in the basic PSO contains the main three vectors:

- Position $\left(\vec{x}_t^i\right)$ –is the position of the i^{th} particle in the t^{th} iteration. The quality of the particle is determined by this vector, (that means the swarm should be able to sense the quality change in the environment and response it) (i.e. $\vec{x}_t^i \in R^d$) (Mubeen S and Dr. Dhananjay, 2022).

- Velocity $\left(\vec{V}_t^i\right)$–is the direction and length of movement of the i^{th} particle in the t^{th} iteration,
- Personal best $\left(\vec{P}_t^i\right)$– is the best position that the i^{th} particle has visited in its life-time (up to the t^{th} iteration). This vector serves as a memory to store the location of highest quality solutions found so far.

All of these d-dimensional vectors $\left(\vec{x}_t^i, \vec{V}_t^i, \vec{p}_t^i\right)$ are updated at every run t for each particle i (Bonyadi and Michalewicz, 2014a).

$$\vec{V}_{t+1}^i = \mu\left(\vec{x}_t^i, \vec{V}_t^i, N_t^i\right) \text{for all } i \tag{2}$$

$$\overrightarrow{X}_t^i, = \xi\left(\vec{x}_t^i, \vec{V}_{t+1}^i\right) \text{for all } i \tag{3}$$

$$\vec{P}_{t+1}^i = \begin{cases} \vec{x}_{t+1}^i & for\ f\left(\vec{x}_{t+1}^i\right) < f\left(\vec{P}_t^i\right) and\ \vec{x}_{t+1}^i \in S \\ \vec{P}_t^i, otherwise \end{cases} \quad for\ all\ i \tag{4}$$

In $\vec{V}_{t+1}^i = \mu\left(\vec{x}_t^i, \vec{V}_t^i, N_t^i\right)$ for all i, where N_t^i known as the neighbor set of particle i, is a subset of personal best positions of the particles that contribute to the velocity update rule of particle i at iteration t. i.e. $N_t^i = \left\{P_t^k \middle| k \in \left\{T_t^i \subseteq \left\{1,2,3,...,n\right\}\right\}\right\}$, where T_t^i is a set of indices of particles which contribute to the velocity update rule of particle i at iteration t.

Clearly, the strategy to determine T_t^i might be different for various types of PSO algorithms and it is usually referred to as the topology of the swarm. Many different topologies have been defined for PSO to date (Zou et.al., 2015), while, e.g.; global best topology, ring topology, wheel topology, and pyramid typology; each of these has some topological influence (Cleghorn and Engelbrecht, 2014c, 2015). Topology in fact determines the set of particles from which a particle should learn (connect to). The function $\mu\left(.\right)$ calculates the new velocity vector for particle i according to its current position, current velocity $\vec{V}_t^i$, and neighbor set N_t^i. In $\vec{X}_{t+1}^i, = \xi\left(\vec{x}_t^i, \vec{V}_{t+1}^i\right)$ for all i, where $\xi\left(.\right)$ is a function which calculates the new position of particle *i* according to its previous position and its new velocity. Usually $\xi\left(\vec{x}_t^i, \vec{V}_{t+1}^i\right) = \vec{x}_t^i + \vec{V}_{t+1}^i$ is used for updating the position of particle i. In Eq.4, the new personal best position $\left(\vec{P}_{t+1}^i\right)$ for particle *i* is updated according to the objective value of its previous personal best position and the current position. In PSO, three update rules (Eq. 2, 3, and 4) are applied to all particles iteratively until a predefined stopping criterion is met. Furthermore, $\vec{x}_o^i$ and $\vec{V}_o^i$ are generated randomly and $\vec{p}_o^i$ is initialized to $\vec{x}_o^i$ for all particles.

In the first version of PSO (Bonyadi and Michalewicz, 2017) called "Basic Particle Swarm Optimization", BPSO, the set N_t^i contained only two vectors that were the personal "best" position of the i-th particle $\left(\vec{p}_t^i\right)$ and that of the "best" position in the whole swarm (known as $\vec{g}_t$), where "best" means the location where the particle had obtained the lowest objective function evaluation. i.e., $T_t^i = \left\{i, \tau_t\right\}$ where $\tau_t = \underset{l=\left\{1,2...,n\right\}}{\arg\min}\left(F\left(\vec{p}_t^l\right)\right)$.

This topology(or τ_t) is called global best topology for PSO. Moreover, the function $\mu\left(.\right)$ in Eq.2 was defined as:

$$\vec{V}_{t+1}^i = \vec{V}_t^i + \varphi_1 R_{1t}^i\left(\vec{p}_t^i - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i\left(\vec{g}_t - \vec{x}_t^i\right) \tag{5}$$

where φ_1 and φ_2 are two constant real numbers ($\varphi_1 > 0$ and $\varphi_2 > 0$) called cognitive and social weights, respectively, also known as acceleration coefficients, and $\vec{p}_t^{\,i}$ and $\vec{g}_t$ are the personal best (of particle i) and the global best vectors, respectively, at iteration t. The role of vector factors *CI* $= \left(\vec{p}_t^{\,i} - \vec{x}_t^{\,i}\right)$ (Cognitive Influence) and *SI* $= \left(\vec{g}_t - \vec{x}_t^{\,i}\right)$ (Social Influence) is to attract the particles to move towards known quality solutions, R_{1t}^i and R_{2t}^i are two random dxd diagonal matrices (Cleghorn and Engelbrecht, 2015), where their elements are random numbers distributed uniformly in $\left[0,1\right]$, i.e., $U\left(0,1\right)$. There is no initialization method for velocity that is superior to other methods in a general case, however, it has been recommended in (Engelbrecht, 2012) for the velocity to be initialized to zero.

In 2015, authors (Wang et.al., 2013) introduced a new coefficient $\omega \in \left(0,1\right)$ called inertia weight, to control the influence of the previous velocity value on the updated velocity equation.

$$\vec{V}_{t+1}^{\,i} = \omega \vec{V}_t^{\,i} + \varphi_1 R_{1t}^i \left(\vec{p}_t^{\,i} - \vec{x}_t^{\,i}\right) + \varphi_2 R_{2t}^i \left(\vec{g}_t - \vec{x}_t^{\,i}\right) \tag{6}$$

Thus, the coefficients ω, φ_1 and φ_2 control influences of the previous velocity for the PSO's convergence, *CI,* and *SI* on the particle movement, respectively. This variant of the PSO is called "Standard Particle Swarm Optimization", SPSO. If the random numbers on the diagonal of matrices R_{1t}^i and R_{2t}^i are set to equal values then these matrices only scale the vectors CI and SI along their directions and was introduced as a common error in the implementation of convergence analysis of PSO (Cleghorn and Engelbrecht, 2015).

2.1 Earlier Review Papers

A number review of articles on PSO have been published to date, to name some (Bonyadi and Michalewicz, 2017). In 2002, seven years after first version PSO was introduced, the first review on convergence analysis of PSO papers was published (Dong ping Tian, 2013). The performance of several convergence analyses of PSO variants in locating global optimum, jumping out of local optima, dealing with noise and the guaranteed convergence modifications, solving multi-objective optimization problems, etc. were reviewed. In addition to the review part of that article, a convergence analysis of PSO variant was proposed that used a function stretching approach (Bonyadi and Michalewicz, 2017) to jump out of local optima.

In 2013, a review article (Dong ping Tian, 2013) on convergence analysis of PSO was published that consisted of two main parts: the standard PSO algorithm and convergence analyses of PSO algorithm with six subsections, including convergence analysis with constriction coefficient, limit, differential equation, matrix, difference equation and transformation. The first part contained different strategies for determining parameters of the velocity update rule (inertia, cognitive, and social weights) and moving principle of particles. In the second part, several PSO convergence analyses methods used to justify PSO convergence condition were summarized, and in the other convergence analyses part, some existing studies related to multiobjective (Mo) PSO variants were reviewed.

2.2 Convergence Analyses of PSO

The existing convergence analysis on the PSO algorithm focuses on the constant convergence analysis on the constant transfer matrix and the random convergence analysis on basis of the random transfer matrix. After PSO introduction in 1990s, it has become the focus in optimization community and has been widely applied in many fields. Much attention has been paid to the improvements of PSO itself, such as the improvement of its parameters, including the inertia weight and convergence factor, the improvement of the update formula based on the velocity position, the improvement

based on the topology of particle swarms, the improvement based on the evolutionary mechanism of the genetic algorithm, including selection, crossover and mutation, and the improvement based on the integration of other approaches, viz., the so-called hybrid soft computing (HSC) and extensive investigations to determine the optimal values of different GA parameters conducted in (Kalita et.al., 2019). The global optima search capability of an advanced variant of PSO, Repulsive particle swarm optimization (RPSO) algorithm is enhanced by allowing each particle to perform a local search and having a chaotic perturbation that helps particles avoid getting trapped in the local optima (Ganesh et.al.(2021), Kalita and Chakraborty (2023) and Kalita et.al.(2021)). However, very few research on the PSO's convergence has been studied so far, which plays a crucial role in establishing the solid theoretical foundation for PSO algorithm. So this paper, from another perspective, thoroughly reviews and analyzes the convergence of PSO in the existing literature, the goal is to provide references and suggestions for PSO researchers to establish a solid theoretical basis.

The convergence analysis of the standard PSO algorithm mainly plays a role on the convergence condition, the convergence speed, parameter selection and the trajectory of each particle. This paper mainly concentrates on the random convergence analysis of the PSO algorithm with time-varying attractor $Q\left(t\right)=\frac{\varphi_1 g\left(t\right)+\varphi_2 p\left(t\right)}{\varphi_1+\varphi_2}$ and the convergence analysis on the PSO algorithm with the time-varying attractor $Q\left(t+1\right)\neq Q\left(t\right)$ and the parameter $\eta=\frac{Q\left(t+1\right)-Q\left(t\right)}{v\left(t\right)}$ in some cases, η is not equal to 0. Main contributions on the random convergence analysis of the random PSO algorithm can be highlighted as follows (Liu et al., 2020).

- The convergence condition and the convergence speed are calculated by the spectral radius on the random/constant transfer matrix was less than 1 and the product of two transfer matrices of the PSO algorithm with time-varying attractor less than 1.
- The spectral radius of one transfer matrix cannot determine the convergence behavior and the divergence behavior, while the spectral radius of the product of several transfer matrices may not smaller than 1 can control the convergence behavior and the divergence behavior.
- The convergence condition is also computed from the perspective of mean and variance .

Finally, the convergence analysis of the random PSO algorithm on benchmark functions can demonstrate the effectiveness of the obtained results in the benchmark functions, while different optimization benchmark functions essentially make the different variable $Q\left(t+1\right)-Q\left(t\right)$, and the value of $Q\left(t+1\right)-Q\left(t\right)$ is mainly viewed as $\eta v\left(t\right)$, and the parameter η is introduced in the transfer matrix $M\left(t\right)$ (Kalita and Chakraborty, 2023).

More recently, there have been some other review papers on specific applications of convergence of PSO method, e.g., applications of convergence of PSO with-in-host dengue infection treatment (Handayani et.al., 2016), application of convergence of PSO in Biostatistics (Kim and Li, 2014) to name some. However, they have not been included in this review paper because their focus was on a specific area.

3. IDENTIFIED LIMITATIONS OF CONVERGENCE ANALYSES OF PSO

Several limitations of SPSO have been identified so far. The term "limitation" refers to an issue that has been proven to prevent the algorithm from performing well in different aspects of operation such as locating high quality solutions or being stable. As studied by the work (Bonyadi et al., 2014a) limitations in SPSO, the two main areas related to the limitations of convergence analyses of PSO

are: convergence and transformation invariance. In the following two sub-sections, articles that have analyzed limitations related to these two topics in convergence analyses of PSO are reviewed.

3.1 Limitations Related to Convergence

We classify limitations related to convergence of PSO into four groups: convergence to a point, configurations of movements, convergence to a local optimum, and expected first hitting time (EFHT), these are defined in the following paragraphs.

One of the earliest convergence analyses of stochastic optimization algorithms was published in (Baba, 1981). An iterative stochastic optimization algorithm is said to converge to a point $\vec{X}$ in a search space in probability if

$$\forall \varepsilon > 0, \lim_{t \to \infty} pr\left(\left|\overrightarrow{x_t} - \vec{X}\right|\right) \geq 1 - \varepsilon \tag{7}$$

where Pr is the probability measure, $\overrightarrow{x_t}$ is a generated solution by the optimization algorithm (a point in the search space) at iteration t, and ε is an arbitrary positive value. There are two possibilities for $\forall \varepsilon > 0, \lim_{t \to \infty} pr\left(\left|\overrightarrow{x_t} - \vec{X}\right|\right) \geq 1 - \varepsilon$ to "convergence to a point" and "convergence to a local optimum" for the point $\vec{X}$ is any point and a local optimum of the objective function in the search space, respectively.

As studied by the work of (Bonyadi and Michalewicz, 2017), these two kinds of convergence have been investigated in detail in different optimization algorithms analyses.

Analysis of convergence to a point is usually conducted for an iterative stochastic optimization algorithm to understand whether the sequence of generated solutions produced by the algorithm is convergent. Such analysis sometimes leads to a set of parameters values for the algorithm that guarantee stability. During the convergence process to a point, however, the sequence of the solutions found by the algorithm might follow different configuration of movements:

- High or low frequency movements
- Larger or smaller jumps
- Faster or slower convergence.

These configurations can play important role on the success of the search process (locating higher quality solutions) as they are closely related to the exploration/exploitation ability of the algorithm (Bonyadi and Michalewicz, 2015a).

As it was presented in (Bonyadi and Michalewicz, 2017) analysis of convergence to a local optimum is conducted to understand whether the final solution found by the algorithm is at least a local optimum. Although finding a local optimum is an important (if not essential) property of the algorithm, it is also important to estimate how long it would take for the algorithm to locate that solution. Thus, the expected number of function evaluations to visit a point within an arbitrary vicinity of a local optimum (known as expected first hitting time, EFHT) is analyzed theoretically for optimization methods. This type of analysis, known as EFHT or run time analysis, has been conducted for evolutionary algorithms (Bonyadi and Michalewicz, 2017) where the convergence rate of evolutionary algorithms, as a related topic to EFHT, has been studied.

3.1.1 Convergence to a Point

The velocity of the particles was constrained by a maximum speed V_{max}, which can be used as a constraint to control the global search ability of the particle swarm. In the original PSO algorithm,

$\omega = 1, c_1 = c_2 = 2$, particles' speed often quickly increases to a very high value which will affect the performance of the PSO algorithm, so it is necessary to restrict particle velocity. The issue swarm explosion (Mubeen and Dr. Dhananjay, 2022) pointed out that the velocity vector of particles in SPSO grows to infinity for some values c_1, c_2, and ω , which causes particles to leave the search space and move to infinity. As the search space is bounded (as defined in Section 2), moving outside of the boundaries is not desirable even if there is a better solution (in terms of the objective value) there.

One of the early solutions for this issue was to restrict the value of each dimension of the velocity to a particular interval $\left[-V_{max}, V_{max}\right]$ (Helwig et.al., 2013).

Then, the movement equations become

$$\vec{V}_{t+1}^{i} = \max\{-V_{max}, \min\left\{V_{max}, \vec{v}_t^{i} + \varphi_1 R_{1t}^{i}\left(\vec{P}_t^{i} - \vec{x}_t^{i}\right) + \varphi_2 R_{2t}^{i}\left(\vec{g}_t - \vec{x}_t^{i}\right)\right\}$$

$$\vec{X}_{t+1}^{i} = \vec{x}_t^{i} + \vec{V}_{t+1}^{i}$$

However, this strategy does not prevent swarm explosion in the general case because it used to clamp the velocities of particles and not the position of particles. Some researchers fixed other parameters and only studied the influence of too large and the effect of both too small and large values of speed on the algorithm (Helwig et.al., 2013).

- A too small value of speed will cause the particles to get trapped in local optima.
- A too large value can cause the particles to oscillate around a position.

Hence, some researchers proposed to also restrict the position of the particles (Bonyadi and Michalewicz, 2017) However, even this strategy is not effective because it may restrict the particles to the boundaries of the search space and prevent effective search. A more fundamental solution to the swarm explosion issue can be achieved through analysis of particles' behavior to find out why the sequence of generated solutions might not be convergent is known as stability analysis (Bonyadi and Michalewicz, 2014d, Bonyadi and Michalewicz, 2017, Cleghorn and Engelbrecht, (2014a) and Liu 2015). The aim of this analysis is to define boundaries for the constant parameters in

$$\vec{V}_{t+1}^{i} = \omega\vec{V}_t^{i} + \varphi_1 R_{1t}^{i}\left(\vec{p}_t^{i} - \vec{x}_t^{i}\right) + \varphi_2 R_{2t}^{i}\left(\vec{g}_t - \vec{x}_t^{i}\right)$$

in such a way that the positions of the particles converge to a point in the search space. The set of all boundaries for all coefficients of a PSO variant that guarantee convergence to a point is called convergence boundaries, determining such boundaries is very helpful for parameter setting purposes in SPSO or a variants of PSO. The reason is that coefficients that are outside of the convergence boundaries are usually not appropriate for optimization purposes as they cause particles to move unboundedly. Hence, looking for a combination of coefficients that results in a good performance of the algorithm is focused on a smaller boundary.

The update rules are some times simplified for stability analysis purposes.

$$\vec{V}_{t+1}^{i} = \omega\vec{v}_t^{i} + \varphi_1 R_{1t}^{i}\left(\vec{P}_t^{i} - \vec{x}_t^{i}\right) + \varphi_2 R_{2t}^{i}\left(\vec{g}_t - \vec{x}_t^{i}\right)$$

$$\vec{X}_{t+1}^{i} = \vec{x}_t^{i} + \vec{V}_{t+1}^{i}$$

Velocity and position update rules are analyzed for an arbitrary particle i. We use the notations *p* and *g,* instead of $\vec{p}$ and $\vec{g}$, when we analyze $\vec{p}$ and $\vec{g}$ in a one-dimensional space. Additionally, $\vec{p}_t^i$ and $\vec{g}_t^i$ are assumed to be steady during the run. Furthermore, the topology of the swarm is ignored in the analysis.

One might argue that the convergence boundaries found under these simplifications are not valid as the simplified system might not exactly reflect the behavior of the original system. However, it should be noted that, in these analyses, the behavior of particles is investigated while they are searching for a new personal best. Considering the one-dimensional space for analysis of SPSO is reasonable because all calculations (including generation of the random values on the diagonal of R_{1t}^i) are done in each dimension independently, hence, all analyses in one-dimensional case are also generalizable to multi-dimensional cases as well (Mubeen and Dr. Dhananjay, 2022). In addition, validity of the convergence boundaries found when global and personal bests are steady and the topology is ignored were studied in (Bonyadi and Michalewicz,2015a, and Cleghorn and Engelbrecht, 2014b, 2014c, 2015) experimentally. It was found that, although these conditions simplify the update rules, the calculated convergence boundaries under these conditions are in agreement with those of found under general conditions determined through experiments.

In addition, a recent theoretical study in (Liu, 2015) showed that the convergence boundaries for the global best particle found under this simplification does not change if $\vec{P}_t^i$ is allowed to be updated (under some conditions) during the run.

Now there are many different kinds of researches about the convergence analyses of PSO algorithm based on stability analysis, and four different types of stability analysis can be found in literature:

- Deterministic model stability analysis
- First-order stability analysis
- Second-order stability analysis and
- Third-order stability analysis.

These types of stability analysis for convergence of particle swarm optimization have been differentiated as: (i) Deterministic model stability analysis excludes the stochastic component in particles and assumes that the particles are moved through a deterministic formulation . (ii) First-order stability analysis studies the expectation of the position of particles (Bonyadi and Michalewicz, 2017, Cleghorn and Engelbrecht, 2014a and Tong et. al., 2019) to ensure that this expectation converges. (iii) The second-order stability analysis studies the variance of the position of particles (Cleghorn and Engelbrecht, 2015 and Liu, 2015), to ensure this variance converges to zero (the convergence of the variance of the particle position is a necessary condition for second-order stability). (iv) Third-order stability analysis studies a stochastic quantity on the sequence of particle positions could be the expected value, or variance, or even skewness and kurtosis (Dong and Zhang, 2019).

Perhaps the first study that considered theoretical analysis with no parameter analysis or convergence boundaries of the trajectory of particles in PSO were conducted in (Ozcan and Mohan, 1999), and the findings formed a basis for further analysis. Based on a deterministic model for particles (the stochastic components were ignored), the trajectory and step size of movement for particles were analyzed. It was found that, for BPSO, the trajectory of a particle is of the form of a random *sinus* wave when $\varphi_1 + \varphi_2 < 4$.

One of the earliest attempts to analyze the convergence behavior of SPSO to find convergence boundaries was done for PSO variant in (Bonyadi and Michalewicz, 2017 and Mubeen and Dr. Dhananjay, 2022), In that paper, in order to simplify the formulation of update rules, stochastic components (r_{1t}^i and r_{2t}^i) were omitted from the system (deterministic model analysis). It was proven that, by using this simplified model, particles positions converge to a stable point if

$$\chi = \frac{2k}{\left|2 - c - \sqrt{c^2 - 4c}\right|} \tag{8}$$

where a coefficient known as constriction factor (or χ) is equal to ω, $c_1 + c_2 = c > 4$, $c_1 = \frac{\varphi_1}{\chi}$, $c_2 = \frac{\varphi_2}{\chi}$, and k is a value in the interval $\left(0,1\right]$ (usually set to 1). With these settings the value of χ is in the interval (0, 1]. The value of k controls the speed of convergence to a fixed point, i.e.; the larger the value of k is, the slower the convergence to the fixed point will be. According to these coefficients, the velocity update rule is written as

$$\vec{V}_{t+1}^{i} = \chi\left\{\vec{v}_t^{i} + \varphi_1 R_{1t}^{i}\left(\vec{P}_t^{i} - \vec{x}_t^{i}\right) + \varphi_2 R_{2t}^{i}\left(\vec{g}_t - \vec{x}_t^{i}\right)\right\} \tag{9}$$

where χ is set by Eq.8, $c_1 + c_2 > 4$, and usually $c_1 = c_2 = 2.05$ that results in $\chi = 0.7298$ (Bonyadi and Michalewicz, 2017). By using these settings the value of each dimension of the velocity vector converges to zero and, consequently, velocity does not grow to infinity. The PSO variant that uses velocity update rule in Eq.9 is called "Constriction Coefficient Particle Swarm Optimization", CCPSO. The stable point where each particle *i* converges to is a point between *p* and *g* (namely, $\frac{c_1 g + c_2 p}{c_1 + c_2}$).

The stability of particles in SPSO was also analyzed in (Bonyadi and Michalewicz, 2017). In that study, the random components were replaced by their expected values ($r_{1t}^{i} = r_{2t}^{i} = 0.5$), which enabled first-order stability analysis. The parameters of the velocity update rule were analyzed to find out how they should be set to guarantee particles settle to their equilibrium. Obviously, $V_t^{i} = 0, x_t^{i} = P_t^{i} = g_t$ and $\vec{x}_{t+1}^{i} = \vec{x}_t^{i}$ at the equilibrium point. It was proven that the expected value of the position of each particle settles to its equilibrium (that is $\frac{\varphi_1 g + \varphi_2 p}{\varphi_1 + \varphi_2}$) if and only if

$\omega\left\langle 1, \varphi\right\rangle 0$, and $2\omega - \varphi + 2 > 0$ where . $\varphi = \frac{\varphi_1 + \varphi_2}{2}$

According to (Bonyadi and Michalewicz, 2017), expectation of particle positions converge to its equilibrium if and only if the value of the parameters are set in such away that they lie inside the convergence boundaries to shows that the relationship between ω and φ. (Trelea, 2003) introduced and showed that the parameter setting procedure results in a good performance of the algorithm for $\omega = 0.6$ and $\varphi_1 = \varphi_2 = 1.7$.

Another first-order stability analysis results reported in (Bonyadi and Michalewicz, 2017), including convergence boundaries and convergence to a point between *p* and *g,* were also confirmed in (Zou et.al., 2015). However, in the latter paper, the speed of convergence was investigated in more depth (see configuration section). Subsequently, another first-order stability analysis was conducted in (Campana et.al., 2010), where the findings in (Bonyadi and Michalewicz, 2017) were also confirmed. The analysis of convergence to a fixed point was conducted, from a different point of view .in (Bonyadi and Michalewicz, 2017). It was justified that the expectation of the position of each particle converges to a point (first-order stability analysis) between the personal and global best vector $\frac{\varphi_1 g + \varphi_2 p}{\varphi_1 + \varphi_2}$ if

and only if $0 \le \omega < 1$ and $0 < \varphi < 4\left(1+\omega\right)$ where $\varphi = \varphi_1 + \varphi_2$. However, it was pointed out that the first-order stability is not enough to guarantee convergence of particles and second-order stability should be also guaranteed. In fact, to guarantee stability, not only the expected position of particles should converge to a fixed value but also the variance of the position should converge to 0 (Bonyadi and Michalewicz, 2015b and Cleghorn and Engelbrecht, 2015). It was presented in (Bonyadi and Michalewicz, 2017) that, setting the coefficients to satisfy

$$\frac{5\varphi - \sqrt{25\varphi^2 - 336\varphi 576}}{24} < \omega < \frac{5\varphi + \sqrt{25\varphi^2 - 336\varphi 576}}{24},$$

where $\varphi = \varphi_1 = \varphi_2$ guarantees the convergence of the variance of the position of particles to a fixed value (a necessary condition for variance of position to converge). Note that this inequality can be simplified to $\varphi < \frac{12\left(\omega^2 - 1\right)}{5\omega}$.

This analysis was slightly modified due to a small error and the corrected version was presented in (Jiang et al, 2007a). This study also recommended coefficients to be restricted $\omega = 0.715$ and $\varphi_1 = \varphi_2 = 1.7$ value for experimental purposes. The authors argued that these coefficient values result in a higher variance during the run that improves the exploration ability of particles. The expected value and the standard deviation of the sequence of generated positions by SPSO were also analyzed in (Cleghorn and Engelbrecht, 2015), and the results found in (Jiang et.al., 2007a) were confirmed. It was checked that convergence of variance to a fixed point (a necessary condition for second-order stability) requires $\varphi < \frac{12\left(\omega^2 - 1\right)}{5\omega - 7}$,where $\varphi = \varphi_1 = \varphi_2$, that is, the same as what was found in (Jiang et al., 2007a). In addition, in (Cleghorn and Engelbrecht, 2015) proved that the variance of positions converges to $h\left(\varphi_1, \varphi_2, \omega\right)$|g - p|, where $h\left(.,.,.\right)$ is a function of inertia weight and acceleration coefficients. Hence, if $h\left(\varphi_1, \varphi_1, \omega\right) \neq 0$ is guaranteed, then particles do not stop moving (non-zero variance) until $p = g$. Therefore, the algorithm is second-order stable only if $p = g$. More recently, in 2018, authors were able to obtain the region defined by $\varphi < \frac{24\left(\omega^2 - 1\right)}{5\omega - 7}$ using only the weak stagnation assumption. The work in (Cleghorn and Engelbrecht, 2015), also implies that the convergence region of $\varphi < \frac{24\left(\omega^2 - 1\right)}{5\omega - 7}$ for $\omega \epsilon \left[-1,1\right]$ the same irrespective of the social network topology utilized by canonical PSO (CPSO).

The first and second-order stability were investigated in (Garcıa-Gonzalo and Fernandez-Martınez, 2014), for SPSO when a generic distribution for ω and $\left(\varphi_1, \varphi_2\right)$ was considered, i.e.; it was assumed that ω is a random variable from an arbitrary probability distribution with the expected value μ_ω and the variance α_ω^2 and φ_1 and φ_2 (ϕ_1 replaces $\varphi_1 R_{1t}^i$. and ϕ_2 replaces φ_2 R_{2t}^i in Eq.6) are also random variables from an arbitrary probability distribution with the expected values μ_{ϕ_1} and μ_{ϕ_2} and the variances $\alpha_{\phi_1}^2$ and $\alpha_{\phi_2}^2$ respectively. It was proven that SPSO is first-order stable if and only if -1 < μ_ω < 1 and $0 < \mu_\varphi < 2\left(\mu_\omega + 1\right)$, where $\varphi = \varphi_1 + \varphi_2$. *Furthermore*, a necessary condition

for the second-order stability is that $-a < \mu_\omega < a$ and $0 < \mu_\varphi < b$, where $a = \frac{1}{\sqrt{c^2+1}}$, $b = \frac{2\left(1-\left(1+c\right)\mu_\omega^{\,2}\right)}{1+d^{2\left(d^2-1\right)\mu_\omega}}$, $c = \frac{\sigma_\omega}{\mu_\omega}$, and $d = \frac{\sigma_\varphi}{\mu_\varphi}$.

It was shown that the regions found to guarantee second-order stability are embedded within the regions that guarantee first-order stability, i.e.; if a SPSO is second-order stable, then it is also first-order stable (Bonyadi and Michalewicz, 2015b).

Recently, some studies have considered more general assumptions to find the convergence boundaries under more realistic conditions. For example, (Cleghorn and Engelbrecht, 2014a), assumed that the personal best of particles and the global best of the swarm are allowed to move and can occupy an arbitrarily large finite number of unique positions and it also showed that the topology does not affect the convergence boundaries, but it might affect the speed of convergence / divergence. The main finding of that study was that SPSO is first-order stable if and only if $-1 < \omega < 1$ and $0 < \varphi < 4\left(1+\omega\right)$ where $\varphi = \varphi_1 + \varphi_2 < 4$ under this more general assumption. This is indeed the same as what was found that in (Bonyadi and Michalewicz, 2017) and other previous studies through first-order stability analysis.

The second-order stability of SPSO was also investigated in (Liu, 2015). It was assumed that the personal best needs to remain unchanged at least for a limited number of iterations (3 in that study), that is, a weaker assumption comparing to what was assumed by previous studies (i.e., the personal best need to always remain constant). The theoretical analysis in that paper proved that the convergence boundaries found in (Gong et. al., 2017*)* are valid under this assumption for the global best particle. Moreover, in (Liu, 2015) assured that the found regions serve as a necessary and sufficient condition for second-order stability of the global best particle. This study recommended $\omega = 0.42$ and $\varphi_1 = \varphi_2 = 1.55$ based on experiments on an extensive set of benchmark functions.

Analysis of stability was also done for a variant of PSO called " Standard PSO 2011", SPSO2011, in . The velocity update rule for SPSO2011 is written as:

$$\vec{V}_{t+1}^{\,i} = \omega\vec{v}_t^{\,i} + H_i\left(\vec{G}_t^{\,i}, \left|\left|\vec{G}_t^{\,i} - \vec{x}_t^{\,i}\right|\right|\right) - \vec{x}_t^{\,i} \tag{10}$$

where $H_i\left(\vec{G}_t^{\,i}, \left|\left|\vec{G}_t^{\,i} - \vec{x}_t^{\,i}\right|\right|\right)$ is a hyper-spherical distribution with the center $\vec{G}_t^{\,i}$ and radius $\left|\left|\vec{G}_t^{\,i} - \vec{x}_t^{\,i}\right|\right|$ given that $\vec{G}_t^{\,i} = \frac{L_t^i + \vec{p}_t^{\,i} + \vec{x}_t^{\,i}}{3}$, $\vec{P}_t^{\,i} = \vec{x}_t^{\,i} + \varphi_1\left(\vec{L}_t^{\,i} - \vec{x}_t^{\,i}\right)$ and $\vec{L}_t^{\,i} = \vec{x}_t^{\,i} + \varphi_2\left(\vec{l}_t^{\,i} - \vec{x}_t^{\,i}\right)$ ($\vec{l}_t^{\,i}$ is the best personal best among the particles in $\vec{T}_t^{\,i}$) (Bonyadi and Michalewicz, 2014d). The analyses of convergence conducted for previous PSO variants are not applicable to SPSO2011. The reason is that updating velocities and positions in all previous PSO variants were done dimension by dimension which enabled researchers to study particles in a one-dimensional space. However, the calculation of $\vec{V}_t^{\,i}$ in SPSO2011 involves generating random points using a hyperspherical distribution with a variance that is dependent on the distance between $\vec{G}_t^{\,i}$ and $\vec{x}_t^{\,i}$. In order to repeat the dimension-wise analysis in SPSO2011 one needs to decompose the random point generation procedure done by the hyperspherical distribution into dimension-wise calculations which might need further effort. Hence, the stability analysis for SPSO2011 was done experimentally in (Bonyadi and Michalewicz, 2014d and 2015a) where it was shown that convergence boundaries for particles in SPSO2011 are dependent on the number of dimensions and these boundaries are different from that of other PSO variants (e.g., SPSO). Thus, good choices for coefficient values in previous PSO variants do not

necessarily lead to a good performance of SPSO2011. In (Bonyadi and Michalewicz,2015a), SPSO2011 also experimentally showed that the convergence boundaries that guarantee second-order stability are not affected even if the global best and personal bests are updated (through a uniform random distribution) during the runs for both SPSO and SPSO2011.

Stability of a stochastic recurrence relation that formulates the position update rule of particles in a wide range of PSO variants (including SPSO) was studied in (Bonyadi and Michalewicz, 2015b). In order to weaken the stagnation assumption, it was assumed that the global and personal bests in that relation are updated through an arbitrary random distribution. It was proven that the necessary and sufficient conditions to guarantee convergence of expectation and variance of generated positions by that relation are independent of the mean and variance of the random distribution by which the global and personal bests are updated.

3.1.2 Configurations of Movement

If the coefficients ω, φ_1 and φ_2 of SPSO are selected in a way that they are inside the convergence boundaries, then the sequence of the positions of particles is not divergent. During the run, however, particles oscillate with different configurations around their equilibrium point until they converge (Bonyadi and Michalewicz,2015a, 2017). The difference between these configurations is a consequence of picking different values for coefficients that plays important role on the performance of the algorithm. For example, a particle that moves smoothly in the search space can potentially be more effective in exploitation phase than a particle that jumps all over the search space. Hence, investigation of these configurations and calculation of corresponding coefficients which exhibit different configurations can be helpful for practitioners. In addition, the speed of convergence, i.e.; how fast the particles positions approach the equilibrium point, can be of importance to determine appropriate coefficient values.

The roots of the characteristic equation of the expected position of particles for each dimension to categorize different oscillations particles positions might exhibit in Eq.11, which was mentioned in (Bonyadi and Michalewicz, 2017).

$$E\left(x_{t+1}\right)+\left(\varphi-\omega-1\right)E\left(x_t\right)+\omega E\left(x_{t-1}\right)=\varphi P \tag{11}$$

where $E\left(.\right)$ is the expectation operator, $\varphi=\frac{\varphi_1+\varphi_2}{2}$, $P=\frac{\varphi_1 g+\varphi_2 p}{\varphi_1+\varphi_2}$. The configuration of generated points by this equation (i.e.; expectation of positions) are:

- **Harmonic** if the imaginary component of both roots of the characteristic equation are non-zero and the real component of the roots are positive,
- **Zigzagging** if the imaginary component of both roots of the characteristic equation are zero and the real component of at least one of the roots is negative,
- **Harmonic–Zigzagging** if the imaginary component of both roots of the characteristic equation are non-zero and the real component of both roots is negative,
- **Non–oscillatory** if the imaginary component of both roots of the characteristic equation are zero and the real component of both roots is positive,

Moreover, experiments in (Bonyadi and Michalewicz, 2017) showed that convergence is faster if the values of ω and φ are closer to the "center" (close to the point (0, 1)) of the convergence boundaries triangle . Thus, particles spend more iterations for exploitation with this setting. Similar analysis for the behavior of particles before convergence was conducted in (Campana et al., 2010) where the same oscillation configurations were observed.

The speed of convergence of the expectation of positions was investigated in (Tong et.al., 2019), where it was proven that the speed of convergence is directly related to $c = \max\left\{\left|\left|\gamma_1\right|\right|, \left|\left|\gamma_2\right|\right|\right\}$ where

$$\gamma_1 = \frac{1+\omega-\varphi_1-\varphi_2+\alpha}{2},\ \gamma_2 = \frac{1+\omega-\varphi_1-\varphi_2-\alpha}{2}, \text{ and } \alpha = \sqrt{\left(1+\omega-\varphi_1-\varphi_2\right)^2 - 4\omega}$$

i.e.; the larger the value of *c* is, the faster the expectation of the position of the particle converges to its equilibrium. It was also assured that the expected value of the particle's positions converge to a fixed point (namely, $P = \frac{\varphi_1 g + \varphi_2 p}{\varphi_1 + \varphi_2}$) if and only if $c < 1$.

In (Bonyadi and Michalewicz, 2015a), they investigated the behavior of particles before convergence through an experimental approach. They considered the generated sequence of particles positions as a time series and they used frequency domain analysis to understand how particles oscillate during the run. They categorized the oscillation patterns into four groups based on the maximum frequency of oscillation (low, low-mid, mid-high, and high frequencies), corresponding to the patterns introduced in (Bonyadi and Michalewicz, 2017). They showed that the results found by their experimental approach is very similar to what has found in (Bonyadi and Michalewicz, 2017).

Hence, they used the same approach to analyze the oscillation of particles in SPSO2011. They found that the boundaries of coefficients corresponding to different patterns of oscillation for SPSO2011 are different from those of SPSO. Their experiments also showed that these boundaries are not sensitive to the number of dimensions.

3.1.3 Convergence to a Local Optimum

There has been a significant amount of research effort devoted to the theoretical study of PSO convergence (Harrison et al.,2017). Convergence to a point ensures that a particle settles to its equilibrium and does not move to infinity. If acceleration and inertia weight are in the convergence boundaries it was proven that all particles converge to $\vec{x} = \vec{p}_t^{\,i} = \vec{g}_t$ and $\vec{V}_t^{\,i} = 0$. (stagnation). However, for any value of coefficients, there is no guarantee that the point that the particles converged to is a local optimum. We can alternatively use the following two conditions for a particle is said to be locally convergent

$$\forall \varepsilon > 0, \lim_{t\to\infty} pr\left(\left|\vec{x} - \vec{X}\right|\right) \geq 1-\varepsilon \tag{12}$$

we say that $\vec{p}_t^{\,i}$ converges to $\vec{X}$ with probability 1,

$$\forall \varepsilon > 0, \lim_{t\to\infty} pr\left(\left|\vec{g}_t - \vec{X}\right|\right) \geq 1-\varepsilon \tag{13}$$

where $\vec{X}$ is a local optimum and *Pr* denotes the probability in (Bonyadi and Michalewicz, 2017 and, Fan and Yan, 2014). Local convergence for PSO variant was investigated in (Zou et.al., 2015) in particular for SPSO. In this paper, the swarm size issue is discussed because it was observed that SPSO does not perform well when the swarm size is too small. The explanation given in (Bonyadi and Michalewicz, 2017 and Mubeen and Dr. Dhananjay, 2022) was that particles in a SPSO with smaller swarm size (e.g., 2) have larger probability to stagnate. A variant of SPSO called "Guaranteed

Convergent Particle Swarm Optimization", GCPSO (Zou et.al, 2015) was proposed in which the local convergence issue was addressed. In GCPSO, a new velocity update rule was introduced for the global best particle τ_t (Bonyadi and Michalewicz, 2017 and Zou et .al., 2015)

$$\vec{V}_{t+1}^{i} = \begin{cases} -\vec{x}_t^i + \vec{g}_t + \omega\vec{V}_t^i + \vec{\rho} & if\ i = \tau_t \\ \omega\vec{V}_t^i + \varphi_1\mathbb{R}_{1t}^i\left(\vec{p}_t^i - \vec{x}_t^i\right) + \varphi_2\mathbb{R}_{2t}^i\left(\vec{g}_t - \vec{x}_t^i\right), otherwise \end{cases} \tag{14}$$

where ρ^j (the value of the j^{th} dimension of $\vec{\rho}$) is a random value determined through an adaptive approach that applies a perturbation to the velocity vector with $\vec{\rho}>0$. At each iteration of GCPSO, a random location in a hyper-rectangle with the j^{th} side length equal to ρ^j is generated. This random location is moved from $\vec{g}_t$ by the vector $\omega\vec{V}_t^i$ to form the new position for the particle τ_t (global best particle). In GCPSO, $\vec{x}_{t+1}^{\tau_t}$ is taken randomly from the gray area while all other particles follow the original position and velocity update rule (Bonyadi and Michalewicz, 2017). The velocity update rule for the particle τ_t in GCPSO forces that particle to always move (implying $\vec{V}_{t+1}^{\tau_t} \neq 0$ for any t). It was proven that in (Zou et.al., 2015) SPSO is not locally convergent while GCPSO is locally convergent and it could provide acceptable results for the number of particles was set to 2. The local convergence issue for SPSO was also discussed widely in (Bonyadi and Michalewicz, 2017 and Schmitt and Wanka, 2013) where it was proven that SPSO is locally convergent for $n=1$ problems; however, it is not locally convergent when $n>1$.

3.1.4 Expected First Hitting Time (EFHT)

To characterize optimization ability of algorithms, we suggest the expected first hitting time (EFHT), i.e.; the time until a search point in the vicinity of the optimum is visited. Witt (2009). argued that most convergence analyses for PSO and PSO variant and impacts of coefficients on movement patterns in the particle swarm optimization algorithm, SPSO including (Bonyadi and Michalewicz, 2017) have been limited to the concept of convergence to an attractor ($\vec{p}_t^i$ or $\vec{g}_t$) and not to a particular optimum. The EFHT (run time) analysis was conducted on the variant GCPSO because at the moment GCPSO was the only PSO variant that was assured to be locally convergent. Analyses showed that the EFHTof GCPSO on the sphere function is asymptotically the same as a simple (1+1) ES–see (Jagerskupper, J., 2008) for the analysis of the EFHT for randomized direct search methods with isotropic sampling, including (1+1) ES. However, performance GCPSO is substantially changed by rotating the search space, and GCPSO is not rotation invariant while (1+1) ES is.

The analysis of EFHT for SPSO was also conducted in (Lehre and Witt, 2013), where the algorithm was applied to some function. It was proven that EFHT for SPSO is potentially infinite even when it is applied to a one-dimensional sphere. It was shown that, in some situations that occur with non-zero probability, the algorithm cannot locate the optimal solution even for a one-dimensional sphere. Note that the setting of the parameters in that paper was different from that of (Schmitt and Wanka, 2013), where it was proven that SPSO is locally convergent for one-dimensional problems.

$$\vec{V}_{t+1}^{i} = \omega\vec{V}_t^i + \varphi_1 R_{1t}^i\left(\vec{p}_t^i - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i\left(\vec{g}_t - \vec{x}_t^i\right) + \Delta_t^i,$$

where the extra noise term Δ_t^i has uniform distribution on the interval $\left[-\frac{\delta}{2}, \frac{\delta}{2}\right]$. A new variant of PSO was introduced in (Lehre and Witt, 2013), called "Noisy Particle Swarm Optimization", NPSO,

in which a uniform randomly generated vector in a hyper-rectangle with the side length $\delta>0$ was added to the velocity vector of each particle. The authors proved that EFHT for NPSO to optimize a sphere function is finite, i.e.; NPSO converges to a local optimum of the sphere function in a finite number of function evaluations. (Ganesh et.al., 2021 and Mubeen and Dr. Dhananjay, 2022) presented an adaptive PSO algorithm that could automatically track changes in the dynamic system, and several environment detection and response strategies were tested on the parabolic benchmark function.It was concluded that more efforts are needed to study the EFHT of SPSO and understand how the algorithm works. A similar analysis was also done in (Bonyadi and Michalewicz., 2014d), for SPSO2011, where it was proven that, with specific setting of parameters, EFHT of the algorithm for any one-dimensional optimization function is infinite, e.g., SPSO2011 does not guarantee to converge to a local optimum under assumptions that take place with non-zero probability.

3.2 Limitations Related to Transformation Invariance

The algorithm $\mathcal{A}$ is invariant under the transformation $T, T:\mathbb{R}^d \to \mathbb{R}^d$, if the performance of $\mathcal{A}$ on any objective function $F\left(\vec{x}\right), F:\mathbb{R}^d \to \mathbb{R}$, is the same as the performance of $\mathcal{A}$ on $F\left(T\left(\vec{x}\right)\right)$, given that the initial state is " properly " set in both cases (Bonyadi and Michalewicz, 2017). For stochastic algorithms, this equality needs to maintain almost surely or in probability distribution. Formally, this definition means that the performance of the algorithm is independent of how the coordinate axes are placed on the search space.

In addition to invariance, it is also expected that the optimization algorithm is as independent as possible from the initial state (so called adaptivity). Therefore, in order to study the transformation invariance property of optimization algorithms, some researchers took the adaptivity for granted and assumed that the initial state is also transformed using the same transformation that the search space is transformed with (*i.e.;* T), (Netjinda et.al., 2015 and Tong et. al., 2019).

Transformation invariance is an important characteristic of an optimization algorithm. If an algorithm is invariant of a transformation T, then, by definition, the performance of the algorithm on a problem P can be generalized to the set of problems $C\left(P \in C\right)$ that are defined by $\left(P;T\right)$. This enables researchers to make stronger statements, favorable or unfavorable, about the performance of an algorithm. Rotation, scaling, and translation are well-known transformations that are frequently used in different areas.

Hence, it is valuable to understand if an algorithm is invariant under these transformations (Bonya and Michalewicz, 2014a, 2015a). One can define the transformation T ($\vec{x}$) by $T\left(\vec{x}\right) = sM\vec{x} + \vec{b}$ to formulate these transformations (rotation, scaling, and translation) where $s \in \mathbb{R}$ is a scale factor and $s \neq 0$, $M \in \left(\mathbb{R}^d x \mathbb{R}^d\right)$ is a rotation matrix, and $\vec{b} \in \mathbb{R}^d$ is a translation vector.

Transformation invariance has been investigated for many optimization algorithms (Bonyadi and Michalewicz, 2017). Among the transformation of rotation, scale, and translation, rotation has received most attention by researchers in the field of PSO for its invariance and variance properties. Probably the main reason for this special attention is that the rotation of the search space potentially makes an objective function non-separable (Tian and Shi, 2018 and Tong et.al., 2019). Moreover, dealing with non-separable optimization functions is of more interest as there are many optimization problems with this characteristic. However, a PSO algorithm is rotation variant if its performance is change by rotating the search space. Translation invariance is probably the most basic characteristic among these transformations that an optimization algorithm must have. The reason is that, if an algorithm is sensitive to translation, the algorithm in fact is " making an assumption " about the location of the optimal solution of the problem at hand. Hence, if the optimal solution is shifted, that assumption is not valid anymore and this, in turn, causes a change to the performance of the algorithm.

Transformation invariance of SPSO was first analyzed in 2007, where it was shown that in (Yang et. al, 2015) SPSO is scale and translation invariant but is rotation variant. The reason that

SPSO is rotation variant is that the effect of random diagonal matrices on the direction of movement is sensitive to the rotation of the coordinate system. In contrast, it was proven that " Linear particle swarm optimization ", LPSO (Bonyadi and Michalewicz, 2017) is rotation, scale, and translation invariant. However, LPSO suffers from a limitation called line search.

In LPSO, if particle *i* oscillates between its personal best and the global best and it is not able to search in other directions, then $CI\Big|\Big|SI$ and $\vec{V}_t^i\Big|\Big|CI$ hold (Bonyadi and Michalewicz, 2015a and Bonyadi et.al,2014a). A new PSO variant which was rotation invariant while it did not have the line search limitation called " Rotation invariant Particle Swarm Optimization ", RPSO and used random rotation matrices rather than random diagonal matrices to perturb the direction of movement in each iteration. A method called exponential map was used to generate the rotation matrices. The idea of exponential map is that: the exponential of any skew-symmetric matrix $S\theta$, where S is a dxd matrix and θ is a scale factor, is a rotation matrix if $\left(S\theta\right)^T=\left(S\theta\right)^{-1}$ and $det\left(S\theta\right)=1$ with the rotation angle θ (Liu, 2015).The exponential of a matrix $S\theta$ is defined by:

$$Q=\sum_{i=1}^{maxC}\left(\frac{1}{t!}\left(S\theta\right)^i\right)+I \tag{15}$$

where S is a dxd skew-symmetric matrix, I is the identity matrix, θ is a scalar, and $maxC\rightarrow\infty$. In order to generate a random rotation matrix, one can generate a random S as $S=\left(P-P^T\right)$, where P is a dxd matrix with elements generated randomly in the interval $\left[-0.5,0.5\right]$. Moreover, the angle of rotation in degrees is α (a real scalar) where $\theta=\frac{\alpha\pi}{180}$.

It is clear in, Eqn.14, that for a finite value for $maxC$, the matrix Q becomes an "approximation" of a rotation matrix with the rotation angle α (see (Mohammad Reza Bonyadi, 2014) for details on how the truncation error is calculated). (Witt, 2009) set the value of $maxC$ to 1, limiting the approximation to one term of the summation only, i.e.; $Q=I+S\theta$, so that the time complexity of generating the rotation is reduced. Also, they claimed that, as the rotation is only considered for small values of α, thisapproximation does not impact the overall results. The time complexity for generating and multiplying the approximated rotation matrix (with $maxC=1$) into a vector is in $O\left(d^2\right)$, including the cost of generating the approximated matrix $\left(O\left(d^2\right)\right)$ plus vector matrix multiplication $\left(O\left(d^2\right)\right)$. For larger values of $maxC$, the order of complexity of generating Q become larger .

A method for generating accurate random rotation matrices was proposed and investigated in (Bonyadi et.al., 2014b) further by the concept of replacing random diagonal matrices by rotation matrices. The method was based on Euclidean rotation matrices that could be used to rotate vectors in any combination of hyper planes. The paper tried to include two techniques for rotation of the velocity vector were investigated: rotation within complexity in $O\left(d^2\right)$ and rotation within complexity in $O\left(d\right)$. As empirical results in terms of the objective value showed that rotation in $O\left(d\right)$ is just slightly worse than rotation in $O\left(d^2\right)$, however, it is much faster. The random diagonal matrices were replaced by rotation matrices in $O\left(d^2\right)$ in many PSO variants proposed in (Bonyadi et.al., 2014a and Zou et.al., 2015). Results based on tested benchmark functions showed that PSO variants usually benefit from random rotation matrices to solve tested functions that included separable and non-separable test problems.

The PSO invariant (SPSO) was applied to several optimization problems when their search space were rotated (Bonyadi and Michalewicz,2014a, 2015a), to test if the performance of the algorithm changed. Results of applying SPSO to the existing problems were compared with other variants such as "Covariance Matrix Adaptation ES", CMA–ES, and " Differential Evolution", DE in (Helwig et.al., 2013). In addition, the performance of the algorithm changes indealing with ill -conditioned functions when the search space of the problem is rotated which indicates that the algorithm is rotation variant. It was found that the performance of SPSO is almost independent of the condition number for ill-conditioned separable optimization functions and SPSO outperforms CMA-ES on such function. However, the performance of SPSO drops almost linearly with the condition number when it is applied to ill-conditioned non-separable functions with condition number larger than 100. Experiments showed that CMA-ES outperforms SPSO in optimizing such functions. A similar comparison was also conducted in (Lu et.al., 2015a) where the findings presented in (Bonyadi and Michalewicz, 2017) related to the good performance of SPSO for separable optimization problems and its poor performance in non-separable optimization problems were confirmed. In addition, the authors argued that the main reason behind this poor performance is that all calculations in SPSO are performed for each dimension separately, which makes the algorithm rotationally variant (Bonyadi and Michalewicz, 2017). The researchers used eight numerical benchmarking functions that represent diverse aspects of typical issues, as well as a real-world application involving data clustering, to test the Starling PSO (Lu et.al, 2015a). The experimental results indicated that the Starling PSO outperformed the SPSO and produced the best solution in several numerical benchmarking functions and the majority of real-world problems in the clustering studies.

It was shown in (Bonyadi and Michalewicz, 2017) that not only SPSO is rotation variant, but also particles in SPSO are biased towards the lines parallel to the coordinate axes. In fact, by tracking the positions of particles in the search space during the optimization process, it was found that particles tend to sample more points along the lines parallel to coordinate axes than other points in the search space, i.e., particles positions are "biased" to the lines parallel to the coordinate axes. The authors used these two issues (rotation variant and bias) to design test problems that are hard or easy for SPSO to optimize.

The combination of local convergence and rotation invariance in PSO algorithm for the variant proposed in (Bonyadi and Michalewicz, 2017) abbreviated as LcRiPSO. It was marked that LcRiPSO is rotationally invariant and locally convergent if the operator m is invariant under any rotation and it satisfies the condition formulated in $\forall y \in S, \exists Ay \subseteq S$, such that $\forall \vec{z} \in Ay, \forall \delta > 0, P(\left|m\left(\vec{y}\right) - \vec{z}\right| < \delta) > 0$. It was proven that if the operator *m* generates a new point following the normal distribution then it satisfies conditions for local convergence and rotation invariance of the algorithm. The time complexity of LcRiPSO using the normal distribution for Q is in $O\left(d\right)$, that is faster than the variants proposed in (Bonyadi et.al., 2014b and Bonyadi and Michalewicz, 2017).

4. MODIFICATIONS OF CONVERGENCE PSO TO DEAL WITH UNCONSTRAINED OPTIMIZATION PROBLEMS

In view of the problems mentioned above, this section review an improved convergence particle swarm optimization algorithm with random sampling of control parameters (SC-PSO), and the main contribution of the present work is delineated as follows. Three different categories of approaches were considered for improving convergence PSO algorithm:

- Setting parameters: It refers to setting the topology, coefficients and population size.
- Modifying components of the algorithm: modifying components refers to changes of the velocity / position update rule (including adding new components; modifying the way they are calculated).
- Fusing the algorithms: fusing the algorithms refers to hybridization of PSO with other methods.

We review convergence of PSO variants which have improved the performance of SPSO through specific settings of its parameters, modification of the velocity/position update rules, and hybridization of the algorithm in the following subsection subsequently.

4.1 Parameters Setting

For basic PSO, parameters have great impact on the performance of algorithm. If they are assigned inappropriately, the trajectories of particles cannot converge and may even be unstable, which will cause that the optimal solution of optimization problems cannot be found. Hence parameter setting has been a problem in iterative optimization methods in the past years.Two different approaches for setting parameters of an evolutionary algorithm might be belonged to:

- Parameter tuning: it comes to setting parameters of an algorithm through experiments to some constant values.
- Parameter control: it, however, comes to design of a strategy which changes the value of parameters during the run.

At present, the control parameters are usually chosen according to the experience or experiments from engineers, so it is not flexible and the exploration ability of convergence of PSO has also been greatly restricted. Parameter control approaches are categorized further into the following main groups up to now: deterministic, adaptive, and self-adaptive.

- In deterministic parameter control approaches a rule (called time-varying rule) is designed to calculate the value of a parameter based on the iteration number.
- In the adaptive parameter control approaches, a function is designed that maps some feedback from the search into the value of the parameter.
- In a self-adaptive parameter control approach the parameters are encoded into individuals and are modified during the run by the optimization algorithm (Bonyadi and Michalewicz, 2017).

We reviewed articles that have studied different parameters for convergence PSO (i.e., topology, coefficients, and population size).

4.1.1 Topology

In the classical PSO, the particles share information via the global attractor, which is the best solution any particle has found so far and which is known to the whole swarm. That means that every particle interacts with every other member of the swarm. The network of individuals in a swarm suggests the same concept as the topology in PSO variants does. The idea was that, topology should affect the explorative and exploitative behavior of the swarm as different topologies impose different speed of propagation of information among particles. Five different topologies in PSO was conducted and were tested, all these topologies on a benchmark of four functions (Cleghorn and Engelbrecht, 2015).

1. Circle (local best /ring) topology, in which every particle *i* is a neighbor of the particles $i-k,\ldots,i+k$. Especially for small k , the local best topology delays the information distribution among the swarm considerably. This results in highest explorative behavior.

Symbolically: $T_t^i = \left\{i, \psi_i\right\}$ where $\psi_i = \underset{l=\left\{i-1,i,i+1\right\}}{\arg\min}\left(F\left(\vec{p}_t^{\,l}\right)\right)$

2. Wheels topology, in which one specific particle (n_o) is adjacent to every other particle. Particle (n_0) acts as a kind of guardian to slow down the distribution of information.

Symbolically: $T_t^i = \left\{i, \tau_i\right\}$ where $\tau_t = \underset{l=\left\{1,2\ldots,n\right\}}{\arg\min}\left(F\left(\vec{p}_t^{\,l}\right)\right)$ and $T_t^{i\neq 1} = \left\{1\right\}$

3. Fully connected (Star/global best) topology, in which any two particles are neighbors.

This topology allows the fastest distribution of information. This results exhibit the highest exploitative behavior.

Symbolically: $T_t^i = \left\{i, \tau_i\right\}$ where $\tau_i = \underset{l=\left\{1,2\ldots,n\right\}}{\operatorname{argmin}}\left\{F\left(\vec{p}_t^{\,l}\right)\right\}$

4. Grid (von Neumann) topology, in which the particles are arranged on a 2-dimensional grid with wraparound edges, such that every particle has exactly 4 neighbors. This topology is seen as a compromise between the fully connected swarm and the ring topology (Cleghorn and Engelbrecht, 2015).
5. Random edge topology, instead of choosing one of the above fixed topologies, a random neighborhood graph is generated according to some distribution.

Symbolically: $T_t^i = \left\{i, j\right\}$ where j is randomly selected among other individuals.

In some PSO variants, the topologies occur not in the described pureforms but as a mixture. E.g., in (Lehre and Witt, 2013), the author uses a ring topology with additional randomly sampled shortcuts. There has been many research on comparing the effects of the different topologies on the quality of the optimization in (Cleghorn and Engelbrecht, 2015). While in all the variants mentioned until now only one member of the neighborhood was actually chosen for the velocity update equation, there have been some attempts to provide particles with knowledge not only from the best but from every neighbor. This idea leads to the "fully informed particle swarm", FIPS (Cleghorn and Engelbrecht, 2015). Instead of selecting one particular neighbor for the local guide, the mean of the local attractors of every neighbor is calculated for updating velocity. This finding was also investigated in(Cleghorn and Engelbrecht, 2015) later in (Liu et.al., 2016), and different topologies were tested under that formulation. All these studies, however, suffer from the lack of proper statistical analysis of the results or an adequate number test cases that affect the generality of the conclusions drawn (Bonyadi and Michalewicz, 2017).

4.1.2 Coefficients

A standard method for improving convergence of PSO performance is parameter adaptation. Instead of assigning fixed values to the coefficients (swarm parameters). The performance of SPSO is affected by tuning the values of its swarm parameters from Eqn.6. Thus, researchers in (Bonyadi et.al, 2014b and Tanweer et.al., 2016) were made several attempts to tune the values of these coefficients for a set of benchmark functions. The general idea of parameter adaptation mechanisms is that during the early iterations, the swarm should explore larger areas while in the end of the optimization process,

the particles are supposed to converge towards one common point. In order to support this behavior, several parameter adaptation mechanisms have been developed (Zhang et al., 2015).

The authors could experimentally show that the performance of PSO significantly depends on the best values for the ω lying in $\left[0.4, 0.9\right]$, while the interval $\left[1.5, 3\right]$ holds the best value for φ_1 and φ_2. The standard swarm parameters for the experiments of most papers are $\omega = 0.72984$, $c_1 = c_2 = 1.496172$.

Figure 1 depicts the convergence curve of PSO with inertia weight decreased linearly from 0.9 to 0.4 during the run. Main disadvantage, once the inertia weight is decreased, the swarm loses its ability to search new areas.

Figure 2 depicts the convergence of PSO with inertia weight increased linearly from 0.4 to 0.9 during the run. From Figure 2, it is seen that, there is no improvement in its best value even after 1000 iterations

We use this parameters for every experiment unless the ones in which different parameters are compared and the choices are explicitly stated. By adjusting the parameters, it is possible to influence the trade-off between exploration, the capability to search in areas that have not been visited before, and exploitation, the capability to refine already good search points, by a linear decreasing inertia weight (ω decreased linearly from 0.9 to 0.4 during the run) was tested on one test problem–this variant was called "Decreasing Inertia Particle Swarm Optimization", DIPSO. The latter (based on the average of the objective value) tested an opposite approach, increasing the inertia weight during the run from 0.4 to 0.9, and they found that this idea actually works better than decreasing inertia weight on some test cases (Bonyadi and Michalewicz, 2017).

The concept of decreasing inertia weight was extended in (Cleghorn and Engelbrecht, 2014a) in a way that a non-linear function was used to decrease the value of the inertia weight:

Figure 1. PSO for ω runs $\left[0.9, 0.4\right]$

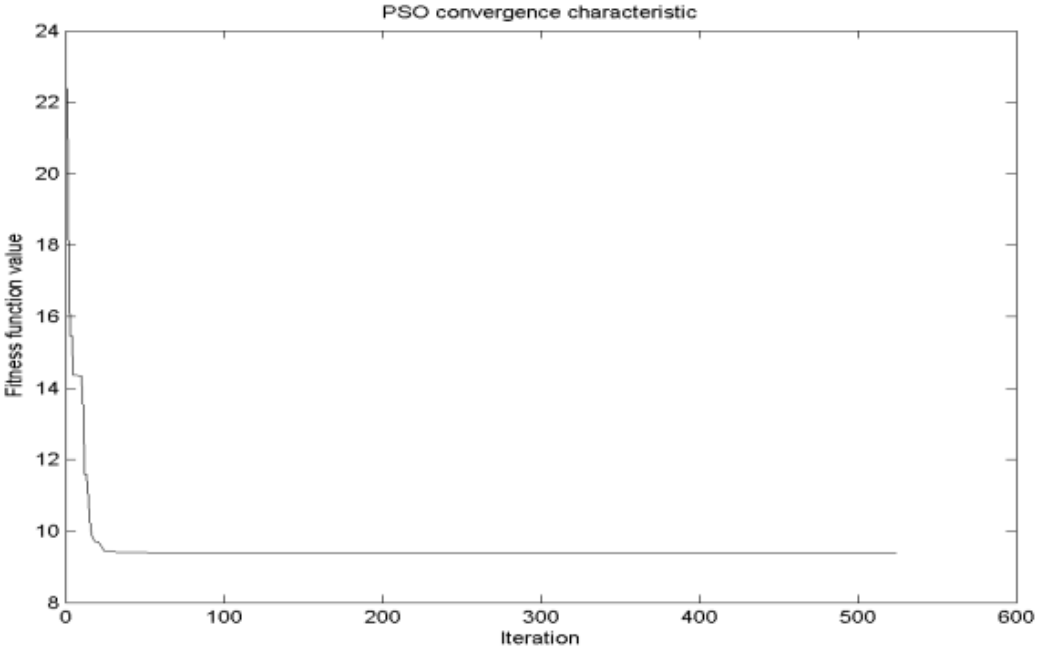
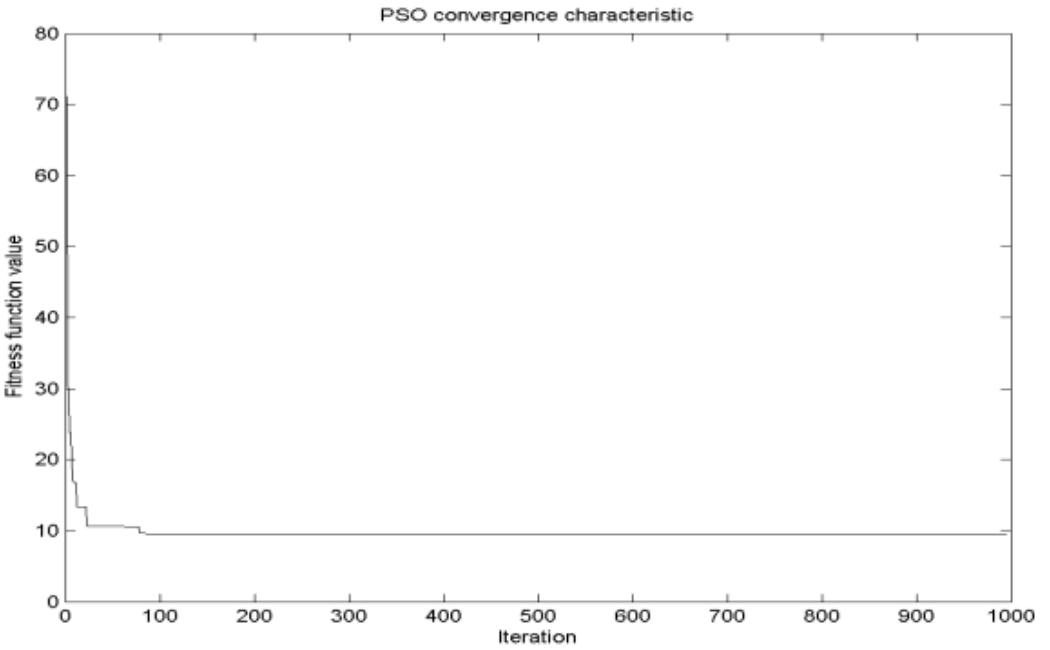


Figure 2. PSO for ω runs $[0.4, 0.9]$



$$\omega_i = \frac{\left(maxi - i\right)^n}{maxi^n}\left(\omega_s - \omega_e\right) + \omega_e ,$$

where ω_s and ω_e were starting and final inertia weights, and $maxi$ was the maximum allowed iterations. With n=1 , this formulation is exactly the same as that in DIPSO. After experimenting with the sphere function, the parameters of this variant were set to: $\omega_s = 0.2, \omega_e = -0.3, n = 1.2,$ and $maxi = 2,000$. This method with the above settings was compared (based on the average of the quality of found solutions) with other optimization methods such as a genetic algorithm and a differential evolution on 17 benchmark problems.

Although inertia weight influences exploration and exploitation behavior of the algorithm, acceleration coefficients also play an important role in this regard. A time-varying approach was proposed in (Mubeen and Dr. Dhananjay, 2022) where the acceleration coefficients were changed during the run (the method was called " Time Varying Acceleration Coefficient Particle Swarm Optimization", PSO-TVAC). In that method, the value of φ_1 (cognitive weight) decreased while the value of φ_2 (social weight) increased during the run of the algorithm. Thus, particles tend to conduct exploration at the beginning and exploitation at the latter stages of the optimization process as experiments confirmed potential benefits of time-varying approaches on some test functions (Bonyadi and Michalewicz, 2017).

Figure 3 depicts the convergence curve of PSO with inertia weight. SPSO, on the other hand, shows stepwise improvement in its best solution over the iterations. It converges at the best optimal value found so far by all the algorithms.

Figure 4 depicts the comparison result of convergent of SPSO with constriction coefficient and have these main advantages

Figure 3. PSO with Inertia weight

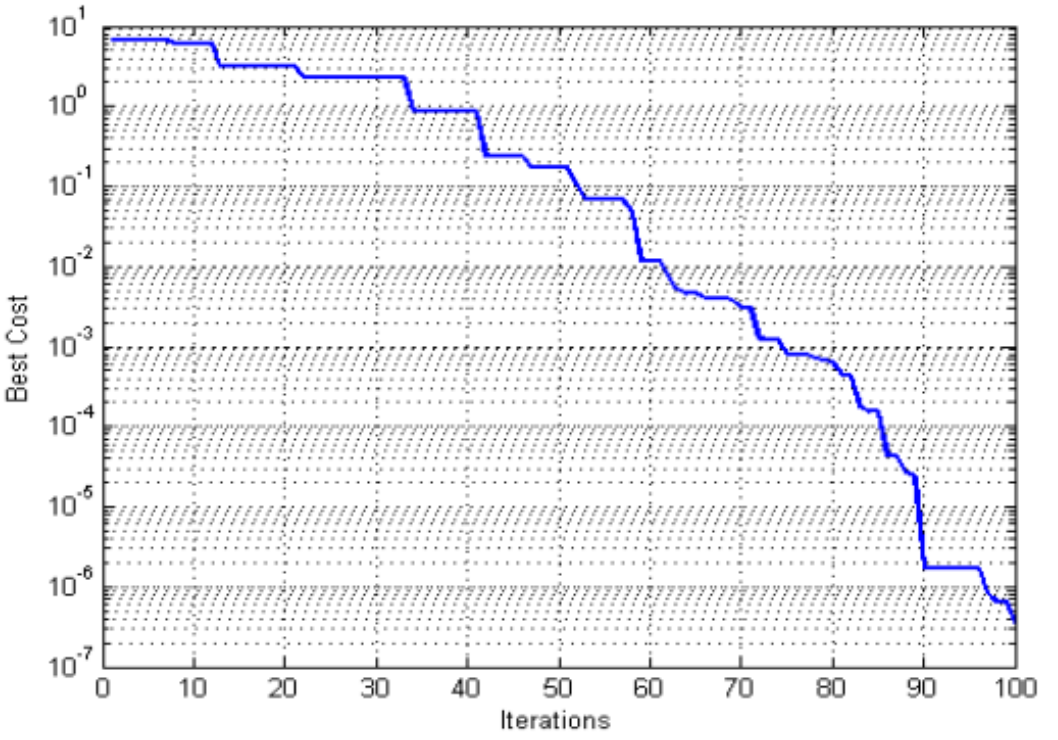
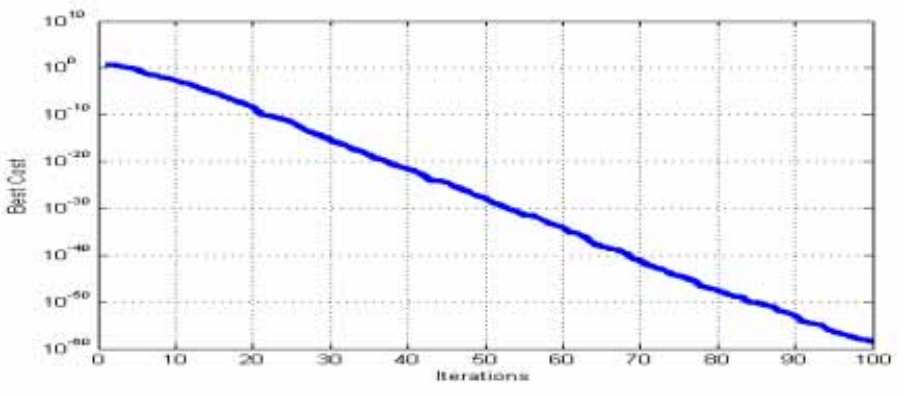


Figure 4. PSO with Constriction coefficient



- Higher efficiency and probability to find the global optima
- Fast convergence
- No overlapping or mutation
- Low computational time

4.1.3 Population Size

There are several important parameters in PSO algorithm, among that we concentrate on the influence of population size on the algorithm. Selection of population size is related to the problems to be solved, but it is not very sensitive to the problem. In most cases, most iterative search methods encourage exploration and exploitation during the early stages and the latter stage of the iterative process, respectively. It is also well-known that concentrating individuals only on the area of the highest potential during the exploitation phase is usually beneficial. Thus, as this area is small relative to the whole search space, a small swarm may perform almost as well as a large swarm. Hence, it might be better to reduce the population size at the latter stages of the optimization process to save on the number of function evaluations. Common selection is $20-50$. In some cases, larger population is used to meet the special needs (Wang et.al., 2018).

There were researches which determined Probably the first attempt to design an adaptive population size in an evolutionary algorithm in 1994. The idea of adapting population size was adopted for convergence PSO via a variant called "Ladder Particle Swarm Optimization", LDPSO. In LDPSO, it is true that the diversity of the swarm was evaluated and the swarm size was adjusted at every predefined period of time. The diversity measure was based on the average Manhattan distance between particles. Comparisons showed that the LDPSO algorithm outperforms some other PSO variants in terms of the average solution quality when it was applied to 5 standard benchmark problems.

In Bonyadi and Michalewicz (2017), it was proposed that the idea of population management, solution sharing technique and a search range sharing strategy for PSO variant was called " Efficient Population Utilization Strategy PSO", EPUS-PSO. The population manager in EPUS-PSO adjusted the population size by:

1. Removing a particle from the swarm when global best vector was improved at least once in the previous *k* consecutive iterations.
2. Adding a new particle to the swarm if the global best vector did not change in the previous *k* consecutive iterations.
3. Replacing an existing one when adding is not permitted (the upper bound of the number of particles in the swarm has been reached).

As above mentioned, in EPUS-PSO, a solution sharing strategy was used a tournament selection approach for global best vector in the velocity update rule of each particle with the probability Pr and the personal best of another particle with probability $q=1-Pr$. The aim of the solution sharing strategy was to give a chance to particles to learn from other particles rather than always using the global best vector in the velocity update rule. Furthermore, a solution sharing strategy was presented to prevent the particles from premature convergence. The algorithm was applied to 15 benchmark test functions, together with their rotated versions in 10 and 30 dimensions.

Results obtained based on the average and standard deviation showed that the proposed algorithm is capable of finding high quality solutions and it is comparable with other PSO variants on the tested problems.

Another essential strategy in PSO called "Incremental Social Learning", ISL it was used to set the population size of SPSO during the run. ISL suggests an increasing population-size approach that in some cases facilitates the scalability of systems composed of multiple learning agents. Researchers were proposed two PSO variants based on the ISL idea:

1. IPSO, whenever the algorithm could not find a satisfactory solution, a new particle was added to the population.
2. IPSOLS (IPSO with a local search), a local search approach was run to gather local information around the current position.

If the local search procedure was not successful, then it was concluded that the particle is already in a local optimum. In this stage, a new particle was added to the swarm that was placed in the search space through a simple social learning approach. The algorithm was applied to some standard optimization test functions and the mean and median of the results were compared with those of other PSO methods. Results showed that the proposed method is capable of finding high-quality solutions and it is comparable with other methods on the tested problems (Bonyadi and Michalewicz, 2017).

4.2 Modification of the Velocity-Position Update Rules

Mubeen and Dhananjay (2022), Shi and Eberhart (1998b), and Zhang et al. (2015) were the first individual to discuss the parameter inertia weight in basic PSO selection. They brought an inertia efficient ω into the PSO and promoted the convergence feature. Probably the first modification of the velocity in the basic algorithm introduced by the name called SPSO. Over the past decades, many attempts were made to improve the velocity and position update rule of SPSO further, by considering these:

1. Two Steps Forward One Step Back
2. Cooperative Particle Swarm Optimization

SPSO was identified an important issue called " two steps for ward, one step back " stated as all dimensions of the position of particles are updated at every iteration, there is a chance that some components in this vector move closer to a better solution, while others actually move away from that good solution. As long as the objective value of the new position vector is better than the objective value of the previous position, SPSO assumes that the solution has been improved, regardless if some dimension values have moved away from the good solution. To address this issue, Bonyadi and Michalewicz (2017) was proposed to divide the search space to *d* subproblems and different variables are updated by different subswarms, the variant is called " Cooperative Particle Swarm Optimization", CPSO . In addition, CPSO performed better than many other variants of PSO, it had two major issues as follows:

- It might converge to pseudominima,
- Its performance depends on the correlation of the subproblems.

As SPSO does not suffer from these issues, a hybrid method was also proposed in Bonyadi and Michalewicz (2017) that combined SPSO with CPSO.

The global best vector in SPSO is considered as an indicator of a potentially high–quality area in the search space. Thus, it would be worthwhile to use this location together with the current location of a particle to generate a new location with no velocity vector in it. This idea is called "extrapolation Particle Swarm Optimization", ePSO (Bonyadi and Michalewicz, 2017), in which the position update rule was revised as follows:

$$\vec{x}_{t+1}^{i} = \vec{g}_t + \alpha\vec{g}_t + \gamma\left(\vec{g}_t - \vec{x}\right) \tag{16}$$

Where $\alpha = k_1 r_t^i$, $\gamma = k_1 e^{k_2\beta}$, $k_1 = k_2 = e^{\left(\frac{-current\ iteration}{\max number\ of\ iterations}\right)}$, r_t^i is a random value between 0 and 1, and $\beta = \dfrac{f\left(\vec{g}_t\right) - f\left(\vec{x}_t^i\right)}{f\left(\vec{g}_t\right)}$.

The term $\vec{g}_t + k_1 \vec{x}_t^i \vec{g}_t$ generates a random point around the global best vector. Note that, using limit concept as $k_1 \to 0$ over time (iterations), the generated points by $\left(\vec{g}_t + k_1 \vec{x}_t^i \vec{g}_t\right) \to \vec{g}_t$ that results in better exploitation around the global best vector at the later stages of the run. The last term $k_1 e^{k_2 \beta} \left(\vec{g}_t - \vec{x}_t^i\right)$ determines the step size that the current position $\left(\vec{x}_t^i\right)$ takes to go towards the global best vector. This step size is controlled by the objective value of the current location and the objective value of the global best vector.

According to the velocity update rule of SPSO, attraction towards the personal or global best vectors does not depend only on φ_1 and φ_2 (the acceleration coefficients), but also depends on the average values of $\vec{p}_t^i - \vec{x}_t^i$ and $\vec{g}_t - \vec{x}_t^i$. It was observed that in SPSO the value of CI is usually smaller than SI for any particle i and any iteration t that results in larger attraction towards $\vec{g}_t$ than $\vec{p}_t^i$. Hence, particles tend to concentrate around $\vec{g}_t$ that reduces the diversity of the swarm. To overcome this issue, a variant of PSO called " Comprehensive Learning Particle Swarm Optimization" CLPSO, was proposed (Ashok et.al., 2016 and Bonyadi and Michalewicz, 2017). In CLPSO, the velocity update rule was modified to the movement equation:

$$\vec{V}_{t+1}^i = \acute{\mathrm{E}}\, \vec{v}_t^i + \varphi_1 R_{1t}^i \left(\vec{P}_t^i - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i \left(\vec{g}_t - \vec{x}_t^i\right) \tag{17}$$

where the value of the j^{th} dimension of $\vec{P}_t^i \left(\vec{P}_t^{ij}\right)$ is calculated by $\vec{p}_t^{ij} = p_t^{l_i(j)j}$, where $l_i\left(j\right) \in \left\{1,2,...,n\right\}$ (index of a particle). In fact, each dimension of $\vec{p}_t^i$ is potentially taken from the personal best of different particles (Bonyadi and Michalewicz, 2017).

It shows potential locations where $\vec{p}_t^i$ might be selected from for a particle i in a swarm of size 5 as an example value of $l_i\left(j\right)$ was set to i with the probability Pr while it was set through a tournament selection to select the value of $l_i\left(j\right)$ with probability $q = 1 - Pr$. Results on 16 benchmark test functions showed that the algorithm outperforms several other PSO variants on multimodal functions, while it performs almost the same as the others on unimodal functions. The discussion over the results was supported by the mean and variance of the found solutions.

The good performance of CLPSO on multimodal optimization problems stems from its effectiveness in exploration. In addition, as mentioned, CLPSO does not perform as well on unimodal optimization problems, which reflects the weak exploitation ability of the algorithm. To improve the ability of the CLPSO algorithm for exploitation. In Bonyadi and Michalewicz (2017) we were introduced new variant of PSO called " Example-based Learning Particle Swarm Optimizer ", ELPSO, Eq.17 was revised as follows:

$$\vec{V}_{t+1}^i = \acute{\mathrm{E}}\, \vec{v}_t^i + \varphi_1 R_{1t}^i \left(\vec{P}_t^i - \vec{x}_t^i\right) + \varphi_2 R_{2t}^i \left(\vec{G}_t - \vec{x}_t^i\right) \tag{18}$$

where $\vec{P}_t^i$ is defined similarly to the one in CLPSO,except the strategy of setting $l_i\left(j\right)$ which was selected randomly from a uniform distribution over all particles and each dimension j of $\vec{G}_t$ (shown by $\vec{G}_t^j$ is randomly taken from a set of previously sampled positions by $\vec{g}_t$). It was shown that the searching interval of each dimension by each particle is larger than what it was in CCPSO and CLPSO, which indicates a better diversification in the swarm. Experiments were conducted in (Nasir et.al., 2012) on 16 benchmark test functions and the t-test was used to compare the results. These experiments showed that ELPSO is more effective than CLPSO in multimodal and unimodal optimization problems.

The velocity update rule of SPSO was revised in Bonyadi and Michalewicz (2017) and a new method called " perturbed Particle Swarm Algorithm ", pPSA, was proposed to prevent SPSO from premature convergence. In pPSA, the vector $\vec{g}_t$ in the velocity update rule of the standard PSO is substituted for $N\left(\vec{g}_t, \sigma^2 I\right)$, where N is the normal distribution and σ^2 is the variance. The value of σ was set through a very simple time-varying strategy. The results of applying the algorithm to some standard either unimodal or multimodal benchmarks showed that the algorithm performs better than SPSO in terms of the quality of solutions and robustness. In comparison to GCPSO, the results showed that both methods perform almost the same on multimodal functions, while GCPSO performs better than pPSA in unimodal functions. Results were compared based on mean, median, and standard deviation, which appears sufficient to show the advantages. No theoretical analyses were included in that article, however, it seems the algorithm can escape stagnation and guarantees local convergence as it uses a mutated global best vector with non-zero variance.

In addition, the algorithm is the same as SPSO in terms of transformation invariance, i.e.; rotation variance with scale and translation invariance.

The behavior of the particles is especially related to the exploitative and explorative diversity of particles. An algorithm variant of PSO called " Diversity enhanced Neighborhood Search Particle Swarm Optimization", DNSPSO, in which the explorative behavior was controlled by enhancing the diversity of the particles in the swarm. At each iteration, the position and velocity of each particle is updated by the rules in SPSO. The new position $\vec{x}_{t+1}^{i}$ is then combined with $\vec{x}_t^{i}$ to generate a trial particle. This is done by taking either the value of $x_t^{i,j} or\ x_{t+1}^{ij}$ for each $j \in \left\{1,\ldots,d\right\}$ with probability Pr. The personal best of the particle i is updated according to the trial particle. Furthermore, two local searches were used to search the neighborhood of the personal best and global best vectors, which actually improved the exploitation ability of the algorithm. The usage of these neighborhood search strategies is beneficial for accelerating the convergence rate. The algorithm was applied to 30 benchmark functions, 10 of them with up to 50 dimensions and 20 of them with 1,000 dimensions, and the results were compared with other methods such as CLPSO, APSO, and CPSO using Friedman test. Comparisons showed that the algorithm is comparable with others for both groups of test functions (Bonyadi and Michalewicz, 2017).

4.3 Hybridization

Over the years, researches on performance improvement of PSO have focused on fusing with other optimization methods. The aim of a hybrid method is to combine different optimization methods to take advantage of the merits of each of them such as:

- To improve the local search ability of the PSO algorithm.
- To increase the diversity and avoid premature convergence of the PSO algorithm.

The improvement is based on the integration of other approaches, viz., the so-called hybrid soft computing (HSC). For instance, using one method to adjust the parameters, including the inertia weight and convergence factor used to set the coefficients of SPSO during the run using DE and chaotic map or running different methods iteratively to improve the outcome of one another (Bonyadi, et.al, 2013 and Kalita et.al., 2019) where a GA was combined with SPSO.

we presented a novel particles movement named as "oscillation" which introduced the bond of each particle attracted by personal best and global best vector and "two steps forward, one step back" were investigated in Zhan et al. (2011).

In order to combine global and personal best vectors and use the combined vector in the velocity update rule, instead to study these cases the "Orthogonal Experimental Design", OED, approach was applied to informed the new PSO variant was called "Orthogonal Learning Particle Swarm

Optimization",OLPSO. The combined position was then used in the velocity update rule exactly as in CLPSO (see Eq.17) where $\vec{p}_t^{\,i}$ was replaced by the vector generated by OED approach. This strategy was tested on SPSO and LPSO where results on some test cases showed significant improvement (based on the t-test) on the performance of these algorithms when OED is used. The optimal solution of most of the tested functions was at the center of the coordinate system (Bonyadi and Michalewicz, 2017)].

Author's presented different strategies to improve a better topology and adaptive coefficients of the PSO algorithm for updating each component were taken from previous studies and combined to generate a new PSO variant, called "Frankensteins Particle Swarm Optimization", FPSO.

The main purpose of the method was to combine different PSO variants together and create a new method that enables them to overcome each other's deficiencies. FPSO components for effective experimental results were:

- The inertia weight setting taken from DIPSO.
- The velocity was updated by FIPS formulation.
- Acceleration coefficients and setting of V_{max} were set by the method used in HPSO-TVAC.
- The topology was set by the method of *adaptive variant*.

Some experimental results showed that FPSO is effective and outperforms the methods it has been composed of. In addition, the same idea was tested in Tang et al. (2011) and a variant called " Feedback Learning Particle Swarm Optimization", FLSPO, was proposed. FLPSO used DIPSO for improve locally convergent

- Inertia weight
- An adaptive approach for φ_1 and φ_2
- An adaptive strategy to use personal best or global best vector in the velocity update rule
- A mutation operator applied to a randomly selected dimension of the global best vector.

As such a mutation operator prevents stagnation, it is very likely that FLPSO is locally convergent. Although PSO performance has improved over the past decades, how to select suitable velocity updating strategy and parameters remains an important research domain. In Andizzon et al. (2015) we proposed a novel example of the basic particle swarm concept, with two types of agents in the swarm, "explorers" and "settlers", that could dynamically exchange their role during the search procedure. This approach can dynamically update the particle velocities at each time step according to the current distance of each particle from the best position found so far by the swarm. With good exploration capabilities, uniform distribution random numbers in the velocity updating strategy may also affect the particle moving. Thus, (Fan andYan, 2014) put forward a "Self-Adaptive PSO with Multiple Velocity Strategies", SAPSO-MVS to enhance PSO performance. SAPSO-MVS could generate self-adaptive control parameters in the total evolution procedure and adopted a novel velocity updating scheme to improve the balance between the exploration and exploitation capabilities of the PSO algorithm, and it was proposed to avoid to tune the PSO parameters manually.

As there are many different position/velocity update rules and each of them contain their good point and bad point in different situations, it would be beneficial to investigate which update rule is more beneficial at the current situation (iteration) and use that strategy for further iterations. For example, a self-adaptive method was proposed in which the most effective PSO variant was selected during the run (iteration) for the problem with different position/velocity update rules (Wang et al, 2013). This new learning algorithm variant was called "Self-Adaptive Learning Particle Swarm Optimization", SALPSO. As each velocity update rule had its own capabilities in exploration or exploitation situations, it was expected that the overall activities of the algorithm are improved. To

explain this situation, for each particle and every k iterations (a constant experimentally set to 10), the probability of using each update rule was updated based on the improvement it produced during the last *k* iterations.

To update the particles, the velocity update rules were selected for each particle at each iteration according to their probabilities. Results showed that SLPSO is effective in dealing with both unimodal and multimodal optimization problems by observing results found based on mean and standard deviation, which proved the effectiveness of the objective function for convergence analysis. In (Ivo and Duarte, 2017) presented a very similar concept by the method was called "Self-Learning Particle Swarm Optimization", SLPSO. The only difference between idea of Self Learning and Self Adaptive Learning is in the implementation of PSO algorithm, such as how to update probabilities and which velocity update rules are used in details. In (Mubeen and Dr. Dhananjay, 2022) proposed Crazy PSO in which particle velocity was randomized within predefined limits. Its aim was to randomize the velocity of some particles, named as "crazy particles" through using a predefined probability of craziness to keep the diversity for global search and better convergence.

The ideas used in SALPSO and SLPSO triggered the emergence of another PSO variant called " Multiple Adaptive Methods for Particle Swarm Optimization", MAM-PSO. In MAM-PSO, all particles were updated by the update rules in SPSO except the global best particle. A mutation operator was proposed in Zou et al. (2015) that generated a random point (according to some distribution) around current and moved the global best particle (the particle which its personal best is $\vec{g}_t$) to that random point rather than a moving the particle according to its velocity. To update the global best particle, two operator approaches, a mutation and a gradient descent operator approach, were designed and one of them was selected randomly at each iteration.

An extensive experiment was conducted that showed MAM-PSO is comparable with other PSO variants such as CPSO and CLPSO, which proved the effectiveness of the objective function for convergence analysis. We conjecture that the algorithm is locally convergent as it uses a mutation operator based on a Cauchy distribution with the center of the current position and a non-zero scale parameter. Furthermore, Meng et al. (2015) and Mubeen and Dhananjay (2022) introduced crisscross search particle swarm optimization (CSPSO), a new hybrid optimization technique and expanded the convergence of PSO algorithm through introducing the possibility of c-means and probability theory, and put forward probabilistic PSO algorithm analysis. There are some other high-quality articles related to the hybridization of PSO, where SPSO was combined with CMA-ES and self-adaptive particle swarm optimization conducted in Harrison et al. (2018) and the following most recent studies focused on in an improved convergence particle swarm optimization algorithm with random sampling of control parameters (SC-PSO), the main contribution and comparison in different aspects such as the convergence performance of SC-PSO, the convergence curves of the PSO variants algorithms are on the some selected functions and the findings represent SC-PSO's convergence speed is faster than of any of the comparison algorithms and its advantages are prominent were discussed in Lijun et al. (2019) and Tong et al. (2019). All population-based metaheuristic optimization algorithms have two main parts—exploration and exploitation. Some optimization techniques are very good at exploration, and some are very good at exploitation. So, an optimizer with good exploration and bad exploitation is often combined with an optimizer with bad exploration and good exploitation to form a hybrid optimizer. Thus, the hybrid optimizers are very good at both exploration and exploitation. Some hybrid metaheuristic techniques are genetic algorithm (GA-PSO), Whale Optimization Algorithm with simulated annealing (WOA-SA), cuckoo search and differential evolution (CS-DE), Particle Swarm Optimization and Cuckoo Search (PSO-CS), Particle Swarm Optimization and Gravitational Search Algorithm (PSO-GSA) etc, well stated in Shankar et al. (2022).

There are several unsolved problems in PSO algorithm research, including but not limited to look for PSO applications. Because most PSO applications are currently limited to continuous, single-objective, unconstrained, deterministic optimization issues, we should focus on (i). potential future directions to investigate the performance of PSO variants to deal with UOPs through theoretical

perspective and general discussions on experimental results on merits of the proposed approach, e.g.; first hitting time, convergence, stability and fixed point and constraint handling method for constraint optimization problems (COPs). (ii). discrete/stochastic, multi-objective, constrained, un-deterministic, dynamic optimization problems. PSO's application areas should be increased at the same time. This path of research started in Bonyadi and Michalewicz (2017) but to the best of our knowledge, remained untouched for optimization algorithms in the field of evolutionary computation.

5. CONCLUSION

From the convergence analyses of PSO related to its convergence properties, transformation invariance, modification of components, coefficients adaptation, population sizing, topology, hybridization reviewed above, we can draw the following important conclusions.

- In Section 3, the first part of this article concentrates on the limitations of PSO that have been theoretically investigated in the past years. These limitations were categorized into two main groups: convergence and transformation invariance.
- In subsection 3.1, the convergence properties of PSO it was pointed out that analyzing different behaviors of particles before convergence and convergence analysis includs convergence to a fixed point, local convergence, and stagnation of other PSO variants and there are not many studies on the first hitting time analysis of PSO variants, constitute potential research directions.
- In subsection 3.2, an important area of study in convergence analyses of PSO is the transformation invariance property of the algorithm. Indeed, there are not many PSO variants that are transformation invariant, which makes this topic an open area for new ideas and analyses presented in some articles (Bonyadi and Michalewicz, 2017), these properties are important for an algorithm since without these properties it is impossible to scale its good performance achieved on a test set to a wide variety of problems.
- In subsection 4.1, apart from theoretical studies, the algorithm parameters in PSO are usually determined depending on the specific problems, application experience, and numerous experiment tests related to modifications of components and parameters of PSO. Topology of the swarm, setting coefficients, and population size are parameters of PSO that were investigated in this paper, so it has no versatility. Hence, how to determine the algorithm parameters conveniently and effectively is another urgent problem to be studied.
- In subsection 4.2, more attention should be emphasized on the highly efficient PSO algorithm and put forward suitable core update formula and effective strategy to balance the global exploration and local exploitation. In addition, articles that modified the velocity update rule of PSO were analyzed.
- In subsection 4.3, some other studies that hybridized the algorithm with other optimization methods to improve the overall performance of the designed optimizer were reviewed. We also reported the computational complexity of some of the methods included in the experimental part of this survey.
- The majority of the research articles in this paper deal with continuous variables. Limited study evidences that the PSO algorithm had some issues dealing with discrete variables.

In the light of the analyses given above, it is clear that the convergence analyses of PSO studied by many researchers are all under certain rigid assumptions. Although these assumptions can simplify the problem models and then implies the concise advantages/ benefits of models can be summarized as follows:

- It has excellent robustness and can be used in different application areas with a little modification.

- It has strong distributed ability, because the algorithm is essentially the swarm evolutionary algorithm, so it is easy to realize parallel computation.
- It can converge to the optimization value quickly.
- It is easy to combine with other algorithms to improve its performance.
- It can be easily studied and analyzed by researchers to identify the path of future work problems and how they can be translated into real-world benefits.

REFERENCES

Agrafiotis, D. K., & Cedeno, W. (2002). Feature selection for structure-activity correlation using binary particle swarms. *Journal of Medicinal Chemistry*, *45*(5), 1098–1107. doi:10.1021/jm0104668 PMID:11855990

Andizzon, G., Cavazzini, G., & Pavesi, G. (2015). Adaptive acceleration coefficient for a new search diversification strategy in particle swarm optimization algorithms. *Inf Sci*, *299*, 337–378. doi:10.1016/j.ins.2014.12.024

Baba, N. (1981). Convergence of a random optimization method for constrained optimization problems. *Journal of Optimization Theory and Applications*, *33*(4), 451–461. doi:10.1007/BF00935752

Banks, A., Vincent, J., & Anyakoha, C. (2008). A review of particle swarm optimization. part ii: Hybridization, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing*, *7*(1), 109–124. doi:10.1007/s11047-007-9050-z

Bonyadi, M., & Michalewicz, Z. (2015a). (to appear). Analysis of stability, local convergence, and transformation sensitivity of a variant of particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*.

Bonyadi, M., & Michalewicz, Z. (2015b). (to appear). Stability analysis of the particle swarm optimization without stagnation assumption. *IEEE Transactions on Evolutionary Computation*.

Bonyadi, M. R. (2014). SPSO2011 – Analysis of Stability, Local Convergence, and Rotation Sensitivity. doi:10.1145/2576768.2598263

Bonyadi, M. R., Li, X., & Michalewicz, Z. (2013). A hybrid particle swarm with velocity mutation for constraint optimization problems. In *Genetic and Evolutionary Computation Conference*. ACM. doi:10.1145/2463372.2463378

Bonyadi, M. R., Li, X., & Michalewicz, Z. (2014a). A hybrid particle swarm with a time-adaptive topology for constrained optimization. *Swarm and Evolutionary Computation*, *18*, 22–37. doi:10.1016/j.swevo.2014.06.001

Bonyadi, M. R., & Michalewicz, Z. (2014a). A locally convergent rotationally invariant particle swarm optimization algorithm. *Swarm Intelligence*, *8*(3), 159–198. doi:10.1007/s11721-014-0095-1

Bonyadi, M. R., & Michalewicz, Z. (2014d). SPSO2011- analysis of stability, local convergence, and rotation sensitivity. In *Genetic and Evolutionary Computation Conference (GECCO).* ACM.

Bonyadi, M. R., & Michalewicz, Z. (2017). *Particle swarm optimization for single objective continuous space problems: a review. Journal of Evolutionary Computation*. doi:10.1162/EVCO_r_00180

Bonyadi, M. R., & Michalewicz, Z. (2017). Impacts of coefficients on movement patterns in the particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, *21*(3), 378–390.

Bonyadi, M. R., & Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: A review. *Evolutionary Computation*, *25*(1), 1–54. doi:10.1162/EVCO_r_00180 PMID:26953883

Bonyadi, M. R., Michalewicz, Z., & Li, X. (2014b). An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal of Heuristics*, *20*(4), 417–452. doi:10.1007/s10732-014-9245-2

Campana, E., Fasano, G., & Pinto, A. (2010). Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization. *Journal of Global Optimization*, *48*(3), 347–397. doi:10.1007/s10898-009-9493-0

Cleghorn, C. W., & Engelbrecht, A. (2015). *Fully Informed Particle Swarm Optimizer: Convergence Analysis*. IEEE Press.

Cleghorn, C. W., & Engelbrecht, A. P. (2014a). A generalized theoretical deterministic particle swarm model. *Swarm Intelligence*, *8*(1), 35–59. doi:10.1007/s11721-013-0090-y

Cleghorn, C. W., & Engelbrecht, A. P. (2014b). Particle swarm convergence: An empirical investigation. In *IEEE Congress on Evolutionary Computation.* IEEE Press.

Cleghorn, C. W., & Engelbrecht, A. P. (2014c). Particle swarm convergence: Standardized analysis and topological influence. In *LNCS* (Vol. 8667, pp. 134–145). Springer.

Cleghorn, C. W., & Engelbrecht, A. P. (2015). Particle swarm variants: Standardized convergence analysis. *Swarm Intelligence*, *9*(2), 177–203. doi:10.1007/s11721-015-0109-7

Cleghorn, C. W., & Engelbrecht, A. P. (2018). Particle swarm stability: A theoretical extension using the non-stagnate distribution assumption. *Swarm Intelligence*, *12*(1), 1–22. doi:10.1007/s11721-017-0141-x

Dong, W. Y., & Zhang, R. R. (2019). Order-3 stability analysis of particle swarm optimization. *Information Sciences*, *503*, 508–520. doi:10.1016/j.ins.2019.07.020

Dong. (2013). A Review of Convergence Analysis of Particle Swarm Optimization. *International Journal of Grid and Distributed Computing, 8*(8), 117-128.

Engelbrecht, A. (2012). Particle swarm optimization: Velocity initialization. In *IEEE Congress on Evolutionary Computation.* IEEE Press.

Fan, Q., & Yan, X. (2014). Self-adaptive Particle swarm optimization algorithm with multiple velocity strategies andits application for p-xylene oxidation reaction process optimization. *Chemometrics and Intelligent Laboratory Systems*, *139*, 15–25. doi:10.1016/j.chemolab.2014.09.002

Ganesh, N., Ragavendran, U., Kalita, K., Jain, P., & Xiao-Zhi, G. (2021). Multi-Objective High-Fidelity Optimization Using NSGA-III and MO-RPSOLC. *Computer Modeling in Engineering & Sciences*, *129*(2), 443–464. Advance online publication. doi:10.32604/cmes.2021.014960

Garcıa-Gonzalo, E., & Fernandez-Martınez, J. L. (2014). Convergence and stochastic stability analysis of particle swarm optimization variants with generic parameter distributions. *Applied Mathematics and Computation*, *249*, 286–302. doi:10.1016/j.amc.2014.10.066

Geng, J., Li, M., Dong, Z., & Liao, Y. (2014). Port throughput forecasting by MARS-RSVR with chaotic simulated annealing particle swarm optimization algorithm. *Neurocomputing, 147*, 239–250.

Gong, Y. J., Li, Y. J., Zhou, Y., Li, Y., Chung, H. S.-H., Shi, Y.-H., & Zhang, J. (2017). Genetic learning particle swarm optimization. *IEEE Transactions on Cybernetics*, *46*(10), 2277–2290. doi:10.1109/TCYB.2015.2475174 PMID:26394440

Handayani, D., Nuraini, N., Tse, O., Saragih, R., & Naiborhu, J. (2016). *Convergence analysis of particle swarm optimization (PSO) method on the with-in host dengue infection treatment model*. AIP Publishing LLC. doi:10.1063/1.4945071

Harrison, K. R., Ombuki-Berman, B. M., & Engelbrecht, A. P. (2017). *Optimal parameter regions for particle swarm optimization algorithms. In 2017 IEEE congress on evolutionary computation*. IEEE. doi:10.1109/CEC.2017.7969333

Harrison, K. R., Ombuki-Berman, B. M., & Engelbrecht, A. P. (2018). Self-adaptive particle swarm optimization: A review and analysis of convergence. *Swarm Intelligence*, *2018*(12), 187–226. doi:10.1007/s11721-017-0150-9

Helwig, S., Branke, J., & Mostaghim, S. (2013). Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *17*(2), 259–271. doi:10.1109/TEVC.2012.2189404

Jagerskupper, J. (2008). Lower bounds for randomized direct search with isotropic sampling. *Operations Research Letters*, *36*(3), 327–332. doi:10.1016/j.orl.2007.10.003

Jiang, M., Luo, Y., & Yang, S. (2007a). Particle swarm optimization-stochastic trajectory analysis and parameter selection. In Swarm Intelligence, Focus on Ant and Particle Swarm Optimization. I-TECH Education and Publishing. doi:10.5772/5104

Kalita, K., & Chakraborty, S. (2023). An efficient approach for metaheuristic-based optimization of composite laminates using genetic programming. *Int J Interact Des Manuf*, *17*(2), 899–916. doi:10.1007/s12008-022-01175-7

Kalita, K., Dey, P., & Haldar, S. (2019). Robust genetically-optimized skew laminates. *Proceedings of the Institution of Mechanical Engineers. Part C, Journal of Mechanical Engineering Science*, *233*(1), 146–159. doi:10.1177/0954406218756943

Kalita, K., Ghadai, R. K., & Chakraborty, S. (2021). A comparative studyon the metaheuristic-based optimization of skew composite laminates. *Engineering With Computers, 38*, 3549–3566. .10.1007/s00366-021-01401-y

Kim, S., & Li, L. (2014). *Statistical identifiability and convergence evaluation for nonlinear pharmacokinetic models with particle swarm optimization*. Academic Press.

Kumar, Singh, & Patro. (2016). Particle Swarm Optimization: A Study of Variants and Their Applications. *International Journal of Computer Applications, 135*(5).

Lehre, P. K., & Witt, C. (2013). *Finite first hitting time versus stochastic convergence in particle swarm optimization. In Advances in Metaheuristics*. Springer.

Liu, J., Ma, X., Li, X., Liu, M., Shi, T., & Li, P. (2020). Random Convergence Analysis of Particle Swarm Optimization Algorithm with Time-varying Attractor. *Swarm and Evolutionary Computation*. Advance online publication. doi:10.1016/j.swevo.2020.100819

Liu, Q. (2015). Order-2 stability analysis of particle swarm optimization. *Evolutionary Computation*, *23*(2), 187–216. doi:10.1162/EVCO_a_00129 PMID:24738856

Liu, Q., Wei, W., Yuan, H., Zhan, Z. H., & Li, Y. (2016). Topology selection for particle swarm optimization. *Information Sciences*, *2016*(363), 154–173. doi:10.1016/j.ins.2016.04.050

Lu, J., Hu, H., & Bai, Y. (2015a). Generalized radial basis function neural network based on an improved dynamic particle swarm optimization and a daboost algorithm. *Neurocomputing*, *152*, 305–315. doi:10.1016/j.neucom.2014.10.065

Meng, A., Li, Z., Yin, H., Chen, S., & Guo, Z. (2015). Accelerating particle swarm optimization using crisscross search. *InfSci, 329*, 52–72.

Mubeen & Yadav. (2022). A review of Particle Swarm Optimization (PSO) algorithm. *International Journal of Mechanical Engineering and Technology*, *13*(7), 19–44.

Nasir, M., Das, S., Maity, D., Sengupta, S., Halder, U., & Suganthan, P. N. (2012). A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Information Sciences*, *209*, 16–36. doi:10.1016/j.ins.2012.04.028

Netjinda, N., Achalakul, T., & Sirinaovakul, B. (2015). Particle Swarm Optimization inspired by starling flock behavior, Appl.soft. *The Computer Journal*, *35*, 411–422.

Ozcan, E., & Mohan, C. K. (1999). Particle swarm optimization: surf ing the waves. In *IEEE Congress on Evolutionary Computation.* IEEE Press.

Poli, R., Kennedy, J., & Blackwell, T. (2007b). Particle swarm optimization: An overview. *Swarm Intelligence, 1*(1), 33–57.

PSO. (2006). *PSO source code.* http://particleswarm.infostandard PSO-2006.c

Schmitt, M., & Wanka, R. (2013). Particle swarm optimization almost surely finds local optima. In *Genetic and Evolutionary Computation Conference.* ACM. doi:10.1145/2463372.2463563

Shankar, R., Ganesh, N., Cep, R., Narayanan, R. C., Pal, S., & Kalita, K. (2022). Hybridized Particle Swarm—Gravitational Search Algorithm for Process Optimization. *Processes (Basel, Switzerland)*, *2022*(10), 616. doi:10.3390/pr10030616

Shi, Y., & Eberhart, R. (1998b). Parameter selection in particle swarm optimization. In Porto, V., Saravanan, N., Waagen, D., and Eiben, A. In *LNCS* (Vol. 1447, pp. 591–600). Springer.

Sousa-Ferreira, I., & Sousa, D. (2017). A review of velocity-type PSO variants. *Journal of Algorithms & Computational Technology*, *11*(1), 23–30. doi:10.1177/1748301816665021

Sun, L., Song, X., & Chen, T. (2019). An Improved Convergence Particle Swarm Optimization Algorithm with Random Sampling of Control Parameters. Journal of Control Science and Engineering.

Tang, Y., Wang, Z., & Fang, J. (2011). Feedback learning particle swarm optimization. *Applied Soft Computing*, *11*(8), 4713–4725. doi:10.1016/j.asoc.2011.07.012

Tanweer, M. R., Suresh, S., & Sundararajan, N. (2016). Dynamic mentoring and self-regulation based PSO algorithm for solving complex real world optimization problem. *Inf Sci*, *326*, 1–24. doi:10.1016/j.ins.2015.07.035

Tian, D., & Shi, Z. (2018). Modified particle swarm optimization and its applications. *Swarm and Evolutionary Computation*, *41*, 49–68. doi:10.1016/j.swevo.2018.01.011

Tong, J. Q., Zhao, Q., & Li, M. (2019). Particle swarm optimization algorithm based on adaptive dynamic change. *Microletronics & Computer*, *36*(2), 6–10.

Trelea, I. C. (2003). The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, *85*(6), 317–325. doi:10.1016/S0020-0190(02)00447-7

Van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences*, *176*(8), 937–971. doi:10.1016/j.ins.2005.02.003

Wang, D., Tan, D., & Liu, L. (2018). *Particle swarm optimization algorithm: an overview, 22(2)*. Springer.

Wang, H., Sun, H., Li, C., Rahnamayan, S., & Pan, J. (2013). Diversity enhanced particle swarm optimization with neighborhood search. *Information Sciences*, *223*, 119–135. doi:10.1016/j.ins.2012.10.012

Witt, C. (2009). Why standard particle swarm optimizers elude a theoretical runtime analysis. In *Foundations of Genetic Algorithms.* ACM.

Xue, Y., Tang, T., & Liu, A. X. (2019). Large-scale Feedforward Neural Network Optimization by a self-Adaptive Strategy and Parameter Based PSO. *IEEE Access : Practical Innovations, Open Solutions*, *7*, 52473–52483. doi:10.1109/ACCESS.2019.2911530

Yang, C., Gao, W., Liu, N., & Song, C. (2015). Low-discrepancy sequence initialized PSO alogorithm with high-order nonlinear time –varying inertia weight. *Applied Soft Computing*, *29*, 386–394. doi:10.1016/j.asoc.2015.01.004

Ye, H., Luo, W., & Li, Z. (2013). Convergence Analysis of Particle Swarm Optimizer and Its Improved Algorithm Based on Velocity Differential Evolution. Computational Intelligence and Neuroscience.

Zhan, Z.-H., Zhang, J., Li, Y., & Shi, Y. (2011). Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *15*(6), 832–847. doi:10.1109/TEVC.2010.2052054

Zhang, L., Tang, Y., Hua, C., & Guan, X. (2015). A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques. *ApplSoftComput, 28*, 138–149.

Zou, R., Kalivarapu, V., Winer, E., Oliver, J., & Bhattacharya, S. (2015). Particle swarm optimization-based source seeking. *IEEE Transactions on Automation Science and Engineering*, *12*(3), 865–875. doi:10.1109/TASE.2015.2441746