CONVERGENCE AND MATHEMATICAL ANALYSIS OF PARTICLE SWARM OPTIMIZATION

Dereje Tarekegn Nigatu

A Dissertation Submitted to the Department of Applied Mathematics,
School of Applied Natural Science

Presented in Fulfillment of the Requirement for the Degree of Doctor of Philosophy in Applied Mathematics (Numerical Analysis)

Office of Graduate Studies
Adama Science and Technology University

December, 2024
Adama, Ethiopia

CONVERGENCE AND MATHEMATICAL ANALYSIS OF PARTICLE SWARM OPTIMIZATION

Dereje Tarekegn Nigatu

Supervisor: Surafel Luleseged Tilahun (Ph.D)
Co-supervisors: Tekle Gemechu Dinka (Ph.D)

A Dissertation Submitted to the Department of Applied Mathematics,
School of Applied Natural Science

Presented in Fulfillment of the Requirement for the Degree of Doctor of Philosophy in Applied Mathematics (Numerical Analysis)

Office of Graduate Studies
Adama Science and Technology University

December, 2024
Adama, Ethiopia

Declaration

I hereby declare that this dissertation entitled " **Convergence and Mathematical Analysis of Particle Swarm Optimization** " is my original work. It has not been submitted for the award of any academic degree, diploma or certificate in any other university. All sources of materials used for this dissertation have been duly acknowledged through appropriate citations.

Name of student Signature Date

Recommendation

We, the supervisors of this dissertation, hereby certify that we have closely supervised the student while doing this dissertation by reading and revising the dissertation entitled "**Convergence and Mathematical Analysis of Particle Swarm Optimization** " prepared under our guidance by **Dereje Tarekegn Nigatu** submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Mathematics (Numerical Analysis). Therefore, we recommend the submission of the dissertation to the department for further review and defense.

Supervisor	Signature	Date
Co-supervisor	Signature	Date

Approval Page

We hereby certify that the recommendations and suggestions made by the board of examiners are appropriately incorporated into the final version of the dissertation entitled " **Convergence and Mathematical Analysis of Particle Swarm Optimization** " by **Dereje Tarekegn Nigatu**.

Supervisor	Signature	Date
Co-supervisor	Signature	Date

We, the undersigned, members of the Board of Examiners of the dissertation open defense by **Dereje Tarekegn Nigatu** have read and evaluated the dissertation entitled " **Convergence and Mathematical Analysis of Particle Swarm Optimization** " and examined the candidate during open defense. This is, therefore, to certify that the dissertation is accepted for partial fulfillment of the requirement of the degree of Doctor of Philosophy in Mathematics (Numerical Analysis).

Chairperson	Signature	Date
Internal Examiner 1	Signature	Date
Internal Examiner 2	Signature	Date
External Examiner 1	Signature	Date
External Examiner 2	Signature	Date

Finally, approval and acceptance of the dissertation is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the candidate's Department Graduate Council (DGC) and School Graduate Committee (SGC).

Department Head	Signature	Date
School Dean	Signature	Date
Office of Postgraduate Studies, Dean	Signature	Date

ACKNOWLEDGMENT

After an intensive period of learning, today marks a significant milestone: writing this note of thanks is the final touch on my dissertation. This journey has profoundly impacted me, not only in the scientific realm but also on a personal level. I wish to take a moment to reflect on the individuals who have supported me throughout this period.

First and foremost, I would like to express my heartfelt gratitude to God. His guidance has empowered me to believe in myself and pursue my dreams. I could not have achieved this without the faith I hold in the Almighty.

Next, I want to extend my sincere thanks to my Supervisor and Co-supervisor, Dr. Surafel Luleseged and Dr. Tekle Gemechu. I deeply appreciate their invaluable guidance, unwavering dedication, and the high academic standards they upheld during the preparation of this dissertation.

I also owe a great deal to my parents for their love, wise counsel, and constant support throughout my life. They have always been there for me, teaching me invaluable lessons and providing the best advice. Their encouragement and prayers have given me the strength to overcome the challenges I faced during this process.

The journey of completing this dissertation has had its ups and downs, and the encouragement from colleagues and friends has been instrumental in helping me reach this point. Therefore, I would like to thank all my fellow PhD students for their support.

Finally, I am grateful to Adama Science and Technology University and Debre Birhan University for providing the necessary funding for my dissertation. I would also like to express my appreciation to the staff of the Department of Mathematics at both universities for their assistance.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

GAs	Genetic algorithms
GP	Gaussian process
P_{mut}	Mutation probability
P_{cx}	Crossover probability
SI	Swarm Intelligence
PSO	Particle Swarm Optimization
ACO	Ant colonies optimization
ASTU	Adama Science and Technology University
DE	Differential Evolution
APSO	Adaptive PSO
EA	Evolutionary Algorithms
SA	Simulated Annealing
MATLAB	Matrix Laboratory
ABC	Artificial Bee Colony
ODE	Ordinary Differential Equation
ANNs	Artificial Neural Networks
HSC	Hybrid Soft Computing
EFHT	Expected First Hitting Time
AC	Ant Colony
CPSO	Constriction Particle Swarm Optimization
QPSO	Quantum-behaved Particle Swarm Optimization
SPSO	Standard Particle Swarm Optimization
CSPSO	Constriction Standard Particle Swarm Optimization
QBPSO	Quantum-Behaved Particle Swarm Optimization
QBCSPSO	Quantum-Behaved Constriction Standard Particle Swarm Optimization
GCPSO	Guaranteed Convergence Particle Swarm Optimizer
PSOVZRFC	Particle Swarm Optimization Velocity to Zero Repulsive Function Constraint
ATP	Theoretical Attractor Point
aBBPSO	alternative Bare Bones Particle Swarm Optimization
PSO-VZP	Particle Swarm Optimization with current Velocity to Zero Problem
GVZRFCP	Generalized Velocity to zero Repulsive Functional Constraint Problem
VZSPSO	Velocity zero Standard Particle Swarm Optimization

ABSTRACT

The particle swarm optimization(PSO) is one of a very useful heuristic algorithms. In PSO, we investigate the simulation of natural swarm behaviors, such as those of birds or insects, which search for significant areas by exchanging information and cooperating rather than competing. A number of artificial particles move through the domain $\mathbb{R}^{\mathbb{D}}$, with each particle's movement influenced by its own experiences as well as those of the swarm agents. In particular, for guaranteeing convergence, necessary and sufficient conditions to a certain swarm parameters that control the behavior of the swarm could be derived. In order to measure, how fast the swarm at a certain time converges, we define and analyze the potential of parameters on a particle swarm. When the swarm is far away from a local optimum, the potential increases in the 1-dimensional case. This reflection turns out to be sufficient to prove the main results, namely that in a 1-dimensional situation, the swarm with probability 1 converges towards a local optimum for a comparatively wide range of objective functions. Additionally, we applied a Markov chain to demonstrate that, for benchmark function tests, the numerical simulation of the PSO algorithm convergence analysis shows that the constriction standard particle swarm optimization (CSPSO) algorithm enhances iteration generation performance. In the general dimension the swarm might not converge towards a local optimum. Instead, it gets adhered to a position where some dimensions have a potential orders of size smaller than others. In such situations the algorithm converges towards a point in the search space, is not even a local optimum which is called swarm stagnates. In order to solve this issue, we propose a variant of PSO, such as standard particle swarm optimization repulsive functional constraint (SPSO-RFC) velocity to zero that guarantees convergence towards a local optimum. As a result, the CSPSO and PSO-RFC algorithms demonstrate better optimal solutions compared to the PSO variants, owing to the effects of parameter settings within the PSO algorithm.

Keywords: PSO, convergence and stability, parameters, Markov chain, PSO variants.

CHAPTER 1

INTRODUCTION

1.1 Background of the Study

In this chapter, we give a brief discussion on the background of swarms and their behavior in nature which is related to convergence of the PSO algorithm or its variants.

1.1.1 Metaheuristic Optimization Alogorithm

Particle swarm optimization (PSO), first studied by Kennedy & Eberhart (1995); Cui & Gao (2012), is a very popular nature-inspired metaheuristic for solving continuous optimization problems. It is designed to reflect the social learning (interaction) of individuals living together in groups and supporting and cooperating with each other by communicating and exchanging information about their best solutions, rather than competing against each other.

Some of the general fields of successful application were mentioned in the work of Dorigo & Blum (2005); Karaboga & Basturk (2007); Antoniou et al. (2009, 2013), where the continuous objective function on a multi-dimensional domain is not given in a closed form, but by a "black box". That means that the only operation that the objective function allows is the evaluation of search points while gradient information is not available. This characteristic allows PSO to be applied to a broader range of problems, including those where the objective function is non-differentiable or complex, thus enhancing its versatility and robustness in various optimization scenarios.

Swarm is a group of individuals with defined rules for individual behaviors and communications. The ability of each individual to deal with the previous experiences of the swarm is called swarm intelligence. This capability guides the swarm toward its optimum goal. PSO is a population-based search technique in which a population of particles start their journey in a space with respect to the current best position introduced (Hossain & El-Shafie, 2014).

The popularity of the PSO framework in various scientific communities is due to the fact that it on the one hand can be realized and, if necessary, adapted to further needs easily, but on the other hand shows in experiments good performance results with respect to the quality of the obtained solution and the time needed to obtain it. By adapting its parameters, users may in real-world applications easily and successfully control the swarm's behavior with respect to "searching where no one has searched before" and "searching around a good position" are called "exploration" and "exploitation ", respectively. Thorough discussion of PSO can be found in (Li & Qian, 2003).

Let an objective function $f : \mathbb{R}^D \to \mathbb{R}$ on a D-dimensional domain be given that (without loss of generality) has to be minimized. A population of particles, each consisting of a position, a velocity and a local attractor, moves through the search space $\mathbb{R}^D$. The population in motion is the swarm whereas the local attractor of a particle is the best position with respect to objective function (f) this particle has encountered early.

In contrast to evolutionary algorithms, the individuals of the swarm cooperate by sharing information about the search space via the global attractor, which is the best position any particle finds early. The discrete representation of time allows for easier numerical computation and simulation of particle motion. The particles move in discrete time iterations. The movement of a particle is governed by equations that depend on the velocity of the particle's and its personal and global attractors and on some additional fixed parameters, controlling the influence of the attractors and velocity on the next iteration (step).

The main goal of this dissertation is to provide the first general mathematical analysis of the quality of the global attractor when it is considered as a solution for objective functions from a very general class F of functions and therefore of the quality of the algorithm's resulting values. The set F consists of all functions that have a continuous first derivative and, in general, have only a bounded area of interest for the particles. Note that the class F of admissible objective functions, for which our convergence results hold, is much more general than the subset of the class of the unimodal functions that is considered by Neshat et al. (2013) in the context of restricted $(1 + 1)$ evolutionary algorithms.

First, we propose a mathematically sound PSO model, which describes the algorithm as a stochastic process over the real numbers. Before presenting our convergence analysis, several previous results, in particular different negative results stating that the PSO algorithm is not an "optimizer", are discussed in light of this new model. It is clear that there will always be a need for better optimization algorithms. The particle swarm optimizer was introduced by Kumar (2021), yet very few formal analyses of the behaviour of the algorithm have been published. Most of the published work was concerned with empirical results obtained by changing some aspect of the original algorithm.

If the algorithm has been shown to be able to solve some difficult optimization problems, what could be said about the infinite number of problems that have not yet been studied empirically? While the results obtained from empirical (trial-and-error) comparisons provide useful insights into the nature of the PSO algorithm, it is clear that a general, theoretical description of the behaviour of the algorithm is important.

A meta-heuristic is formally defined as an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies used to structure information in order to find efficiently near-optimal solutions.

Meta-heuristic is considered to be an algorithmic structure that generally applied to a variety of optimization problems with only a few modifications for adapting to the given problem. Due to complexity and numerical property of the problem, a black box solver was developed which gives better solution.

In recent years, several new extension of PSO algorithms were subsequently developed, with the aid of the theoretical description of the PSO method. These algorithms were constructed to improve the performance of the PSO technique.

1.2 Swarm Intelligence in Nature

Swarm is any living organism, as a collection of identical individuals evolving in a nonsynchronized way. Examples of swarms in nature include fishes schooling, bird flocking, animal herding, hawk hunting, and bacterial growth. They in nature are extremely varied in terms of size: they can vary from a few individuals living in a small area to large colonies spread on a broad territory and consisting of millions of individuals. They are also decentralized, meaning that there is no leader in the swarm who guides the other members to perform the intended tasks.

For centuries, the concept of intelligence has been invariably linked to human beings, neglecting the notion that other natural creatures could develop sophisticated strategies other than merely instinctive behaviors. However, a simple observation of nature provides multiple cases that contradict this assertion.

Swarm intelligence (SI) is the collective behavior of decentralized systems composed of many individuals that coordinate their activities using self-organization (Cui & Gao, 2012; Cui et al., 2019). A typical example is given by the collective behavior of colonies of several social insects (ants, termites, bees, wasps, fireflies, and so on). Swarm intelligence systems are inspired from biological systems which are usually consisted of a population of simple agents interacting locally with one another and globally with their environment.

The ant colonies, bird flocking, bee colonies, and fish schooling are some example of natural SI systems (Dorigo & Blum, 2005; Antoniou et al., 2009, 2013; Karaboga & Basturk, 2007; Li & Qian, 2003; Neshat et al., 2013). Based on these systems, several optimization algorithms such as ant colony (AC), artificial bee colony (ABC), and PSO have been developed. A standard textbook on PSO, treating both the social and computational paradigms, is given (Eberhart, 2001).

Optimization is a process that finds a best, or optimal, solution for the problem. An optimization problem is defined as: Finding values of the variables that minimize or maximize the objective function while satisfying the constraints. The term optimization refers to both minimization and maximization tasks. A task involving the maximization of the function f is equivalent to the task of minimizing $-f$, therefore the terms minimization, maximization and optimization can be used interchangeably (Rao, 2019; Beni, 2020).

This dissertation considers minimization, formally defined by:
Given $f : R^n \longrightarrow R$, find

$$x^* \in R^n \;\; for \;\; which \;\; f(x^*) \leq f(x) \, or \, f(x^*) \geq f(x), \;\; \forall x \in R^n \tag{1.1}$$

The Optimization problems are based on these factors:

(i). An objective function which is to be minimized or maximized.

(ii). A set of unknowns or variables that affect the objective function values.

(iii). A set of constraints that allow the unknowns to take on certain values but exclude others.

Approximation methods which can find a good enough solution in a reasonable time, these methods can be categorized as the most known algorithms form of heuristics and meta-heuristics. The significant difference between both is that heuristics are more problem-dependent than meta-heuristics.

Meta-heuristic algorithms are among these approximate techniques as PSO algorithm. The prominent features of PSO are its easy implementation, robustness to control parameters and computation and efficiency in continuous problems compared with other existing heuristic algorithms such as the genetic algorithm (Bansal et al., 2011).

1.2.1 Particle Swarm Optimization

Particle swarm optimization is a population-based stochastic optimization originated from artificial life and evolutionary computation. PSO is motivated by the social behavior of organisms is an evolutionary algorithm based on swarm intelligence which simulates birds or fish predation. The PSO algorithm is initialized with a population of random candidate solutions, conceptualized as particles. Each particle is assigned a randomized velocity and location iteratively moved through the problem search space.

It is attracted towards the location of the best fitness achieved so far by the particle itself and by the location of the best fitness achieved so far across the whole population. The PSO algorithm needs some tuning parameters that greatly influence the algorithm performance often referred to as the exploration-exploitation trade-off.

Exploration is the ability to test various regions in the problem space in order to locate a good optimum, hopefully the global one. Exploitation is the ability to concentrate the search around a promising candidate solution in order to locate the optimum accurately. In PSO, the velocity equation is used to calculate the velocity of particles based on their best position and the position of the best particle in the swarm (1.2) particle positions can be updated according to their current positions $(x_i's)$ and velocities $(v_i's)$, is given in (1.3). This iterative process continues until reaching the termination criteria (Dorigo & Blum, 2005).

The velocity and position of each particle are updated as

$$v_{n(i)} = v_{n(i-1)} + c_1 r_{1n(i)}(bp_{n(i)} - x_{n(i)}) + c_2 r_{2n(i)}(bg_{n(i)} - x_{n(i)}) \tag{1.2}$$

$$x_{n(i+1)} = x_{n(i)} + v_{n(i)} \tag{1.3}$$

where, $v_{n(i-1)}$ is the velocity of n^{th} particle in past iteration and $v_{n(i)}$ is the velocity of n^{th} particle in current iteration. $bp_{n(i)}$ is the best position of n^{th} particle so far, $bg_{n(i)}$ is the position of the best particle of the swarm so far, and $x_{n(i-1)}$ and $x_{n(i)}$ are the position of n^{th} particle respectively in the current and the next iterations. The constants c_1 and c_2 are called the cognitive and social parameters, and are responsible for defining a particle's affinity towards its own best and the global best solutions, respectively. The variables $r_{1n(i)}$ and $r_{2n(i)}$ are drawn from a uniform random distribution [0, 1] and are the source of randomness in the search behavior of the swarm.

Three components of velocity equation are respectively the initial, cognitive(own), and social parts. Cognitive and social weights, control how much the global and individual best positions should influence the particle's velocity and trajectory.

A complete theoretical analysis of the selection of the algorithm parameters remains empirical to a large extent (Kennedy & Eberhart, 1995; Wang, 2018). Based on this analysis, the authors derived a reasonable set of tuning parameters, as confirmed by (Eberhart & Shi, 2000). The newly established parameter selection guidelines are applied to standard benchmark functions.

The references by Kennedy & Eberhart (1995) and Wang (2018) contain considerable mathematical complexity, making it challenging to derive user-oriented guidelines for parameter selection in a specific problem. However, results from the dynamic system theory are used for a relatively simple theoretical analysis of the algorithm which results in graphical guidelines for parameter selection. Metaheuristic in general for all PSO in particular has some advantages when compared to other continuous optimization techniques; for instance:

(i). It does not make assumptions on the continuity and differentiability of the objective function to be optimized.

(ii). It does not need to compute the gradient of the error function and

(iii). It does not need good initial starting points or deep prior knowledge about the most promising areas of the search space.

1.2.2 Application of PSO within Swarm Intelligence (SI)

The swarm intelligence algorithm is a search algorithm based on the principles of evolution and natural genetics. Having been established as a valid approach to problems that require efficient and effective search, PSO algorithms are increasingly finding widespread application in business, scientific, and engineering circles. But this application of the PSO algorithm to have a solution may depend on the initial population, the number of iterations, the velocity components, the acceleration coefficients, and so on. The results also change depending on the number of generation of the evolution or degree of convergence. Ever since the PSO algorithm has been introduced by Eberhart & Kennedy (1995), and Kennedy & Eberhart (1995), it has been successfully applied in various fields, like engineering. This popularity is due to the comprehensible performance of PSO as well as the simplicity of its operation.

Many scholars applied PSO to solve their problems in the fields of structural engineering, environmental engineering, hydrological engineering, and geotechnical engineering (Gholizadeh & Salajegheh, 2009; Mashhadban et al., 2016; Wan, 2013; Najafzadeh & Tafarojnoruz, 2016; Kuok et al., 2010; Armaghani et al., 2014; Sadoghi Yazdi et al., 2012). In addition to the aforementioned applications, PSO is also used as a learning algorithm in artificial neural networks (ANNs) that are widely used in geotechnical engineering (Armaghani et al., 2014; Hasanipanah et al., 2016).

On the other hand, PSO as a robust global search algorithm is a suitable alternative to modify the weight and bias of ANNs for getting higher performance prediction. Hence, a combination PSO-ANN algorithm receives advantages of both PSO and ANN to solve or optimize the problems in which PSO searches for global minimum and ANN uses the global minimum to find the best result. Despite the superiority of PSO in solving complex problems, it is occasionally faced with some limitations such as premature and divergence phenomena (Neshat, 2013).

Consequently, some modifications have been suggested to avoid these limitations and enhance the PSO capabilities by introducing inertia weight, constriction factor, crossover operation, self-adaptation, and logistic dynamic weight (Lu et al., 2006; Zhang et al., 2013; Løvbjerg et al., 2001; Eberhart & Shi, 2000; Shi & Eberhart, 1999; Ni & Deng, 2013).

To the best of our knowledge the impact of parameter setting, modifying the component of the algorithm, and fusing the algorithms for improving convergence of the PSO algorithm together by taking into account performance evaluation and comparison on variant of PSO algorithm is not studied. This is addressed in this dissertation by considering the logical influence of cognitive and social influences with respect to the length of the time step. In this study, the PSO algorithm that incorporates various parameter factors will be investigated both quantitatively and numerically. We proposed PSO variants with convergence analysis.

1.3 Statement of the Problem

Swarm has for many years been considered a global issue and major causes of SI. Many researchers invest their effort in investigating the dynamics of swarm and to control its convergence. From interactions with those scientists, mathematicians and engineers have developed a significant and effective tool, namely PSO algorithms giving an insight into the interaction between the swarm and particle, the dynamics of swarm behaviour, how to control swarm parameters, and eventually how to stop (converge) it. Algorithm of PSO has flourished since the days of (Kennedy & Eberhart, 1995). They developed an original PSO algorithm of velocity and position. There are no mathematical studies undertaken for performance improvement of PSO by modifying its parameters, increase population diversity, hybrid with other optimization methods and the applications of PSO in different areas. Therefore, this study investigated the convergence and stability analysis of the mathematical algorithm for the impact of parameters on PSO.

1.4 Objectives of the Study

1.4.1 General Objective of the Study

The general objective of this study is investigating convergence and mathematical analysis of the PSO algorithm.

1.4.2 Specific Objectives of the Study

The specific objectives of this study are to

• analyze the convergence of PSO.

• construct PSO variants based on different parameters.

• conduct a parametric setup analysis of the reformulated algorithms.

• assess the effect of each parameter on the convergence of the particles.

• analyze stability of PSO methods.

1.5 Significance of the Study

The convergence analysis of standard particle swarm optimization with different approaches has contributed various achievement in science field in recent years. Thus, the study is significant for the following:

- A theoretical analysis of the behaviour of the PSO algorithm under different parametric settings. This analysis provides PSO algorithm that can be used to predict the long-term behaviour of swarm particles.
- It would help to determine the effect of PSO algorithm convergence in SI management, considering inertia weight, learning factors and random number.
- It would provide alternative means of testing convergence of PSO algorithm.
- It will be a useful reference for any future study for monitoring and controlling strategy for the convergence of PSO algorithms with selected parameters.

1.6 Scope of the Study

This study focused on the development and analysis of numerical methods for convergence and stability of Metaheuristic algorithm. Because of the difficulty of finding criteria for stopping iteration or convergence point of algorithm, we only focused on particles swarm optimization of the form standard PSO algorithm. The numerical experiments are carried out using MATLAB or Mathematica software both in the computer laboratories in ASTU, and DBU.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews the PSO algorithm with termination criteria developed by different researches to study different techniques with different improvement parameters that are directly related to the objectives of this study.

2.1 Particle Swarm Algorithm

PSO convergence has attracted many researchers'attention since it was introduced in 1995. There has been a considerable amount of work done in developing the original version of PSO algorithm, through empirical simulations (Kennedy & Eberhart, 1995; Eberhart, 2001). In population-based optimization methods, proper control of global exploration and local exploitation is crucial in finding the optimum solution efficiently.

Shi & Eberhart (1998a) introduced the concept of inertia weight to the original version of PSO, in order to balance the local and global search during the optimization process. Generally, in population-based search optimization methods, considerably high diversity is necessary during the early part of the search to allow use of the full range of the search space. On the other hand, during the latter part of the search, when the algorithm is converging to the optimal solution, fine-tuning of the solutions is important to find the global optima efficiently.

Considering these concerns, Shi & Eberhart (1999), and Jain et al. (2018) have found a significant improvement in the performance of the PSO method with a linearly varying inertia weight ω over the generations. The mathematical representation of this concept is given by

$$v_i d = \omega \times v_i d + c_1 \times rand(\cdot) \times (p_i d - x_i d) + c_2 \times Rand(\cdot) \times (p_g d - x_i d) \tag{2.1}$$

where ω is given by $\omega = (\omega_1 - \omega_2) \times \frac{(MAXITER - iter)}{MAXITER} + \omega_2$
with ω_1 and ω_2 are the initial and final values of the inertia weight, respectively, iter is the current iteration number and MAXITER is the maximum number of allowable iterations.

A number of studies have been conducted to highlight the criteria to control iteration and its dynamic by developing different type of convergence technique, where as Kennedy & Eberhart

(1995) developed PSO as a stimulation of birds swarm.

Reynolds (1987) described three simple rules governing the behavior of individuals within a swarm, which serve as foundational concepts for Particle Swarm Optimization as noted by (Kennedy & Eberhart, 1995; Beni, 2020). Although these simple rules model the behavior of individuals, their combination produces a complicated behavior for the swarm:

(i). Individuals avoid collision with others.

(ii). Individuals go toward the goal of swarm.

(iii). Individuals go to the center of swarm.

The process of decision making related to individuals is other basic concept of PSO. Each individual of the swarm makes decision based on the following two factors:

(i). The own experiences of individual that is its best results so far.

(ii). The experiences of other individuals in the swarm that is the best results in the whole swarm.

At the starting step of the original PSO a certain number of individuals, called particles, are distributed in the search space by using a random pattern and each particle is a representative of a feasible solution (Cheng et al., 2007; Jain et al., 2018).

The schematic structure of a particle in PSO involving three divided parts as its current position, best position, and velocity. The current position, best position, and velocity of particles record respectively the current coordinates, best coordinates, and velocity vectors of a particle in D-dimensional space, where D starts from one (Kalatehjari, 2013). Consequently, for a particle in D-dimensional space, a 3D-dimensional particle is desirable.

To select an appropriate termination criterion, it should be noted that the termination condition does not cause a prematurely converge and it should protect against oversampling of the fitness (Engelbrecht, 2007). The following termination criteria are frequently used in PSO:

(i). Termination when the maximum number of iterations is exceeded.

(ii). Termination when unsatisfactory solution is found based on the condition of each problem.

(iii). Termination when no improvement is achieved over a certain number of iterations.

These criteria are applied to ensure that PSO is able to converge on a feasible solution. In fact, PSO tries to make the objective function as minimum or maximum depend to the problem to be solved. To lead the swarm toward this aim, the fitness value of each particle is determined by evaluating its current position by the objective function. After evaluation the fitness of all particles. The convergence of PSO algorithms should be based on the framework of random search algorithm of Feng et al. (2013), and it has been proved by Van den Bergh & Engelbrecht (2006) that PSO is not a global optimization algorithm and also cannot be guaranteed to converge to a local optimum.

Trelea (2003) utilized the linear dynamic system theory with constant coefficient to analyze the stability of basic PSO, and Clerc & Kennedy (2002) established a constraint model of PSO described by only five parameters; then the convergence and trajectories of particles in phase plane were analyzed. The PSO convergence analysis investigates whether a global optimal solution can be reached when particle swarm converges.

Jiang et al. (2007b) presented a stochastic convergence analysis on the standard PSO (SPSO) by using stochastic process theory. Combining with finite element grid technique, Poli & Langdon (2007) set up a discrete Markov chain model of the bare-bones particle swarm optimization.

Cai et al. (2009) developed an absorbing Markov process model of PSO. They suggested the attaining-state set as the key factor of convergence analysis and proposed an improved convergence method in terms of the theorem of expanding attaining-state set. Based on the convergence criterion of pure random search algorithm (Solis & Wets, 1981; Jain et al., 2018).

Van den Bergh & Engelbrecht (2010) proved that the original PSO is neither a global search algorithm, nor a local one. Sudholt & Witt (2010) gave a lower bound for the time needed to optimize any pseudo-Boolean functions with a unique optimum. To prove upper bounds they transferred a fitness-level argument that is well-established for evolutionary algorithms to PSO.

Sun et al. (2012) put forward the quantum-behaved particle swarm optimization (QPSO) algorithm and discussed the convergence of QPSO. It was proved that QPSO is a global convergence algorithm. The study by Lehre & Witt (2013) showed that the convergence properties of the basic PSO in stagnation may be detrimental to find a sufficiently good solution within reasonable time, and that the basic PSO may have infinite expected first hitting time on some functions.

Recently, Ren et al. (2011) analyzed the global convergence of PSO by introducing the transition probability of particle. Several properties related to Markov chain were investigated and it was found that the particle state space is non-recurrent and PSO is not globally convergent from the viewpoint of the transition probability.

According to the difference model of PSO, Feng et al. (2013) examined the Markov properties of the state sequences of a single particle and particle swarm and calculated the transition probability of a particle. The transition probability to the optimal set was then deduced by combing the law of total probability with the Markov properties, which proves that SPSO can reach the global optimum in probability.

Ping Tian (2013) surveyed the existing work on the convergence analyses of PSO. Yuan & Yin (2014) proved convergence of the swarm for a stochastic PSO with evolving estimate. Although many methods have been proposed for the PSO convergence analysis, most analyses are based on ergodicity of the Markov process which strongly depends on the transition matrix and their eigenvalues (Poli & Langdon, 2007). Therefore, current PSO convergence analyses are very complicated and time-consuming, especially when the population size is large.

2.2 Mathematical Analysis of Particle Swarm Optimization

The martingale theory avoids solving complex transition matrix eigenvalues and the evolutionary sequence of particle swarm with the best fitness value corresponds to a super martingale. Thus the martingale theory is used to effectively simplify the SPSO convergence analysis. Another important variant of standard PSO is the constriction factor approach of PSO (CPSO), which was proposed in (Eberhart & Shi, 2000; Clerc & Kennedy, 2002). The velocity of CPSO is updated by the following equation (2.2) as below:

$$v_{ij}(t+1) = \chi.\{v_{ij}(t) + c_1.rand1.(pbest_{ij}(t) - x_{ij}(t)) + c_2.rand2.(gbest_j(t) - x_{ij}(t))\} \quad (2.2)$$

where, χ is called a constriction factor, given by equation (2.3):

$$\chi = 2/(|2 - \varphi + \sqrt{(\varphi^2 - 4\varphi)})| \quad (2.3)$$

where, $\varphi = c_1 + c_2\ , \quad \varphi > 4.$

The system in (2.2) and (2.3) is a discrete-time system. The variable t denotes generation or iteration and not continuous time. The CPSO ensures the convergence of the search procedures and can generate higher-quality solutions than the standard PSO with inertia weight on some studied problems (Neshat, 2013).

Inspired by the works of Eberhart & Shi (2000); Clerc & Kennedy (2002); Ping Tian (2013); Yuan & Yin (2014), we proposed to use a new PSO variants and analysis convergence by parameter control. This study is planned in alignment with the following research methodology.

CHAPTER 3

METHODOLOGY

In this chapter, we outline the methodology employed which encompasses key components: the study area, data collection, and the mathematical procedures utilized for analysis.

3.1 Introduction

As PSO is in nature a population-based stochastic optimization algorithm, the motion of particle swarm can be described by a Markov chain based on the stochastic process. According to the mathematical description of Standard PSO (SPSO), from the work of Van den Bergh & Engelbrecht (2006), substituting (3.1) in to (3.2)

$$v_i^{k+1} = \omega v_i^k + c_1 r_1^k (p_i^k - x_i^k) + c_2 r_2^k (g^k - x_i^k) \tag{3.1}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad , \tag{3.2}$$

the following non-homogeneous recurrence relation is obtained:

$$x_i^{k+1} = x_i^k + \omega(x_i^k - x_i^{k-1}) + \phi_1^k (p_i^k - x_i^k) + \phi_2^k (g^k - x_i^k) \tag{3.3}$$

where

$$\phi_1^k = c_1 r_1^k, \;\; \phi_2^k = c_2 r_2^k$$

the definitions of the Markov properties and the transition probability of swarm state can be given as follows.

Definition. Let $\{Y^k,\ k \geq 0\}$ be a discrete stochastic process, where Y^k takes values in a general state space S.

For $\forall k \geq 0$ and $i^l \in S(l \leq k+1)$, if $P\{Y^0 = i^0,\ Y^1 = i^1,\ \ldots\ ,\ Y^k = l^k\} > 0$ and

$$P\{Y^{k+1} = i^{k+1} \mid Y^0 = i^0,\ Y^1 = i^1,\ \ldots,\ Y^k = i^k\} = P\{Y^{k+1} = i^{k+1} | Y^k = i^k\},$$

$\{Y^k,\ k \geq 0\}$ is then defined as a Markov chain (Lawler, 2018).

The particle state and the swarm state of

$$x_i^{k+1} = x_i^k + \omega(x_i^k - x_i^{k-1}) + \phi_1^k (p_i^k - x_i^k) + \phi_2^k (g^k - x_i^k)$$

are defined, and the proof of Markov property of the swarm state and the transition probability of the swarm state is calculated.

Mathematical analysis techniques for PSO algorithms are employed to establish conditions or threshold parameters that define particle behavior. We utilize the Markov chain method to derive convergence. Additionally, we analyze the constriction coefficient of model parameters to assess their impact on the convergence of SPSO results or predictions. Convergence analysis using differential equations is conducted by examining the relationship between inertia weight and particle velocity. Furthermore, the convergence of PSO is studied through difference equations, matrices, and limit concepts. Estimation of certain parameters of the model or parameter fitting is done by using the data obtained from the benchmark function using statistical tools and numerical software. Study of dynamical behaviour in terms of threshold parameters and simulations are also done using suitable benchmark function.

3.2 Data Analysis

To obtain the necessary data for studying swarm behavior in relation to convergence analysis, data from published papers are used to provide parameter values. Using these data, we investigated the algorithm we developed to describe the dynamics behavior of the convergence of the SPSO and its variant algorithms.

3.3 Methodology

In this dissertation, we presented a numerical model which describes the impact of parameter variability on PSO algorithms with convergence speed control and best cost effectiveness analysis using the PSO algorithm for parameter variation with respect to the PSO hybrid algorithm and comparison with the optimal solution obtained by the applying procedures.

The objective of this dissertation is achieved by performing the theoretical and numerical methods. Firstly, the domain in which the algorithm is mathematically well defined and numerically meaningful was determined.

The qualitative and quantitative analysis of the model is presented. Both local and global stability analysis of the equilibrium points of the model have been performed using probabilistic matrix and repulsive functional, respectively.

In addition, the nature of the algorithm is identified using the method by Dorigo & Blum (2005); Dorigo & Stützle (2019) that is based on the center manifold theory. Constriction analyses of basic parameter numbers have been studied to determine which parameter plays a greater role in the convergence of the algorithm. The parameters of the model were estimated with real data using constriction formula to validate our algorithm.

Moreover, the PSO algorithm is extended to the optimal/velocity control problem. The existence of the optimal controls parameter is illustrated. Then the necessary conditions of the optimal control parameters have been derived using the constriction principle. Numerical method for solving benchmark function can then be employed to solve the resulting stagnation problem. The method called Social-Based Particle Swarm Optimization (SBPSO) is used to solve global (optimality) convergence system.

Lastly, numerical simulation of the algorithm is done using the MATLAB software to supplement the analytical solution of the algorithm. A cost-effective analysis is described with the purpose of finding the most cost-effective (fitness value) strategy for the prevention of stagnation (premature); the most effective and less costly strategy is obtained using the method called SPSO parameter control, which is defined as the ratio between a value in the interval (0, 1], and the value depends on acceleration coefficients. In this method, a combination of more than one controls at a time is implemented to identify the most optimal and least- cost strategy which minimizes the iteration number.

3.4 Ethical Issues

This proposed study contains only general research work that does not involve any kind of risk for its development. Furthermore, the researcher is committed to follow all ethical issues during this studies as per the rules and regulation of the university.

3.5 Outcomes

The finding of the research are :

- The mathematical algorithm with parameter uniform convergent.
- The best termination criteria strategy and cost-effective analysis for the PSO algorithm convergence used for researchers on this filed.
- It may serve as reference for those need to conduct research on this area.
- Articles were published to be used by the research community.

CHAPTER 4

CONVERGENCE ANALYSIS OF PARTICLE SWARM OPTIMIZATION

4.1 Introduction

Particle swarm optimization (PSO) algorithm is a stochastic optimization technique based on the behaviour of swarm (Agrafiotis & Cedeño, 2002; Beni, 2020). Main design idea of the PSO algorithm model is closely related to evolutionary algorithm and artificial life. In this chapter, we proposed algorithm that describes the convergence of PSO.

4.2 Algorithm

Just like evolutionary algorithm; PSO is also a population based algorithm to simultaneously search large region in the solution space of the optimized objective function. Artificial life studies the artificial systems with life characteristics (Van Den Bergh, 2001). It has been successfully applied to many problems, such as artificial neural network training, function optimization, fuzzy control, and pattern classification of Engelbercht (2005); Engelbrecht (2006); Poli (2008), to name a few. Because of its ease of implementation and fast convergence to acceptable solutions, PSO has received broad attention (Poli, 2008).

Since 1995, different aspects of the original or basic version of PSO have been modified and many variants have been 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 response it). However, we briefly discuss some of these methods from theoretical perspectives: convergence to local optima, transformation invariance, and the time complexity of the methods (Derrac et al., 2011).

4.3 Standard PSO

We consider a minimization problem defined as follows: find

$$\vec{x}^* \in S \subseteq \mathbb{R}^d \;\; such\ that\ \; \forall \vec{x} \in S,\; f(\vec{x}^*) \leq f(\vec{x}) \tag{4.1}$$

where S is the search space defined by $\{\vec{x} : -\infty < l^i \leq x^i \leq u^i < \infty,\ i = 1, \ldots, d\}$, 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 $f(.)$ is the objective function. In Kennedy & Eberhart (1995); Jain et al. (2018) 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 presented by Eberhart & Kennedy (1995); Jain et al. (2018) contains main three vectors:

- Position($\vec{x}_t^i$)–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. $\vec{x}_t^i \in \mathbb{R}^d$) (Van Den Bergh, 2001).
- Velocity ($\vec{V}_t^i$)–is the direction and length of movement of the i^{th} particle in the t^{th} iteration,
- Personal best ($\vec{p}_t^i$)– 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 (Eberhart & Kennedy, 1995; Beni, 2020).

All of these d-dimensional vectors ($\vec{x}_t^i$, $\vec{V}_t^i$, $\vec{p}_t^i$) are updated at every run t for each particle i:

$$\vec{V}_{t+1}^i = \mu(\vec{x}_t^i,\ \vec{V}_t^i,\ N_t^i)\ \ for\ all\ i \tag{4.2}$$

$$\vec{x}_{t+1}^i = \xi(\vec{x}_t^i,\ \vec{V}_{t+1}^i)\ \ for\ all\ i \tag{4.3}$$

$$\vec{P}_{t+1}^i = \begin{cases} \vec{x}_{t+1}^i, & for\ f(\vec{x}_{t+1}^i) < f(\vec{P}_t^i)\ and\ \ \vec{x}_{t+1}^i \in S \\ & \qquad\qquad\qquad\qquad for\ all\ \text{i} \\ \vec{P}_t^i, & otherwise \end{cases} \tag{4.4}$$

In $\vec{V}_{t+1}^i = \mu(\vec{x}_t^i,\ \vec{V}_t^i,\ N_t^i)$ 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. $N_t^i = \{P_t^k \mid k \in \{T_t^i \subset \{1, 2, 3, \ldots, n\}\}\}$ 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 (Van den Bergh & Engelbrecht, 2010). Example, global best topology, ring topology, wheel topology, pyramid typology; each of these have some advantages and disadvantages (Mendes et al., 2004). Topology in fact determines the set of particles from which a particle should learn (connect to). The function $\mu(.)$ calculates the new velocity vector for particle i according to its current position, current velocity V_t^i, and neighbor set N_t^i. In $\vec{x}_{t+1}^i = \xi(\vec{x}_t^i,\ \vec{V}_{t+1}^i)$ for all i, where $\xi(.)$ is a function which calculates the new position of particle i according to its previous position and its new velocity. Usually $\xi(x_t^i, V_{t+1}^i) = x_t^i + V_{t+1}^i$ is used for updating the position of particle i. In (4.1), the new personal best position ($\vec{P}_{i+1}^i$) for particle i is updated according to the objective value of its previous personal best position and the current position. In PSO, the three update rules $(4.2 - 4.4)$ are applied to all particles iteratively until a predefined stopping criterion is met. Also, $\vec{x}_o^i$ 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, Kennedy & Eberhart (1995), 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 ($\vec{p}_t^i$) 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., $T_t^i = \{i,\ \tau_t\}$ where $\tau_t = \underset{l=\{1,\ 2,\ \ldots,\ n\}}{arg\ min}\ (F(\vec{P}_t^l))$. This topology (or τ_t) is called global best topology for PSO. Also, the function $\mu(.)$ in (4.3) was defined by Kennedy & Eberhart (1995); Bonyadi & Michalewicz (2017) as:

$$\vec{V}_{t+1}^i = \vec{V}_t^i + \varphi_1 R_{1t}^i(\vec{P}_t^i - \vec{x}_t^i) + \varphi_2 R_{2t}^i(\vec{g}_t - \vec{x}_t^i) \tag{4.5}$$

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 $CI = (\vec{p}_t^i - \vec{x}_t^i)$ (Cognitive Influence) and $SI = (\vec{g}_t - \vec{x}_t^i)$ (Social Influence) is to attract the particles to move towards known quality solutions, R_{1t}^i and R_{2t}^i are two random d x d diagonal matrices, where their elements are random numbers distributed uniformly in $[0, 1]$, that is, $U(0, 1)$ (De Oca et al., 2009). Note that matrices R_{1t}^i and R_{2t}^i are generated at each iteration t for each particle i separately. There is no velocity initialization method that is superior to other methods in the general case; however, Engelbrecht (2012) has recommended that the velocity be initialized to zero.

In 1998, the authors Shi & Eberhart (1998a) and in 2013, Wang (2013) introduced a new coefficient $\omega \in (0, 1)$ called inertia weight, to control the influence of the previous velocity value on the updated velocity equation.

$$\vec{V}_{t+1}^i = \omega\vec{V}_t^i + \varphi_1 R_{1t}^i(\vec{P}_t^i - \vec{x}_t^i) + \varphi_2 R_{2t}^i(\vec{g}_t - \vec{x}_t^i) \tag{4.6}$$

Thus, the coefficients ω, φ_1, and φ_2 control influences of the previous velocity, CI, and SI on the particle movement, respectively. This variant of the PSO is called "Standard Particle Swarm Optimization", SPSO. In SPSO, the role of the inertia weight ω is considered to be crucial for the PSO's convergence (H. Liu et al., 2020). 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. This setup (considering similar random numbers on the diagonal of the random matrices) was introduced as a common error in the implementation of convergence analysis of PSO.

A number of articles on PSO have been published to date to name some (Clerc & Kennedy, 2002; Van den Bergh & Engelbrecht, 2002; Trelea, 2003; Y.-l. Zheng et al., 2003; Van den Bergh & Engelbrecht, 2004). In 2002, seven years after first version PSO was introduced, the first review on convergence analysis of PSO paper was published (Ping Tian, 2013). The performance of several convergence analysis 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 of Sahu et al. (2012) to jump out of local optima.

In 2013 review article by Ping Tian (2013) on convergence analysis of PSO was published that consisted of two main parts: the standard PSO algorithm and the PSO convergence analyses with six subsections including convergence analysis with constriction coefficient, limit, differential equation, matrix, difference equation, and Z 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 multi-objective (Mo) PSO variants were reviewed.

4.3.1 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). 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, we can see, 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 chapter mainly concentrates on the random convergence analysis of the PSO algorithm with time-varying attractor $Q(t) = \frac{\varphi_1 g(t)+\varphi_2 p(t)}{\varphi_1+\varphi_2}$ and the convergence analysis on the PSO algorithm with the time-varying attractor $Q(t+1) \neq Q(t)$ and the parameter $\eta = \frac{Q(t+1)-Q(t)}{v(t)}$ in some cases is not equal to 0.
Main contributions on the random convergence analysis of the random PSO algorithm can be highlighted as follows.

- 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 (J. Liu et al., 2021).

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 $Q(t+1) - Q(t)$, and the value of $Q(t+1) - Q(t)$ is mainly viewed as $\eta v(t)$, and the parameter η is introduced in the transfer matrix $M(t)$ (J. Liu et al., 2021). .

4.4 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 Wilke et al. (2007); Bonyadi, Li, & Michalewicz (2014) limitations in SPSO, the two main areas related to the limitations of convergence analyses of PSO are: convergence and transformation invariance. Under this section, we are going to mention some limitations related to these two topics in PSO convergence analyses.

4.4.1 Limitations Related to Convergence

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

One of the earliest and recent convergence analyzes of stochastic optimization algorithms was published in (Matyas et al., 1965; Baba, 1981; Solis & Wets, 1981; Doerr, 2020). An iterative stochastic optimization algorithm is said to converge to a point $\vec{X}$ in a search space in probability if

$$\forall \varepsilon > 0, \ \lim_{t \to \infty} Pr(| \ \vec{x}_t - \vec{X} \ |) \geq 1 - \varepsilon \tag{4.7}$$

where Pr is the probability measure, $\vec{x}_t$ 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 (4.7) 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. Based on the work of Birbil et al. (2004); Dasgupta et al. (2009); Doerr (2020), these two kinds of convergence have been investigated in detail in different optimization algorithms analyses.

The convergence analysis 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 an analysis sometimes leads to a set of parameter values for the algorithm that guarantee stability. However, during the convergence process to a point, the sequence of the solutions found by the algorithm might follow different patterns:

- high or low frequency movements
- larger or smaller jumps
- faster or slower convergence.

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

As presented by Dorea (1983), 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, where the convergence rate of evolutionary algorithms, as a related topic to EFHT, has been studied (He & Yao, 2002; Ratnaweera et al., 2004; Shakya & Santana, 2012; Mirjalili, 2018).

4.4.2 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 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 exploration by Clerc & Kennedy (2002); D. Kumar et al. (2021), 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 standard PSO 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 $[-V_{max},\ V_{max}]$ (Y.-L. Zheng et al., 2003; Helwig & Wanka, 2007, 2008; Helwig et al., 2012). Then, the movement equations become

$$\vec{V}^i_{t+1} = \max\{-V_{max}, min\{V_{max}, \vec{V}^i_t + \varphi_1 R^i_{1t}(\vec{P}^i_t - \vec{x}^i_t) + \varphi_2 R^i_{2t}(\vec{g}_t - \vec{x}^i_t)\}\}$$
$$\vec{X}^i_{t+1} = \vec{X}^i_t + \vec{V}^i_{t+1}$$

However, this strategy does not prevent swarm explosion in the general case (Helwig et al., 2012). 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.

- 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 (Helwig & Wanka, 2007; Bansal, 2019). 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 & Michalewicz, 2014b; Cleghorn & Engelbrecht, 2014a; Bansal, 2019).
The aim of this analysis is to define boundaries for the constant parameters in

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

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.

The update rules are sometimes simplified for stability analysis purposes. 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. Also, $\vec{p}_t^i$ and $\vec{g}_t^i$ are assumed to be steady during the run. Further, 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 (Poli et al., 2007).

In addition, validity of the convergence boundaries found when global and personal bests are steady and the topology is ignored were studied by Bonyadi & Michalewicz (2015a); Cleghorn & Engelbrecht (2014b, 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 of 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 (Clerc, 1999; Clerc & Kennedy, 2002).
(ii) First-order stability analysis studies the expectation of the position of particles to ensure that this expectation converges (Trelea, 2003; Van den Bergh & Engelbrecht, 2006; Cleghorn & Engelbrecht, 2014a).
(iii) The second-order stability analysis studies the variance of the position of particles to ensure this variance converges to zero (the convergence of the variance of the particle position is a necessary condition for second-order stability) (Jiang et al., 2007b; Poli, 2009; Liu, 2015) .
(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 & Zhang, 2019).

Perhaps the first study that considered theoretical analysis of the trajectory of particles in PSO was conducted in (Ozcan & Mohan, 1999; Bansal, 2019). 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 $\varphi_1 + \varphi_2 < 4$. Although no parameter analysis or convergence boundaries were introduced by Ozcan & Mohan (1999); Bansal (2019), the findings in that article formed a basis for further analysis.

One of the earliest attempts to analyze the convergence behavior of SPSO to find convergence boundaries was done for PSO variant in (Clerc & Kennedy, 2002; Bansal, 2019). In order to simplify the formulation of update rules, stochastic components (r^i_{1t} and r^i_{2t}) 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

$$\chi = \frac{2k}{\mid 2 - c - \sqrt{c^2 - 4c} \mid} \tag{4.8}$$

where a coefficient known as constriction factor(or χ) is equal to $\omega, \quad c_1 + c_2 = c > 4,\ c_1 = \frac{\varphi_1}{\chi},\ c_2 = \frac{\varphi_2}{\chi}$, and k is a value in the interval $(0,\ 1]$ (usually set to 1 (Clerc & Kennedy, 2002)). 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

$$\vec{V}_{t+1}^i = \chi\{\vec{V}_t^i + \varphi_1 R_{1t}^i(\vec{P}_t^i - \vec{x}_t^i) + \varphi_2 R_{2t}^i(\vec{g}_t - \vec{x}_t^i)\} \tag{4.9}$$

where χ is set by (4.8), $c_1 + c_2 > 4$, and usually $c_1 = c_2 = 2.05$ that results in $\chi = 0.7298$ (Clerc & Kennedy, 2002). 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 (4.9) 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 $\frac{c_2 g + c_1 p}{c_1 + c_2}$).

The stability of particles in SPSO was also analyzed in (Trelea, 2003). 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 $\omega < 1$, $\varphi > 0$, and $2\omega - \varphi + 2 > 0$ where $\varphi = \frac{\varphi_1 + \varphi_2}{2}$. According to Trelea (2003), expectation of particle positions converge to its equilibrium if and only if the value of the parameters are set in such a way that they lie inside the gray triangle (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 by Trelea (2003) including convergence boundaries and convergence to a point between p and g, were also confirmed in (Van den Bergh & Engelbrecht, 2010). However, the speed of convergence was investigated in configuration movement. Subsequently, another first-order stability analysis was conducted by Campana et al. (2010), where the findings of Trelea (2003) were also confirmed. The analysis of convergence to a fixed point was conducted, from a different point of view, in (Jiang et al., 2007b). 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 $\{\frac{\varphi_2 g + \varphi_1 p}{\varphi_1 + \varphi_2}\}$ if and only if $0 \leq \omega < 1$ and $0 < \varphi < 4(1 + \omega)$ 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 (Poli et al., 2007; Poli, 2009; Bonyadi & Michalewicz, 2015b). It was proved by Jiang et al. (2007b) 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 $\varphi < \frac{12(\omega^2-1)}{5w-7}$.

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 by Poli et al. (2007); Poli (2009) 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 $\varphi < \frac{12(\omega^2-1)}{5\omega-7}$ where $\varphi = \varphi_1 = \varphi_2$, that is, the same as what was found in (Jiang et al., 2007a). In addition, Poli (2009) proved that the variance of positions converges to $h(\varphi_1,\ \varphi_2,\ \omega)|g-p|$ where $h(.,.,.)$ is a function of inertia weight and acceleration coefficients. Hence, if $h(\varphi_1,\ \varphi_2,\ \omega) \neq 0$ is guaranteed then particles do not stop moving (non-zero variance) until $p = g$. Therefore, the algorithm is second-order stable only if $p = g$.

The first and second-order stability were investigated by García-Gonzalo & Fernández-Martínez (2014) for SPSO when a generic distribution for ω and (φ_1, φ_2) 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 σ^2_ω and φ_1 and φ_2 (ϕ_1 replaces $\varphi_1 R^i_{1,t}$ and ϕ_2 replaces $\varphi_2 R^i_{2,t}$ in (4.6) are also random variables from an arbitrary probability distribution with the expected values μ_{ϕ_1} and μ_{ϕ_2} and the variances $\sigma^2_{\phi_1}$ and $\sigma^2_{\phi_2}$ respectively. It was proven that SPSO is first-order stable if and only if $-1 < \mu_\omega < 1$ and $0 < \mu_\phi < 2(\mu_\omega + 1)$ where $\phi = \phi_1 + \phi_2$. Also, a necessary condition for the second-order stability is that $-a < \mu_\omega < a$ and $0 < \mu_\phi < b$, where $a = \frac{1}{\sqrt{c^2+1}}$, $b = \frac{2(1-(1+c)\mu^2_\omega)}{1+d^2+(d^2-1)\mu_\omega}$, $c = \sigma_\omega/\mu_\omega$, and $d = \sigma_\phi/\mu_\phi$. 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.

Recently, some studies have considered more general assumptions to find the convergence boundaries under more realistic conditions. For example, Cleghorn & Engelbrecht (2014a) assumed that the personal best of the particles and the global best of the swarm are allowed to move and can occupy an arbitrarily large finite number of unique positions. The main finding of that study was that SPSO was stable in first-order if and only if $-1 < \omega < 1$ and $0 < \varphi < 4(1+\omega)$ where $\varphi = \varphi_1 + \varphi_2 < 4$ under this more general assumption. This is in fact the same as what was found in Trelea (2003) and other previous studies using first-order stability analysis. In article of Cleghorn & Engelbrecht (2014a), it also showed that the topology does not affect the convergence boundaries, but it might affect the speed of convergence or divergence.

The second-order stability of SPSO was also investigated by (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 Jiang et al. (2007b); Poli (2008) are valid under this assumption for the global best particle. Also, 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 (Birbil et al., 2004; Bonyadi & Michalewicz, 2014b). The velocity update rule for SPSO2011 is written as:

$$\vec{V}^i_{t+1} = \omega\vec{v}^i_t + \mathscr{H}_i(\vec{G}^i_t, \|\ \vec{G}^i_t - \vec{x}^i_t\ \|) - \vec{x}^i_t \tag{4.10}$$

where $\mathscr{H}_i(\vec{G}^i_t, \|\ \vec{G}^i_t - \vec{x}^i_t\ \|)$ is a hyper-spherical distribution with the center $\vec{G}^i_t$ and radius $\|\ \vec{G}^i_t - \vec{x}^i_t\ \|$ given that $\vec{G}^i_t = \frac{\vec{L}^i_t + \vec{p}^i_t + \vec{x}^i_t}{3}$, $\vec{P}^i_t = \vec{x}^i_t + \varphi_1(\vec{L}^i_t - \vec{x}^i_t)$ and $\vec{L}^i_t = \vec{x}^i_t + \varphi_2(\vec{l}^i_t - \vec{x}^i_t)$ ($\vec{l}^i_t$ is the best personal best among the particles in T^i_t) (Clerc, 2011). 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 $\vec{V}^i_t$ in SPSO2011 involves generating random points using a hyperspherical distribution with a variance that is dependent on the distance between $\vec{G}^i_t$ and $\vec{x}^i_t$. 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 by Bonyadi & Michalewicz (2015a, 2014b), where it was shown that the 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 SPSO2011 also experimentally showed that the convergence boundaries that guarantee second-order stability are not affected even if the global best and personal best are updated (through a uniform random distribution) during the runs for both SPSO and SPSO2011 (Bonyadi & Michalewicz, 2015a).

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 & 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.

4.4.3 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 (Trelea, 2003; Bonyadi & Michalewicz, 2015a). 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, that is, 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 (4.11), which was mentioned in (Trelea, 2003; Bonyadi, 2019).

$$E(x_{t+1}) + (\varphi - \omega - 1)E(x_t) + \omega E(x_{t-1}) = \varphi P \tag{4.11}$$

where $E(.)$ is the expectation operator, $\varphi = \frac{\varphi_1+\varphi_2}{2}$, and $P = \frac{\varphi_1 p+\varphi_2 g}{\varphi_1+\varphi_2}$. 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,

In addition, experiments by Trelea (2003) 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 to exploit with this setting. Similar analysis for the behavior of particles before convergence was conducted by Campana et al. (2010), where the same oscillation patterns were observed.

The speed of convergence of the expectation of positions was investigated by Van den Bergh & Engelbrecht (2006), where it was proven that the speed of convergence is directly related to $c = max\{||\gamma_1||, \|\gamma_2\|\}$ where, $\gamma_1 = \frac{1+\omega-\varphi_1-\varphi_2+\alpha}{2}$, $\gamma_2 = \frac{1+\omega-\varphi_1-\varphi_2-\alpha}{2}$ and $\alpha = \sqrt{(1+\omega-\varphi_1-\varphi_2)^2 - 4\omega}$ that is, 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 $P = \frac{\varphi_2 g+\varphi_1 p}{\varphi_1+\varphi_2}$) if and only if $c < 1$.

Bonyadi & Michalewicz (2015a) 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 (Trelea, 2003). They showed that the results found by their experimental approach is very similar to what has found in (Trelea, 2003; Bonyadi, 2019). 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.

4.4.4 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

$$\forall \varepsilon > 0,\ \lim_{t\to\infty} Pr(|\vec{p}_t^i - \vec{X}|) \geq 1 - \varepsilon \tag{4.12}$$

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

$$\forall \varepsilon > 0,\ \lim_{t\to\infty} Pr(|\vec{g}_t - \vec{X}|) \geq 1 - \varepsilon \tag{4.13}$$

where $\vec{X}$ is a local optimum and Pr denotes the probability in (Fan & Yan, 2014; Bonyadi & Michalewicz, 2017). Local convergence for PSO variant was first investigated by Van den Bergh & Engelbrecht (2002) in particular for SPSO. In this dissertation, 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 by Van den Bergh & Engelbrecht (2002) 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 and it was proposed by Van den Bergh & Engelbrecht (2002), in which the local convergence issue was addressed. In GCPSO, a new velocity update rule was introduced for the global best particle τ_t (Van den Bergh & Engelbrecht, 2010; Bonyadi & Michalewicz, 2017)

$$\vec{V}_{t+1}^i = \begin{cases} -\vec{x}_t^i + \vec{g}_t + \omega\vec{V}_t^i + \vec{\rho} & if\ \ i = \tau_t \\ \omega\vec{V}_t^i + \varphi_1 R_{1t}^i(\vec{p}_t^i - \vec{x}_t^i) + \varphi_2 R_{2t}^i(\vec{g}_t - \vec{x}_t^i), & otherwise \end{cases} \tag{4.14}$$

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 hyper-rectangle of confidence region while all other particles follow the original position and velocity update rule (Bonyadi & 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 SPSO is not locally convergent while GCPSO is locally convergent, and it could provide acceptable results when the number of particles was set to 2 (Van den Bergh & Engelbrecht, 2010). The local convergence issue for SPSO was also widely discussed by Schmitt & Wanka (2013); Bonyadi & Michalewicz (2017), where it was proven that SPSO is locally convergent for $n = 1$ problems; however, it is not locally convergent when $n > 1$.

4.4.5 Expected First Hitting Time (EFHT)

To characterize the optimization ability of algorithms, we suggest the expected first hitting time (EFHT), that is, 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 SPSO including Clerc & Kennedy (2002); Trelea (2003); Jiang et al. (2007b) have been limited to the concept of convergence to an attractor ($\vec{p}_t$ or $\vec{g}_t$) and not to a particular optimum. EFHT (run time) analysis was conducted on the variant GCPSO because at the moment GCPSO was the only PSO variant that was guaranteed to be locally convergent. The analyzes showed that the EFHT of GCPSO on the sphere function is asymptotically the same as a simple (1+1) ES- for the analysis of EFHT for randomized direct search methods with an isotropic sampling, including (1+1) ES. However, the performance of GCPSO is substantially changed by rotating the search space, and GCPSO is not rotation invariant, while (1+1) ES is (Jägersküpper, 2008).

EFHT analysis for SPSO was also conducted by Lehre & Witt (2013), where the algorithm was applied to some function. It was proven that EFHT for SPSO is potentially infinite even when 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 study was different from that of Schmitt & Wanka (2013), where it was proven that SPSO is locally convergent for one-dimensional problems.

$$\vec{V}^i_{t+1} = \omega \vec{V}^i_t + \varphi_1 R^i_{1t}(\vec{p}^i_t - \vec{x}^i_t) + \varphi_2 R^i_{2t}(\vec{g}_t - \vec{x}^i_t) + \Delta^i_t,$$

where the extra noise term Δ^i_t has uniform distribution on the interval $[-\delta/2,\ \delta/2]$.

A new variant of PSO was introduced by Lehre & Witt (2013) called "Noisy Particle Swarm Optimization", NPSO, in which a uniform randomly generated vector in a hyper-rectangle with 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, that is, NPSO converges to a local optimum of the sphere function in a finite number of function evaluations. 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 by Bonyadi & Michalewicz (2014b) for SPSO2011, where it was proven that, with a specific setting of parameters, the EFHT of the algorithm for any one-dimensional optimization function is infinite, example, SPSO2011 does not guarantee to converge to a local optimum under assumptions that take place with non-zero probability.

4.4.6 Limitations Related to Transformation Invariance

An algorithm $\mathscr{A}$ is invariant under transformation $T, T : \mathbb{R}^d \longrightarrow \mathbb{R}^d$, if the performance of $\mathscr{A}$ on any objective function $F(\vec{x}),\ F : \mathbb{R}^d \longrightarrow \mathbb{R}$, is the same as the performance of $\mathscr{A}$ on $F(T(\vec{x}))$, given that the initial state is "properly " set in both cases (Bonyadi & 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 in the search space.

In addition to invariance, it is also expected that the optimization algorithm is as independent as possible from the initial state (the 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 with which the search space is transformed with T (Hansen et al., 2011).

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 $C(P \in C)$ that are defined by $(P;\ T)$. 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 whether an algorithm is invariant under these transformations (Wilke et al., 2007; Hansen et al., 2011; Bonyadi & Michalewicz, 2014a, 2015a). Transformation $T(\vec{x})$ can be defined by $T(\vec{x}) = sM\vec{x} + \vec{b}$ to formulate these transformations (rotation, scaling, and translation) where $s \in \mathbb{R}$ is a scale factor and $s \neq 0$, $M \in (\mathbb{R}^d$ x $\mathbb{R}^d)$ is a rotation matrix, and $\vec{b} \in \mathbb{R}^d$ is a translation vector.

Transformation invariance has been investigated for many optimization algorithms (Hansen, 2000; Deb & Deb, 2014). Among the transformations of rotation, scale, and translation, rotation has received the most attention from 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 (Hansen et al., 2011).

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 a rotation variant if its performance is changed 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 to the problem at hand. Hence, if the optimal solution is shifted, that assumption is no longer valid and this, in turn, causes a change to the performance of the algorithm.

The transformation invariance of SPSO was first analyzed in 2007, where it was shown that SPSO is scale and translation invariant but is a rotation variant (Wilke et al., 2007). The reason why SPSO is a 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 is rotation, scale, and translation invariant (Bonyadi & Michalewicz, 2017). However, LPSO suffers from a limitation called line search. In LPSO, the particle i oscillates between its personal best and the global best and cannot search in other directions if $CI \parallel SI$ and $\vec{V}_t^i \parallel CI$ hold (Bonyadi, Li, & Michalewicz, 2014; Bonyadi & Michalewicz, 2015a).

Wilke et al. (2007) proposed a new PSO variant that 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 d x d matrix and θ is a scale factor, is a rotation matrix if $(S\theta)^T = (S\theta)^{-1}$ and $det(S\theta) = 1$ with the rotation angle θ (Moler & Van Loan, 2003).

The exponential of a matrix $S\theta$ is defined by:

$$Q = \sum_{i=1}^{maxC} (\frac{1}{i!}(S\theta)^i) + I \tag{4.15}$$

where S is a d x d 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 $S = (P - P^T)$, where P is a d x d matrix with elements generated randomly in the interval $[-0.5, 0.5]$. Also, the angle of rotation in degrees is α (a real scalar) where $\theta = \frac{\alpha\pi}{180}$. It is clear in (4.15) that for a finite value for $maxC$, the matrix Q becomes an "approximation" of a rotation matrix with rotation angle α in (Moler & Van Loan, 2003).

Witt (2009) set the value of $maxC$ to 1, limiting the approximation to only one term of the summation, i.e., $Q = I + S\theta$, so that the time complexity of generating the rotation is reduced. In addition, they claimed that, as rotation is only considered for small values of α, this approximation does not affect the overall results. The time complexity for generating and multiplying the approximated rotation matrix (with $maxC = 1$) in a vector is in $O(d^2)$, including the cost of generating the approximated matrix $(O(d^2))$ plus vector matrix multiplication $(O(d^2))$. For larger values of $maxC$, the order of complexity of the generation of Q becomes larger.

A method for generating accurate random rotation matrices was proposed and investigated by Bonyadi, Michalewicz, & Li (2014), furthermore, 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 hyperplanes. The study tried to include two techniques for the rotation of the velocity vector: rotation within complexity in $O(d^2)$ and rotation within complexity in $O(d)$. Although, empirical results in terms of the objective value showed that rotation in $O(d)$ is just slightly worse than rotation in $O(d^2)$, it is much faster. The random diagonal matrices were replaced by rotation matrices in $O(d^2)$ in many PSO variants proposed by (Van den Bergh & Engelbrecht, 2010; Nickabadi et al., 2011; Bonyadi, Li, & Michalewicz, 2014). 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.

Salomon (1996) investigated that rotating the search space of a problem potentially makes the problem non-separable. The PSO invariant (SPSO) was applied to several optimization problems when their search space was rotated to test if the performance of the algorithm changed (Hansen et al., 2008, 2011; Bonyadi & Michalewicz, 2014a, 2015a). 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 (Hansen & Ostermeier, 1996). In addition, the performance of the algorithm changes when the search space of the problem is rotated, indicating that the algorithm is a rotation variant.

Hansen et al. (2011) also studied the performance of these methods in dealing with ill-conditioned functions. 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 a 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 by Hansen et al. (2011), where the findings presented by Hansen et al. (2008) 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.

It was shown by Spears et al. (2012) that not only SPSO is a rotation variant, but also the particles in SPSO are biased towards the lines parallel to the coordinate axes. In fact, tracking the positions of particles in the search space during the optimization process, it was found that the particles tend to sample more points along the lines parallel to the coordinate axes than other points in the search space, that is, the positions of the particles 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 the PSO algorithm for the variant proposed by Bonyadi & 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 satisfies the condition formulated in $\forall \vec{y} \in S, \exists A_y \subseteq S$, such that $\forall \vec{z} \in A_y, \forall \delta > 0, P(| m(\vec{y}) - \vec{z} |< \delta) > 0$. It was proven that if the operator m generates a new point following the normal distribution then it satisfies the conditions for local convergence and rotation invariance of the algorithm. The time complexity of LcRiPSO using the normal distribution for Q is in $O(d)$, which is faster than the variants proposed in (Wilke et al., 2007; Bonyadi, Michalewicz, & Li, 2014; Bonyadi & Michalewicz, 2017).

4.5 Modifications of Convergence PSO to deal with Unconstrained Optimization Problems

In view of the problems mentioned above, this section proposes 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 to improve the 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 or 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 study convergence of PSO variants that have improved SPSO performance through a specific setting of its parameters, modification of velocity or position update rules, and hybridization of the algorithm.

4.5.1 Parameters Setting

For basic PSO, the parameters have a great impact on the performance of the 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 challenge in iterative optimization methods in the past years (Eiben et al., 1999; Karafotias et al., 2014). 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 the 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 have been further categorized 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 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.

we have studied different parameters for convergence PSO (that is, topology, coefficients, and population size).

4.5.2 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 (Kennedy, 1999; Q. Liu et al., 2016).

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: $T_t^i = \{i, \psi_t\}$ where $\psi_t = \underset{l=\{i-1,\ i,\ i+1\}}{argmin} \{F(\vec{p_t^i})\}$

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: $T_t^1 = \{1, \tau_t\}$ where $\tau_t = \underset{l=\{1,\ldots,n\}}{argmin}\{F(\vec{p_t^i})\}$ and $T_t^{i \neq 1} = \{1\}$

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: $T_t^i = \{i, \tau_t\}$ where $\tau_t = \underset{l=\{1,\ldots,n\}}{argmin}\{F(\vec{p_t^i})\}$.

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 (Mendes et al., 2004).

5. Random edge topology, instead of choosing one of the above fixed topologies, a random neighborhood graph is generated according to some distribution.
Symbolically: $T_t^i = \{i, j\}$ where j is randomly selected among other individuals.

In some PSO variants, the topologies occur not in the described pure forms but as a mixture, the authors M. Hu et al. (2012) use 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 (Clerc, 1999; Kennedy & Mendes, 2002; Mendes et al., 2004; Q. Liu et al., 2016). 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 (Mendes et al., 2004).

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 by Mendes et al. (2004) and later by Karafotias et al. (2014), 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 on (Bonyadi & Michalewicz, 2017).

4.5.3 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 (4.6). Thus, researchers Clerc (1999); Eberhart & Shi (2000); Anthony (2001) 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. 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 by Shi & Eberhart (1998b); Jain et al. (2018) on one test problem–this variant was called "Decreasing Inertia Particle Swarm Optimization", DIPSO. Latter, (based on the average of the objective value) by Y.-l. Zheng et al. (2003) tested an opposite approach, increasing the inertia weight during the run from 0.4 to 0.9, and we found that this idea actually works better than decreasing inertia weight on some test cases.

The concept of decreasing inertia weight was extended by Chatterjee & Siarry (2006) in a way that a non-linear function was used to decrease the value of the inertia weight:

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

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 by Ratnaweera et al. (2004), 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 & Michalewicz, 2017).

4.5.4 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 problems. 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, 2018).

There were researches which determined Probably the first attempt to design an adaptive population size in an evolutionary algorithm in 1994 was presented in (Arabas et al., 1994). The idea of adapting population size was adopted by Mukherjee & Chatterjee (2006) 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 the work of Hsieh et al. (2008), 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, 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. Further, a solution range 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 was used to set the population size of SPSO during the run (De Oca et al., 2009, 2010). 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.

4.5.5 Modification of the Velocity-Position Update Rules

Shi & Eberhart (1998a,b) were the first individual to discuss the parameter inertia weight in basic PSO. 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 forward one step back " stated as all dimensions of the position of particles are up-dated 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 (Van den Bergh & Engelbrecht, 2004). 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, Van den Bergh & Engelbrecht (2004) was proposed to divide the search space to d sub-problems and different variables are updated by different sub-swarm, 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 sub-problems.

As SPSO does not suffer from these issues, a hybrid method was also proposed by Van den Bergh & Engelbrecht (2004) 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 presented by Arumugam et al. (2009), in which the position update rule was stated given as follows:

$$\vec{x}^i_{t+1} = \vec{g}_t + \alpha\vec{g}_t + \gamma(\vec{g}_t - \vec{x}) \tag{4.16}$$

where

$$\alpha = k_1 r^i_t, \;\; \gamma = k_1 e^{k_2\beta}, \;\; k_1 = k_2 = e^{\left(\frac{-current\ iteration}{max\ number\ of\ iterations}\right)},$$

r^i_t is a random value between 0 and 1, and $\beta = \frac{f(\vec{g}_t) - f(\vec{x}^i_t)}{f(\vec{g}_t)}$. The term $\vec{g}_t + k_1 r^i_t \vec{g}_t$ generates a random point around the global best vector. Note that, using limit concept as $k_1 \longrightarrow 0$ over time (iterations), the generated points by $(\vec{g}_t + k_1 r^i_t \vec{g}_t) \longrightarrow \vec{g}_t$ that results in better exploitation around the global best vector at the later stages of the run. The last term $k_1 e^{k_2\beta}(\vec{g}_t - \vec{x}^i_t)$ determines the step size that the current position $(\vec{x}^i_t)$ 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 (Janson & Middendorf, 2005).

In CLPSO the velocity update rule was modified to the movement equation :

$$\vec{V}_{t+1}^i = \omega \vec{v}_t^i + \varphi R_t^i (\vec{p}_t^i - \vec{x}_t^i) \tag{4.17}$$

where the value of the j^{th} dimension of $\vec{p}_t^i(\vec{p}_t^{i,j})$ is calculated by $\vec{p}_t^{i,j} = p_t^{l_i(j),j}$ where $l_i(j) \in \{1, 2, \ldots, n\}$ (index of a particle). In fact, each dimension of $\vec{P}_t^i$ is potentially taken from the personal best of different particles (Bonyadi & 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 $l_i(j)$ was set to i with the probability Pr while it was set through a tournament selection to select the value of $l_i(j)$ 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. Also, 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. A new variant of PSO called "Example-based Learning Particle Swarm Optimizer ", ELPSO was introduced by Huang et al. (2012), where (4.17) was given as follows:

$$\vec{V}_{t+1}^i = \omega \vec{v}_t^i + \varphi_1 R_{1t}^i (\vec{p}_t^i - \vec{x}_t^i) + \varphi_2 R_{2t}^i (\vec{G}_t - \vec{x}_t^i) \tag{4.18}$$

where $\vec{P}_t^i$ is defined similarly to the one in CLPSO (except the strategy of setting $l_i(j)$ which was selected randomly by 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 by 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 by Xinchao (2010) 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 was substituted for $N(\vec{g}_t, \sigma^2 I)$, 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, 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, that is, rotation variance with scale and translation invariance .

The behavior of the particles especially related to the exploitative and explorative is diversity of particles. An algorithm variant of PSO called "Diversity enhanced with Neighborhood Search Particle Swarm Optimization", DNSPSO, was proposed by Van den Bergh & Engelbrecht (2006) 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 $\vec{x}^i_{t+1}$ is then combined with $\vec{x}^i_t$ to generate a trial particle. This is done by taking either the value of $x^{i,j}_t$ or $x^{i,j}_{t+1}$ for each $j \in \{1, \ldots, d\}$ with probability Pr. The personal best of the particle i is updated according to the trial particle. Also, 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 was shown to be 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 & Michalewicz, 2017).

4.5.6 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 based on the integration of other approaches, viz., the so-called hybrid soft computing (HSC). For instance using one method to adjust 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 by Parsopoulos & Vrahatis (2002); Kao & Zahara (2008); Alatas et al. (2009) where a GA was combined with SPSO.

A novel particle movement named "oscillation"was presented by Parsopoulos & Vrahatis (2004), which introduced the bond of each particle attracted by the personal best and global best vectors. Additionally, the approach of 'two steps forward, one step back' was investigated in (Zhan et al., 2009). 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 (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 (4.17) 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.

De Oca et al. (2009) was presented different strategies to improve a better topology and adaptive coefficients of the PSO algorithm for updating each component is 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 proposed in (Janson & Middendorf, 2005).

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

Ardizzon et al. (2015) proposed a novel example of the basic particle swarm concept, with two types of agents in the swarm, the "explorers" and the "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 & Yan (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, 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 is 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. A very similar concept by the method was called "Self-Learning Particle Swarm Optimization", SLPSO presented in (C. Li et al., 2011). The only difference between idea of Self Learning and Self Adaptive Learning was in the implementation of PSO algorithm such as how to update probabilities and which velocity update rules are used in details.

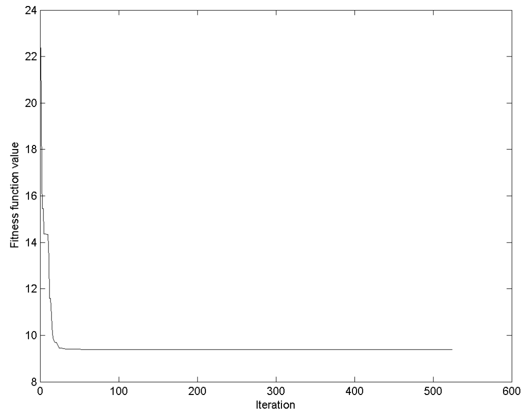
Roy & Ghoshal (2008) proposed Crazy PSO in which particle velocity was randomized within predefined limits. Its aim was to randomize the velocity of some particles, named as "crazy particles" through using a predefined probability of craziness to keep the diversity for global search and better convergence. The ideas used in SALPSO and SLPSO triggered the emergence of another PSO variant called "Multiple Adaptive Methods for Particle Swarm Optimization", MAM-PSO (Helwig & Wanka, 2007).

In MAM-PSO, all particles were updated by the update rules in SPSO except the global best particle. A mutation operator was proposed by Van den Bergh & Engelbrecht (2010) 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 moving the particle according to its velocity. To update the global best particle, two operators approach, a mutation and a gradient descent operator approach, were designed and one of them was selected randomly at each iteration.

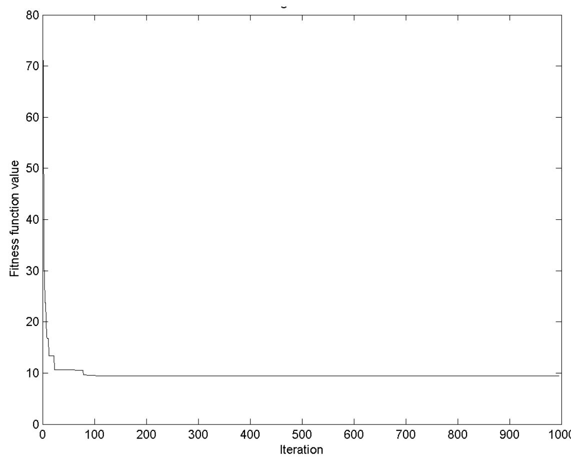
An extensive experiment was conducted that showed MAM-PSO is comparable with other PSO variants such as CPSO and CLPSO, which proved the effectiveness of the objective function for convergence analysis. We conjecture that the algorithm is locally convergent as it uses a mutation operator based on a Cauchy distribution with the center of the current position and a non-zero scale parameter. Furthermore, Medasani & Owechko (2005) 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 such as Muller et al. (2009), where SPSO was combined with CMA-ES.

4.6 Numerical Simulations

The authors could experimentally showed that the performance of PSO significantly depends on the best values for the ω lies in $[0.4, 0.9]$ while the interval $[1.5, 3]$ holds the best value for φ_1 and φ_2. The standard swarm parameters for the experiments of most studies are $\omega = 0.72984,\ c_1 = c_2 = 1.496172$. Figure 4.1(a) 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 4.1(b) depicts the convergence of PSO with inertia weight increased linearly from 0.4 to 0.9 during the run and there is no improvement in its best value even after 1000 iterations. Figure 4.2 (a) depicts



(a) PSO for ω runs [0.9, 0.4]

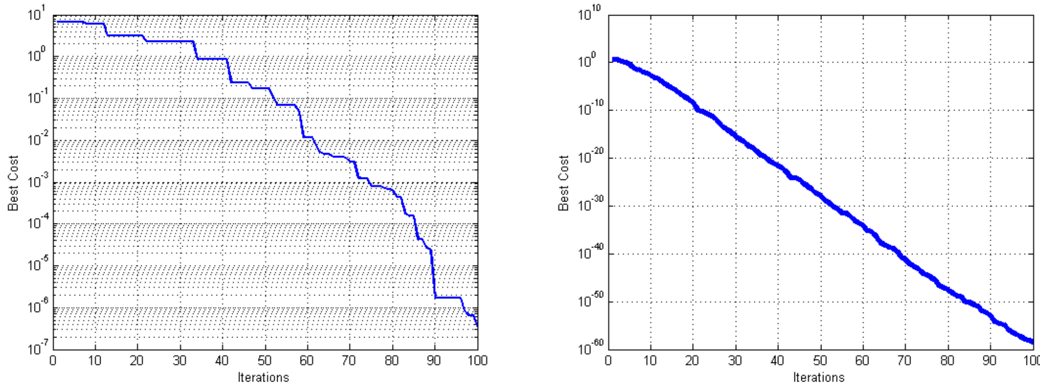


(b) PSO for ω runs [0.4, 0.9]

Figure 4.1: Evaluation of PSO algorithms with coefficient, ω

the convergence curve of PSO with inertia weight. SPSO, on the other hand, shows step wise improvement in its best solution over the iterations. It converges at the best optimal value found so far by all the algorithms. Figure 4.2 (b) also depicts the comparison result of convergent of SPSO with constriction coefficient and have these main advantages

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



(a) PSO with Inertia weight

(b) PSO with Constriction coefficient

Figure 4.2: Evaluation of PSO algorithm with coefficient, χ

CHAPTER 5

CONVERGENCE ANALYSIS OF PARTICLE SWARM OPTIMIZATION ALGORITHMS FOR DIFFERENT CONSTRICTION FACTORS

5.1 Introduction

The optimization techniques are fundamentally important in engineering and sciences. The PSO algorithm was first introduced by Kennedy & Eberhart (1995) as a stochastic optimization technique of swarm particles (population). The motivation was primarily to model the social behavior of birds flocking. The meta-heuristic optimization algorithms (PSO) work effectively in many areas such as robotics, wireless networks, power systems, job-shop schedules, human health care and classifying or training of ANN (artificial neural network), and so on (Gad, 2022). In PSO, the potential solutions, called particles, fly through the problem space (domain) by applying their intelligent collective behaviors.

SPSO (Standard Particle Swarm Optimization) is a variant of the original PSO algorithm proposed by (Kennedy & Eberhart, 1997). It introduced modifications to improve the convergence properties and overall performance of the algorithm. SPSO has since become a popular choice in the field of optimization and has been widely applied to solve a diverse range of problems. The PSO algorithm is competitive in performance with its other well-known many variants such as SPSO, and CPSO algorithms and is also an efficient optimization framework (Banks et al., 2007; AlRashidi & El-Hawary, 2008; Bonyadi & Michalewicz, 2014a).

Lately, researches on PSO mainly intended on algorithmic implementations, enhancements, and engineering applications with interesting findings derived under the system that assumes a fixed attractor (A. Kumar et al., 2016). Nevertheless, a comprehensive mathematical explanation for the general PSO is still quite limited. For instance, the works on stability and convergence analyses are two key problems of great significance that need to be investigated in depth because many of the works have given attention for standard PSO. The PSO algorithm depends on three parameters (factors): the inertial, the cognitive and social weight, to guarantee the stability of PSO. Stability analysis of PSO is mainly motivated by determining which combination of these parameters encourages convergence.

The working rule of PSO method is closely connected with the stability analysis, which investigates how the essential factors affect the swarms dynamics, and under what conditions particle swarm converges to some fixed value. For the first time, stability analysis of the particle dynamics was carried out in (Clerc & Kennedy, 2002). The study by Van den Bergh & Engelbrecht (2006) indicates that particle trajectories could converge to a stable point. A more generalized stability analysis of the particle dynamics was conducted using the Lyapunov stability theorem (Kadirkamanathan et al., 2006). Recently, based on a weak stagnation assumption Liu (2015) studied the order-2 stability of PSO, and a new definition of stability was proposed with an order-2 stable region.

Dong & Zhang (2019) analyzed order-3 recurrence relation of PSO kinematic equations based on two strategies to obtain the necessary and sufficient conditions of its convergence. The convergence analysis determines, if a global optimum solution can be achieved when a particle swarm converges. Using stochastic process theory Jiang et al. (2007b) presented a stochastic convergence analysis on the standard PSO. Combining with the finite element grid technique, Poli & Langdon (2007) set up a discrete model of Markov chain of the bare-bones PSO.

An absorbing Markov process model of particle swarm optimization (PSO) was developed by Cai et al. (2009). They identified the attaining-state set as a key factor in convergence analysis and proposed an enhanced method for evaluating convergence based on the expansion theorem related to the attaining-state set. The basic PSO is neither a global nor a local search algorithm, based on the convergence criterion of the pure random search algorithm (Solis & Wets, 1981; Van den Bergh & Engelbrecht, 2010).

To give a lower bound for the time required to optimize any pseudo-Boolean function with a unique optimum and to justify the upper bounds, Sudholt & Witt (2010) assigned an argument of optimum level that is deep-rooted for evolutionary algorithms of particle swarm optimization. The work of Sun et al. (2012) discussed the convergence of quantum-behaved particle swarm optimization (QBPSO) and proved that it is a global convergence algorithm.

As discussed in stagnation of the convergence properties for basic PSO may be disadvantageous in finding a sufficiently good solution within a logical time, and it may have an infinite expected first hitting time on some functions (Lehre & Witt, 2013). Recently, the existing work on PSO convergence analyses of including documents from 2013 was surveyed. The stochastic approximation technique in the PSO algorithm was used to prove the convergence of swarm in (Yuan & Yin, 2014). The global convergence of PSO, Armaghani et al. (2014) was investigated by introducing the transition probability of the particles. Several properties related to the Markov chain were investigated, and it was found that the particle state space is not repeated and the PSO is not globally convergent from the point of view of the transition probability (Xu & Yu, 2018).

Based on the different models of PSO examined by Feng et al. (2013), the Markov properties of the state sequences of a single particle and the swarm one determine the transition probability of a particle. The transition probability of the optimal set is deduced by combining the law of total probability with the Markov properties of Sun et al. (2004); Flori et al. (2022), which proves that SPSO can reach the global optimum in probability. Although many methods by Poli & Langdon (2007) have proposed PSO convergence analysis, most analyzes are based on the assignment of stochastic systems of the Markov process, which strongly depends on the transition matrix and their eigenvalues. Therefore, when the population size is large, current PSO convergence analyses are very refined and investigate different PSO variants algorithms to obtain a solution that converges to global minimum.

Building on our recent work on the convergence analysis of PSO, we propose a variant known as CSPSO (Constriction Standard Particle Swarm Optimization), an algorithm designed for solving optimization problems. In construction, a constriction factor integrated with an inertia weight is used in the construction, resulting in a fast-convergence method that reaches an optimal solution within the search space in a limited number of iterations.

5.2 The PSO Algorithm and Some Related Studies

In the PSO with K particles in which each particle is treated as an individual in the D-dimensional space, the position and velocity vectors of the i-th particle in the t-th iteration are
$X_i^t = (X_{i1}^t, X_{i2}^t, \ldots, X_{iD}^t)$ and $V_i^t = (V_{i1}^t, V_{i2}^t, \ldots, V_{iD}^t)$, respectively.
In the SPSO algorithm, as described by Shi & Eberhart (1999); Bonyadi & Michalewicz (2014a), at iteration t, the dth dimension of the particle i's velocity and position P_i^t represents the local best position, x_i^t is the current position, and g^t is the global best position. Both are updated as

$$\begin{aligned} V_i^{t+1} &= \omega V_i^t + c_1 r_1^t (P_i^t - X_i^t) + c_2 r_2^t (g^t - X_i^t), \\ X_i^{t+1} &= X_i^t + V_i^{t+1}, \end{aligned} \tag{5.1}$$

for $1 \leq i \leq K$; ω is an inertia weight; and c_1 and c_2 are called acceleration coefficients in real-space, $\mathbb{R}$.

Vector $P_i^t = (P_{i1}^t, P_{i2}^t, \ldots, P_{iD}^t)$ is the best previous position of the particle i called the personal best position (Pbest), and vector $g^t = (g_1^t, g_2^t, \ldots, g_D^t)$ is the position of the best particle among all the particles in the population and called global best position (gbest). The parameters r_1^t and r_2^t are sequences of two different random positive numbers in the uniform random distribution in $(0,\ 1)$ i.e., $U(0,1)$.

Generally, the value of V_{id}^t is restricted within the interval $[-V_{max},\ V_{max}]$, for each $d \in \{1, 2, \ldots, D\}$. Without loss of generality, we usually consider the minimization problem. Minimize $f(X)$, such that

$$X \in S \subset \mathbb{R}^D, \tag{5.2}$$

where $f(X)$ is an objective function continuous almost everywhere and S is a feasible solution space. From (5.1) the non-homogeneous recurrence relation (NHRR) is obtained by Van den Bergh & Engelbrecht (2006)

$$X_i^{t+1} = -\omega x_i^{t-1} + (1+\omega)x_i^t + \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t), \tag{5.3}$$

where $\varphi_1^t = c_1 r_1^t,\ \varphi_2^t = c_2 r_2^t$.
From NHRR, P^t and g^t, for $1 \leq i \leq K$ are updated, respectively as follows:

$$P_i^{t+1} = \begin{cases} x_i^{t+1} : for\ f(x_i^{t+1}) < f(P_i^t) \\ P_i^t \quad : otherwise; \end{cases} \tag{5.4}$$

$$g^0 = \underset{1 \leq i \leq k}{arg\ min}\{f(x_i^o)\}, \quad g^{t+1} \quad = \underset{1 \leq i \leq k}{arg\ min}\{f(x_i^{t+1}), f(g^t)\}.$$

From (5.2) and (5.3), the process of the particle's velocity and position change can be obtained, respectively as follows. They are a second-order difference equations

$$V_i^{t+2} + (\varphi^t - \omega - 1)V_i^{t+1} + \omega V_i^t = 0 \tag{5.5}$$

$$X_i^{t+1} + (\varphi^t - \omega - 1)x_i^t + \omega x_i^{t-1} = \varphi_1^t P_i^t + \varphi_2^t g^t = \varphi^t O_i^t \tag{5.6}$$

where, $\varphi^t = \varphi_1^t + \varphi_2^t$

$$O_i^t = \frac{\varphi_1^t p_i^t + \varphi_2^t g^t}{\varphi^t}. \tag{5.7}$$

The terms $(\varphi^t - \omega - 1)x_i^t$ and ωx_i^{t-1} on the left side of (5.6), both memorize the past values of position (that is, the memory item of position). Since the value of the item $\varphi^t O_i^t$ on the right side of (5.6) is obtained from the previous experience of particles (that is, the learning item of position), and in particular O_i^t is the attractor at the t th iteration in (5.7).

One lets

$$Q_x = \max_{x^t \in S_x \subset R} \mid x(t) \mid . \tag{5.8}$$

For $p^t,\ g^t \in S_x,\ |p^t| \leq Q_x$ and $|g^t| \leq Q_x$. From (5.8), $O^t \in S_o$ means $|O^t| \leq Q_o$ for all t.

Constriction in SPSO introduces a constriction coefficient, typically represented by the symbol χ, which controls the balance between the cognitive and social components. The constriction coefficient restricts the velocities of particle within a certain range to prevent excessive exploration or exploitation.

The velocity update equation in SPSO with constriction (CSPSO) can be described as follows;

$$V_i^{t+1} = \chi * (\omega V_i^t + \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t)).$$

5.2.1 Convergence of Some PSO Variants

The importance of a hybrid method is to combine different optimization methods to take advantage of the virtues of each of the methods. In addition to standard PSO, several variants of the algorithm were developed by A. Kumar et al. (2016) to improve the performance of PSO.

The SPSO

$$X_i^{t+1} = X_i^t + V_i^{t+1},$$

has a scalar function of position if $x_i^t = p_i^t = g_i^t$ for a particle, that is particle's update depends only on its previous velocity. This can make the algorithm to stop to flow on the swarm's global best position, even if that position is not a local optimum. For instance, based on (5.4), the guaranteed convergence PSO, GCPSO, overcomes this problem by using a modified position and velocity update equation for the global best particle, which forces that particle to search for a better position in a confined region around the global best position.

The GCPSO can be used with neighborhood topologies such as star, ring, and Von Neumann. Neighborhoods have a similar effect in the GCPSO, as observed by Van den Bergh & Engelbrecht (2010); Lehre & Witt (2013), as they do in the SPSO. Shi & Eberhart (1999) introduced the concept of linearly decreasing inertia weight with generation number into PSO to improve algorithmic performance.

The particles converge to a weighted average (O_i^t) between their personal and local best positions as stated by Van den Bergh & Engelbrecht (2006), which is called the theoretical attractor point (ATP). Kennedy (2003) has proposed that the entire velocity update equation is replaced by a random number sampled from a Gaussian distribution(Gd) around the ATP, with a deviation of the magnitude of the distance between the personal and global best. The resultant algorithm is called the bare bones PSO (BBPSO).

Kennedy also proposed an alternative bare bones PSO (aBBPSO) in 2003, where the particle sampled from the previous Gd is reunited with the particle's personal best position. The performance of PSO with a small and larger nearby region might be better in multimodal and unimodal problems, respectively (Kennedy & Mendes, 2002). Dynamically changing neighborhood structures has been proposed to avoid insufficiencies in fixed nearby regions (Liang & Suganthan, 2005).

The quantum-behaved particle swarm optimization was proposed to show many advantages to the traditional PSO. Sun et al. (2004) proposed a quantum-behaved particle swarm optimization (QBPSO) algorithm and discuss the convergence of QBPSO within the framework of the random algorithm's global convergence theorem. Inspired by natural speciation, some researchers have introduced evolution methods into PSO (Parrott & Li, 2006; Brits et al., 2007). The problem of premature convergence was studied on a perturbed particle swarm algorithm presented based on the new particle update strategy (Xinchao, 2010). To solve optimization problems, Tang et al. (2011) developed a feedback-learning PSO algorithm with a quadratic inertia weight, ω.

Hybridized PSO with a local search technique can be referred to for locating optimal solutions for multiple global and local solutions in physical fitness of more than one global optimal solution for the optimization problem using a memetic algorithm (H. Wang et al., 2012). An example-based learning PSO was proposed by Huang et al. (2012) to overcome PSO failures by maintaining a balance between swarm diversity and convergence speed.

A variation of the global best PSO where the velocity update equation does not hold a cognitive component is called social PSO, expressed as

$$V_i^{t+1} = \omega V_i^t + \varphi_2(g_i^t - x_i^t), \tag{5.9}$$

The individuals are only supported by the global best position and their previous velocity. The particles are attracted towards the global best position, instead of a weighted average between the global best position and their personal best positions, leading to very fast convergence (Lehre & Witt, 2013; Bonyadi & Michalewicz, 2014a).

5.3 Relations of CSPSO and Markov chain

The global convergence of CSPSO is analyzed based on properties of Markov chain and the transition probabilities of particle velocity and position are also computed.

$$V_i^{t+1} = \begin{cases} \chi * (V_i^t + \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t)), \ for\ \omega = 1 \\ \chi * (\omega V_i^t + \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t)), \ otherwise \end{cases} \tag{5.10}$$

In (5.10), the velocities of the particles are updated using two main components: the cognitive component $(P_i^t - x_i^t)$ and the social component $(g^t - x_i^t)$. The cognitive component guides a particle towards its personal best position, while the social component directs a particle towards the best position found by the entire swarm.

We introduce some notation and recall some useful definitions and results that are needed for the rest of the chapter. Enhance comprehension, development, and advancement of knowledge in the optimization field. The following definitions provide a formal description of this property based on a single particle model (Feng et al., 2013; Xiao & Yin, 2014; Xu & Yu, 2018; Lawler, 2018).

Definition 1. (Stochastic process and Markov property).
Let all the variables be defined within the context of a common probability space or probability measure;

1. The random variables $Y = (Y^0, Y^1, \ldots, Y^t)$ in a sequence are called stochastic processes.
2. Let Y^t be a value in state space S, and the sequence $\{Y^t\}_{t\geq 0}$ is a discrete stochastic process. For every $t \geq 0$ and $i^l \in S(l-1 \leq t)$.
3. The discrete stochastic process is a Markov chain.
If the probability $Pr\{Y^{t+1} = i^{t+1} \mid Y^0 = i^0,\ Y^1 = i^1, \ldots, Y^t = i^t\} = Pr\{Y^{t+1} = i^{t+1} \mid Y^t = i^t\} > 0$, and $Pr\{Y^0 = i^0,\ Y^1 = i^1, \ldots, Y^t = i^t\} > 0$.

Definition 2. (State of particle). The state of particle $\kappa_i^t = (x_i^{t-1},\ x_i^t,\ p_i^t,\ g^t)$ at the t-th iteration of particle i in (5.3).
The state of particle space is a set of all possible states of particle, denoted as S.
κ_i^t, the update probability of the state of the particle can be calculated based on Proposition-1.
Proposition 1. If the accelerating factors $\varphi_1^t,\ \varphi_2^t$ in CSPSO satisfy $\varphi_1^t,\ \varphi_2^t \in U(0, c)$, then the probability for particle i changes from the position x_i^t to the spherical region centered at x_i^{t+1} with radius $\varrho_t > 0$. The event $A_i = \{\kappa_i^{t+1} \mid \kappa_i^t\}$, defining the state of the particle i in the t-th iteration is updated to the state in the $(t+1)$-th iteration, for each $i \in \{1, 2, \ldots, K\}$ can be

computed as

$$Pr(A_i) = \frac{\varrho_t^3}{\chi\omega \mid\mid x_i^t - X_i^{t-1} \mid\mid \;\; c\chi \mid\mid p_i^t - X_i^t \mid\mid c\chi \mid\mid g^t - X_i^t \mid\mid}, \tag{5.11}$$

c is a constant within $U(0, c)$ and $\delta \to 0$, where

$$\varrho = c\chi * \begin{cases} \mid\mid p_i^t - X_i^t \mid\mid, \; for \; f(x_i^t) - \delta \le f(P_i^t) \le f(x_i^t) + \delta \\ \mid\mid g^t - X_i^t \mid\mid, \; for \; f(x_i^t) - \delta \le f(g^t) \le f(x_i^t) + \delta \end{cases} \tag{5.12}$$

proof. The 1-step transition probability of the i th state of particle, P_i^{t+1} and g^{t+1} are determined by x_i^{t+1} for transferring κ_i^t to κ_i^{t+t} based on the following SPM-Single Particle Model by Xiao & Yin (2014)

$$x_i^{t+1} = x_i^t + \chi * (\omega(x^t - x_i^{t-1}) + \varphi_1^t(P_i^t - x_i^t) + \varphi_2^t(g^t - x_i^t)), \tag{5.13}$$

x_i^{t+1} determined by $\chi\omega, \; \chi\varphi_1$ and $\chi\varphi_2$.

Three conditions in the 1-step transition probability are as follows:

1. $f(x_i^t) - \delta \le f(P_i^t), \; f(g^t) \le f(x_i^t) + \delta$. $x_i^{t+1} = x_i^t + \chi\omega(x_i^t - x_i^{t-1})$ is determined uniquely by $\chi\omega$ where $\chi\omega$ is unknown constant, having

$$P(x_i^{t+1} \mid \kappa_i^t) = \frac{\int_{x_i^{t+1} - \frac{1}{2}\varrho}^{x_i^{t+1} + \frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t + \chi\omega(x_i^t - x_i^{t-1})} dy} = \frac{\varrho}{\chi\omega \mid\mid x_i^t - x_i^{t-1} \mid\mid} \tag{5.14}$$

2. $f(x_i^t) - \delta \le f(P_i^t), \; f(g^t) \le f(x_i^t) + \delta$.
   Ordering implies $g^t \in \{p_i^t, \; g^t\}$

$$x_i^{t+1} = x_i^t + \chi\omega(x^t - x_i^{t-1}) + \chi\varphi_2^t(g^t - x_i^t) \tag{5.15}$$

   Here φ_2^t is random variable because x_i^{t+1} is determined by $\chi\omega(x^t - x_i^{t-1})$ and $\chi\varphi_2^t(g^t - x_i^t)$

$$\begin{aligned} P(x_i^{t+1} \mid \kappa_i^t) &= \frac{\int_{x_i^{t+1} - \frac{1}{2}\varrho}^{x_i^{t+1} + \frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t + \chi\omega(x_i^t - x_i^{t-1})} dy} * \frac{\int_{\varphi_1^t - \frac{1}{2}\varrho}^{\varphi_i^t + \frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t + \chi c(g^t - x_i^t)} dy} \\ &= \frac{\varrho}{\chi\omega \mid\mid x_i^t - x_i^{t-1} \mid\mid} * \frac{\varrho}{\chi c \mid\mid g^t - x_i^t \mid\mid} \end{aligned} \tag{5.16}$$

3. $f(x_i^t) + \delta < f(P_i^t), \; f(g^t) < f(x_i^t) - \delta$.

$$x_i^{t+1} = x_i^t + \chi\omega(x_i^t - x_i^{t-1}) + \chi\varphi_1^t(P_i^t - x_i^t) + \chi\varphi_2^t(g^t - x_i^t)) \tag{5.17}$$

when $x_i^{t+1} \in \mathbb{R},\ x_i^{t+1}$ is determined by $\chi * (\omega, \varphi_1, \varphi_2)$

$$\begin{aligned} P(x_i^{t+1} \mid \kappa_i^t) &= \frac{\int_{x_i^{t+1}-\frac{1}{2}\varrho}^{x_i^{t+1}+\frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t+\chi\omega(x_i^t-x_i^{t-1})} dy} * \frac{\int_{\varphi_1^t-\frac{1}{2}\varrho}^{\varphi_1^t+\frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t+\chi c(P_i^t-x_i^{t-1})} dy} * \frac{\int_{\varphi_2^t-\frac{1}{2}\varrho}^{\varphi_2^t+\frac{1}{2}\varrho} dy}{\int_{x_i^t}^{x_i^t+\chi c(g^t-x_i^{t-1})} dy} \\ &= \frac{\varrho}{\chi\omega \mid\mid x_i^t - x_i^{t-1} \mid\mid} * \frac{\varrho}{\chi c \mid\mid P_i^t - x_i^t \mid\mid} * \frac{\varrho}{\chi c \mid\mid g^t - x_i^t \mid\mid} \end{aligned} \tag{5.18}$$

From conditions in $1 - 3$,

$$\lim_{P_i^t \to X_i^t} \frac{\varrho}{\chi\omega \mid\mid P_i^t - X_i^t \mid\mid} = 1$$

$$\lim_{g^t \to X_i^t} \frac{\varrho}{\chi c \mid\mid g_i^t - X_i^t \mid\mid} = 1.$$

$$P(A_i) = \frac{\varrho^3}{\chi(\omega \mid\mid x_i^t - x_i^{t-1} \mid\mid c \mid\mid P_i^t - x_i^t \mid\mid c \mid\mid g^t - x_i^t \mid\mid)} \tag{5.19}$$

δ is a vector approaching to zero. When

I. ϱ is the radius of x_i^{t+1}

II. $f(x_i^t) - \delta \leq f(P_i^t) \leq f(x_i^t) + \delta,\quad \varrho = c\chi \mid\mid P_i^t - x_i^t \mid\mid$

III. $f(x_i^t) - \delta \leq f(g^t) \leq f(x_i^t) + \delta,\ \varrho = c\chi \mid\mid g^t - x_i^t \mid\mid$

Definition 3. (State of the swarm). The state of swarm in (5.3), in iteration t, denoted as η^t, is defined as $\eta^t = (\kappa_1^t,\ \kappa_2^t, \ldots, \kappa_K^t)$.

The state of swarm space is a set of all possible states of swarm, denoted as ϖ (Xu & Yu, 2018).

Proposition 2. (Markov chain). The set of collection of the swarm state $\{\eta^t\}_{t\geq 1}$ is a Markov chain (Feng et al., 2013).

Proof. The proof follows by referring to the equation of position that the state of swarm $\eta^{t+1} = (\kappa_1^{t+1},\ \kappa_2^{t+1}, \ldots, \kappa_K^{t+1})$ at iteration $t+1$ depends on only the state of swarm $\eta^t = (\kappa_1^t,\ \kappa_2^t, \ldots, \kappa_K^t)$. at iteration t. Therefore, $\{\eta^t\}_{t\geq 1}$ is a Markov chain.

Definition 4. Let Γ_1^n denote the σ-field generated by particles state $\kappa_1^t,\ \kappa_2^t, \ldots, \kappa_n^t,\ (K \geq n)$ and define

$$\phi(\Gamma_1^n,\ \kappa_{n+1}^t) = sup\{\mid Pr(B \backslash A) - Pr(B) \mid : A \in \Gamma_1^n,\ B \in \sigma(\kappa_{n+1}^t)\}, \tag{5.20}$$

$$\phi = \underset{1 \leq n \leq K-1}{sup} \phi(\Gamma_1^n,\ \kappa_{n+1}^t) \tag{5.21}$$

Due to the weak interdependent relationship among the particles, ϕ is approximately small.

Proposition 3. The transition probability from η^t to η^{t+1} satisfies

$$\mid Pr(\eta^{t+1} \mid \eta^t) - \prod_{i=1}^{K} Pr(\kappa_i^{t+1} \mid \kappa_i^t) \mid \leq \mu \tag{5.22}$$

where μ can be made small enough, therefore, $\mu = (2^{K-1} - 1)\phi$.

Proof. Based on the Definition 4, one has $\mid Pr(B \backslash A) - Pr(B) \mid \leq \phi$.

The event $\{\kappa_i^{t+1} \mid \kappa_i^t\}$ denoted as A_i means that the state of particle i at the t-th iteration is changed to the state at the $(t+1)$-th iteration, for each $i \in \{1, 2, \ldots, K\}$.

$$Pr(\prod_{i=1}^{K} A_i) = Pr(\eta^{t+1} \mid \eta^t) \tag{5.23}$$

is the transition probability from η^t to η^{t+1}. Because g^{t+1} depends on x_i^t and P_i^t for all $1 \leq i \leq K,\ A_1,\ A_2, \ldots, A_K$ are not independent random events. According to (5.6) and the conditional probability, one has the following cases:

Case 1: $Pr(A_1A_2) = Pr(A_1)Pr(A_2 \mid A_1) \leq Pr(A_1)[P(A_2) + \phi] \leq Pr(A_1)Pr(A_2) + \phi,$

Case 2: $Pr(A_1)Pr(A_2) - \phi \leq Pr(A_1)[Pr(A_2) - \phi] \leq Pr(A_1)Pr(A_2 \mid A_1) = Pr(A_1A_2).$

This implies

$$Pr(A_1)Pr(A_2) - \phi \leq Pr(A_1A_2) \leq Pr(A_1)Pr(A_2) + \phi \tag{5.24}$$

Case 3: $Pr(A_1A_2A_3) = Pr(A_1)Pr(A_2 \mid A_1)Pr(A_3 \mid A_1A_2) \leq Pr(A_1)[Pr(A_2) + \phi][Pr(A_3) + \phi]$
$\leq Pr(A_1)Pr(A_2)Pr(A_3) + 3\phi,$

Case 4: $Pr(A_1)Pr(A_2)Pr(A_3) - 3\phi \leq Pr(A_1)[Pr(A_2) - \phi][Pr(A_3) - \phi] \leq Pr(A_1)Pr(A_2 \mid A_1)$
$Pr(A_3 \mid A_1A_2)$
$= Pr(A_1A_2A_3)$

Similarly we can get

$$Pr(A_1)Pr(A_2)P(A_3) - 3\phi \leq Pr(A_1A_2A_3) \leq Pr(A_1)Pr(A_2)Pr(A_3) + 3\phi \tag{5.25}$$

Then,

$$\prod_{i=1}^{K} Pr(A_i) - (2^{K-1} - 1)\phi \ \leq Pr(\prod_{i=1}^{K} A_i) \leq \prod_{i=1}^{K} Pr(A_i) - (2^{K-1} - 1)\phi \tag{5.26}$$

is the transition probability from η^t to η^{t+1}. Let $\mu = (2^{K-1} - 1)\phi$. We have

$$| \, Pr(\eta^{t+1} \mid \eta^t) - \prod_{i=1}^{K} P(\kappa_i^{t+1} \mid \kappa_i^t) \, | \leq \mu \tag{5.27}$$

From source (5.1), the interdependent relationship among the particles is weak, ϕ in (5.21) is sufficiently small so that the fact that K is finite implies that μ is a small enough positive number.

5.3.1 The CSPSO Algorithm and Its Probabilistic Convergence Analysis

We present the convergence analysis for the version of the standard PSO with constriction coefficient (CSPSO), by analogy to the method of analyzing the convergence of the PSO in (Kennedy & Mendes, 2002). We also based on concepts of definitions and results in Section 5.3 above. Our analysis has the advantage of providing a much easier method to realize the convergence of the PSO with the constriction coefficient (χ) in comparison to the original analysis (Jiang et al., 2007b). To conduct the convergence analysis of the SPSO with constriction coefficient (CSPSO), we consider the time step value $\triangle \tau$ to describe the dynamics of the PSO, and rewrite the velocity and position update formulas in (5.1) as follows:

$$V_i^{t+1} = \chi * \{\omega V_i^t + \varphi_1 \frac{(P_i^t - x_i^t)}{\triangle \tau} + \varphi_2 \frac{(g_i^t - x_i^t)}{\triangle \tau}\}, \tag{5.28}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \triangle \tau, \tag{5.29}$$

By replacing (5.28) in to (5.29), we obtain the following probabilistic CSPSO:

$$X_i^1 = X_i + \chi * \{\omega V_i + \varphi_1 \frac{(P_i - x_i)}{\triangle \tau} + \varphi_2 \frac{(g_i - x_i)}{\triangle \tau}\} \triangle \tau, \tag{5.30}$$

$$X_i^1 = X_i + \chi \omega V_i \triangle \tau + \chi \varphi \left(\frac{\chi \varphi_1 P_i + \chi \varphi_2 g_i}{\chi \varphi} - x_i \right) \tag{5.31}$$

By rearranging the terms in (5.31), we obtain

$$X_i^1 = (1 - \chi\varphi) X_i + \chi\omega \triangle \tau V_i + \chi\varphi_1 P_i + \chi\varphi_2 g_i. \tag{5.32}$$

Also, by rearranging the terms in (5.29), we obtain

$$V_i^1 = -\frac{\chi\varphi}{\triangle \tau} X_i + \chi\omega V_i + \chi\varphi_1 \frac{P_i}{\triangle \tau} + \chi\varphi_2 \frac{g_i}{\triangle \tau}. \tag{5.33}$$

We combine the above two (5.32) and (5.33) to have the following matrix form:

$$\begin{pmatrix} X_i^1 \\ V_i^1 \end{pmatrix} = \begin{pmatrix} 1 - \chi\varphi & \chi\omega \triangle \tau \\ -\frac{\chi\varphi}{\triangle\tau} & \chi\omega \end{pmatrix} \begin{pmatrix} X_i \\ V_i \end{pmatrix} + \begin{pmatrix} \chi\varphi_1 & \chi\varphi_2 \\ \frac{\chi\varphi_1}{\triangle\tau} & \frac{\chi\varphi_2}{\triangle\tau} \end{pmatrix} \begin{pmatrix} P_i \\ g_i \end{pmatrix} \tag{5.34}$$

which can be thought of as a discrete dynamic system representation for the PSO in which $(X \;\; V)^T$ is the state subject to an external input $(P_i \;\; g_i)^T$, and the two terms on the right side of (5.34) correspond to the dynamic and input matrices, respectively (D. Hu et al., 2021).

Supposing that no external excitation exists in the dynamic system, $[P_i, g_i]^T$ is constant, that is, other particles cannot find better positions. Then, a convergent behavior could be maintained. If it converges as $\tau \to \infty$, $(X_i^1 \;\; V_i^1)^T \to (X_i \;\; V_i)^T$. That is, the dynamic system becomes:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 - \chi\varphi & \chi\omega \triangle \tau \\ -\frac{\chi\varphi}{\triangle\tau} & \chi\omega \end{pmatrix} \begin{pmatrix} X_i \\ V_i \end{pmatrix} + \begin{pmatrix} \chi\varphi_1 & \chi\varphi_2 \\ \frac{\chi\varphi_1}{\triangle\tau} & \frac{\chi\varphi_2}{\triangle\tau} \end{pmatrix} \begin{pmatrix} P_i \\ g_i \end{pmatrix}$$

which holds only when $V_i = 0$ and $X_i = P_i = g_i$, where the convergent point is an equilibrium point if there is no external excitation, but better points are found by the optimization process with external excitation. For (5.34), D. Hu et al. (2021) has mentioned a sufficient strategies of improved convergence via theoretical analysis to get the relationship among $\chi,\ \omega$ and φ at the condition of convergence.

The derived probabilistic CSPSO can utilize any probabilistic form of prior information in the optimization process, and therefore the benefits from prior information can lead probabilistic CSPSO to more probable search region and help optimize more quickly with hierarchical use of parameters.

The probabilistic CSPSO results show that better convergence rate and quality of the optimized problem models than original PSO (Lee & Mukerji, 2014). By substituting (5.28) into (5.29), (5.29) is transformed into (5.35)

$$V_i^1 = \chi\omega V_i + \frac{\chi\varphi}{\triangle \tau}\left[P(i) - x_i\right] \tag{5.35}$$

where $P(i) = \frac{\varphi_1 P_i + \varphi_2 g_i}{\varphi}$.
Let $y_i = P(i) - x_i$, then (5.32) and (5.35) can be transformed in to, (5.36) and (5.37)

$$V_i^1 = \chi\omega V_i + \frac{\chi\varphi}{\triangle \tau} y_i \tag{5.36}$$

$$Y_i^1 = \chi\omega V_i + \left(1 + \frac{\chi\varphi}{\triangle \tau}\right) y_i \tag{5.37}$$

Combining (5.36) an iterative equation in the form of vector is obtained as (5.38)

$$\begin{pmatrix} V_i^1 \\ Y_i^1 \end{pmatrix} = \begin{pmatrix} \chi\omega & \frac{\chi\varphi}{\triangle\tau} \\ \chi\omega & 1 + \frac{\chi\varphi}{\triangle\tau} \end{pmatrix} \begin{pmatrix} V_i \\ y_i \end{pmatrix} \tag{5.38}$$

which can be viewed as a general forecasting model of Markov chain as follows;
$\eta^t = PK_i^t$, where, K_i^t is a vector as presented in tbelow

$(\eta^t)^T = \begin{bmatrix} V_i^1 & Y_i^1 \end{bmatrix}, \quad P^T = \begin{bmatrix} \chi\omega & \chi\omega \\ \frac{\chi\varphi}{\Delta\tau} & 1 + \frac{\chi\varphi}{\Delta\tau} \end{bmatrix}, \quad (K_i^t)^T = \begin{bmatrix} V_i & y_i \end{bmatrix}$
(5.38), the model with no external excitation is useful in studying the evolution of certain systems over repeated trials as a probabilistic (stochastic) model (D. Hu et al., 2021).

Using the Markov chain method, the position $(\eta^t)^T(t+1)$ of the d th element of the i th particle at the $(t+1)$ th iteration in CSPSO can be computed using the following formula:

$$(\eta^t)^T(t+1) = [(k_i^t)^T(t)]XP^T \tag{5.39}$$

superscript T denotes the transposition.

The study also addresses the CSPSO algorithm that can be analyzed using Markov chain theory, and it also presents the convergence properties of the algorithm in the context of almost sure convergence, based on Schmitt & Wanka (2013); Yuan & Yin (2014)

1. As the algorithm progresses and more iterations are performed, the algorithm will converge to an optimal solution with a probability of 1 and
2. Given sufficient time and iterations, the algorithm will find the globally optimal solution or arbitrarily approach it.

5.3.2 Stability Analysis of CSPSO

We further get insight into the dynamic system in (5.39). First, we solve the characteristic equation of the dynamic system from $det(P^T - \lambda I) = 0$ as follows:

$$\lambda^2 - (1 + \chi\omega + \frac{\chi\varphi}{\Delta\ \tau})\lambda + \chi\omega = 0.$$

The eigenvalues are obtained as follows: $\lambda_{1,2} = \frac{(1+\chi\omega+\frac{\chi\varphi}{\Delta\tau}\pm\gamma)}{2}$, $\gamma = \sqrt{(1 + \chi\omega + \frac{\chi\varphi}{\Delta\tau})^2 - 4\chi\omega}$ with $\lambda_1 \geq \lambda_2$. The explicit form of the recurrence relation (5.29) is then given by

$$Y_i^1(t) = r_1 + r_2\lambda_1^t + r_3\lambda_2^t$$

where r_1, r_2, and r_3 are constants determined by the initial conditions of the system. From updated velocity

$$V_i^1(t+1) = \frac{Y_i^1(t+1) - Y_i(t)}{\Delta\ \tau} \tag{5.40}$$

results in $V_i^1(t+1) = \frac{r_2(\lambda_1^{t+1}-\lambda_1^t)+r_3(\lambda_2^{t+1}-\lambda_2^t)}{\Delta\tau} = (r_2\frac{\lambda_1-1}{\Delta\tau})\lambda_1^t + (r_3\frac{\lambda_2-1}{\Delta\tau})\lambda_2^t$
$k_1 = \frac{r_1(\lambda_1-1)}{\Delta\tau}$ and $k_2 = \frac{r_2(\lambda_2-1)}{\Delta\tau}$

$$\lim_{t\to\infty} V_i^1(t+1) = \lim_{t\to\infty} k_1\lambda_1^t + \lim_{t\to\infty} k_2\lambda_2^t$$

$$\lim_{t\to\infty} X_i^1(t+1) = \begin{cases} \lim_{t\to\infty} x_i^1(t) \;\; if \; max(||\lambda_1||, ||\lambda_2||) < 1, \\ (k_1 \; or \; k_2 \; or \; k_1 + k_2) + \lim_{t\to\infty} x_i^1(t) \; if \; max(||\lambda_1||, \; ||\lambda_2||) = 1 \end{cases} \tag{5.41}$$

(5.41) implies that if the CSPSO algorithm is convergent, then velocity of the particles will decrease to zero or stay unchanged until the end of the iteration.

5.3.3 Constriction Factor and Its Impact

When the PSO algorithm is run without controlling the velocity, the system explodes after a few iterations. To control the convergence properties of a particle swarm system, an important model having constriction factor and ω together is shown below:

$$V_i^{t+1} = \chi * \{\omega V_i^t + \varphi_1(P_i^t - x_i^t) + \varphi_2(g_i^t - x_i^t)\}, \tag{5.42}$$

$\kappa = \chi \mid 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \mid, \;\; \varphi_1 + \varphi_2 = \varphi \geq 4, \;\; 0 \leq \kappa \leq 1.$
Under these assumption conditions, the particle's trajectory in the CSPSO system is stable.

$$\omega^{t+1} = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{t_{max}}\right) t, \; \omega_{max} > \omega_{min} \tag{5.43}$$

where, ω_{max} and ω_{min} are the predefined initial and final values of the inertia weight respectively, t_{max} is the maximum iteration number, and t is the current iteration number for a linearly decreasing inertia weight scheme.

5.3.4 Global Convergence Analysis of QBCSPSO

A sequence generated by the iterative PSO algorithm converges to a solution point. Several PSO variants were proposed to enhance convergence performance of PSO by Sun et al. (2004); A. Kumar et al. (2016), which combines quantum results with CSPSO, denoted as QBCSPSO. In this subsection, the global convergence of QBCSPSO is investigated.

From the Monte Carlo method, the current velocity for the position x_i^{t+1} of the d th element of the i th particle at the $(t+1)$ th iteration in QBCSPSO can be obtained using the following formula:

$$V_i^{t+1} = Y_i^t \pm L_i^t \, ln(u_i^{-(t+1)}), \;\; u_i^{t+1} \sim U(0,1) \tag{5.44}$$

where $U_i^t \in (0, \; 1)$. Referring to Sun et al. (2004, 2012), with δ the (wave) function $\delta(Y_i^{t+1}) = \frac{1}{\sqrt{L_i^t}} \, exp(-Y_i^{t+1}/L_i^t)$ for $Y_i^{t+1} =\mid x_i^{t+1} - P_i^t \mid$, and the characteristic length L_i^t is obtained by

$$L_i^t = \gamma |x_i^t - C^t| \tag{5.45}$$

the term C^t used in (5.45) is $C^t = \frac{1}{K}\sum_{i=1}^{K} P_i^t$ (Sun et al., 2004).

The contraction-expansion coefficient γ, which can be adjusted to balance the trade-off between the global and local exploration ability of the particles during the optimization process for two main purposes

- a larger γ value enables particles to have a stronger exploration ability but a less exploitation ability.
- a smaller γ allows particles to have a more precise exploitation ability (Chen et al., 2009).

5.4 Numerical Simulations

To demonstrate the applicability and efficiency of the CSPSO algorithm, two well-known test functions in a global optimization were widely used in evaluating the performance of evolutionary methods, and have the global minimum at the origin or very close to the origin. We compare the performance of PSO, SPSO, CPSO and CSPSO.
Example 5.1. Unimodal function

$$\begin{gathered} min\ f(x_i) = \sum_{i=1}^{K} x_i^2 \\ subject\ to - 10 \leq x_i \leq 10. \end{gathered} \tag{5.46}$$

Example 5.2. Multi modal function

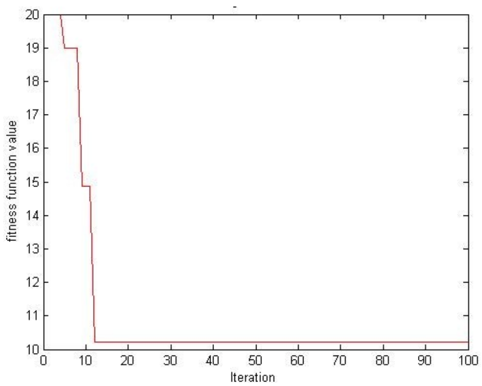
$$\begin{gathered} min\ f(x) = \sum_{i=1}^{K} -x_i sin(\sqrt{\mid x_i \mid}) \\ subject\ to - 10 \leq x_i \leq 10. \end{gathered} \tag{5.47}$$

In the experiments, inertia weight decreases from 0.9 to 0.4, the generation stops when $E_i = \mid Fg^t(x_i) - Fp^t(x_i) \mid \leq tolerance$ is satisfied. Here, Fp is the function value of the best personal in current iteration and Fg denotes the global optimum and $c_1 = c_2 = 1.49$ and $c_1 = c_2 = 2$ are used in PSO and CSPSO, respectively.
For all algorithms, results are averaged over 100 independent runs and iterations while the population size is 50. Following the recommendations of the previous literature's, the best function value settings of some compared algorithms are summarized in Table 1.
The mean velocity v^{t+1} of (5.46 & 5.47) are shown using Table 5.1 and 5.2 and also graphically depicted as Fig. 5.1 & 5.2, respectively. For the algorithms in Table 5. 1, Fig.5.1 shows the convergence of PSO without controlling factor inertia weight exploded. One of the main limitations of PSO is that particles prematurely converge to a local solution.

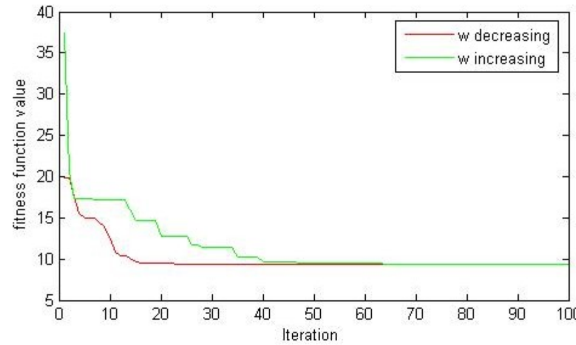
Table 5.1: Comparison of algorithms on optimization test functions

№	**best fun**	**algorithm**	**best run**	**best variables**	$\omega \in range$
1.	10.2213	Basic PSO	35	$[\,0.3809,\ 1.4531,\ 3.1164\,]$	
2.	9.3944	SPSO	4	$[\,0.4381,\ 1.4566,\ 3.1052\,]$	$[0.9,\ 0.4]$
3.	9.3945	SPSO	27	$[\,0.4384,\ 1.4564,\ 3.1051\,]$	$[0.4,\ 0.9]$
4.	9.3941	CSPSO	29	$[\,0.4379,\ 1.4568,\ 3.1053\,]$	$[0.9,\ 0.4]$
5.	9.3941	CSPSO	28	$[\,0.4381,\ 1.4567,\ 3.1052\,]$	$[0.4,\ 0.9]$

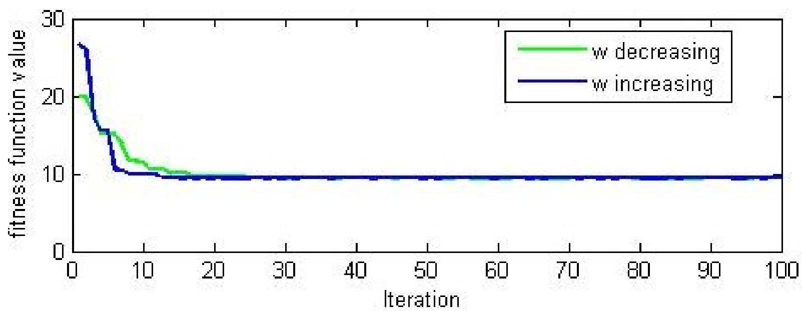
The evaluation results of the compared algorithms are shown in Fig.5.1 (c) for decreasing and increasing the inertia weight, respectively. Fig.5.2 shows the evolution of the inertial weight of the compared algorithms over the running time. The main disadvantage is that once the inertia weight is decreased, the swarm loses its ability to search new areas (Ezugwu et al., 2022). Fig.5.1(c) shows the evolution of the convergence characteristic for CSPSO based on the inertial weight during the run. CSPSO can avoid premature convergence by performing efficient exploration that can help find better solutions as the number of iterations increases, and it can avoid premature convergence by balancing exploration and exploitation. The algorithm CSPSO has shown a fast convergence speed on unimodal functions.



(a) Basic PSO with no inertia weight for (5.46)



(b) Evaluation of SPSO for (5.46)



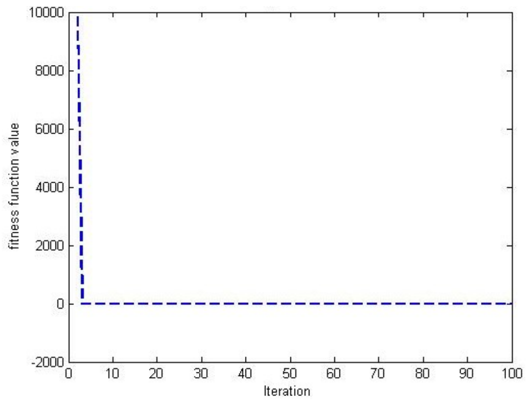
(c) Evaluation of CSPSO for (5.46)

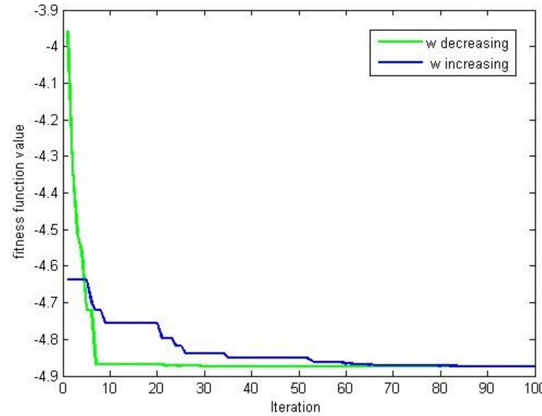
Figure 5.1: Evaluation of the algorithms on Example 5.1.

In order to confirm the performance on multi-modal functions, we carry out a similar simulation by using (5.47). The same set of parameters is assigned for all algorithms in Table 5.1 as in (5.46). Here in this function, the number of local minima increases exponentially with the problem dimension. Its global optimum value is approximately -5.74 as we see from Table 5.2. Fig.5.2 $(a) - (c)$ are the simulations obtained from the results of Table 5.2, and all the figures meet the objective of the CSPSO algorithm for the optimization problem given in (5.47) and its evaluation as shown in Fig.5.2(c).

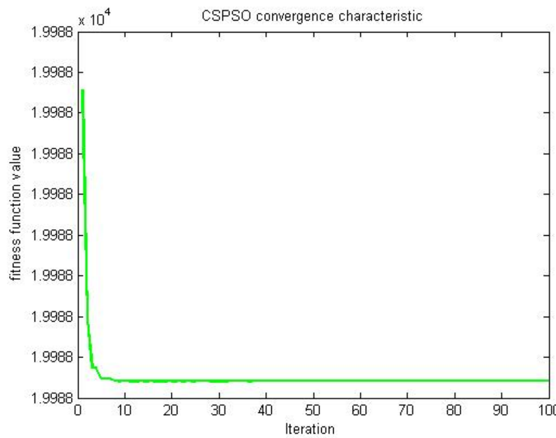
Table 5.2: Comparison of algorithms on optimization test functions of multi modal

No̲	**algorithm**	**best fun**	**best run**	**best variables**	$\omega \in range$
1.	CPSO	-6.12	01	$[\,1.79,\ 0,\ 3.20]$	$\omega = 1$
2.	SPSO	-6.73	32	$[\,1.79,\ 0,\ 3.21]$	$[0.4,\ \ 0.9]$
3.	SPSO	-8.133	89	$[\,1.79,\ 0,\ 3.21]$	$[0.9,\ \ 0.4]$
4.	CSPSO	-11.83	69	$[5.23, 5.23, 5.23\,]$	$[0.9,\ \ 0.4]$
5.	CSPSO	-5.74	39	$[\,1.79,\ 0,\ 3.21]$	$[0.4,\ \ 0.9]$

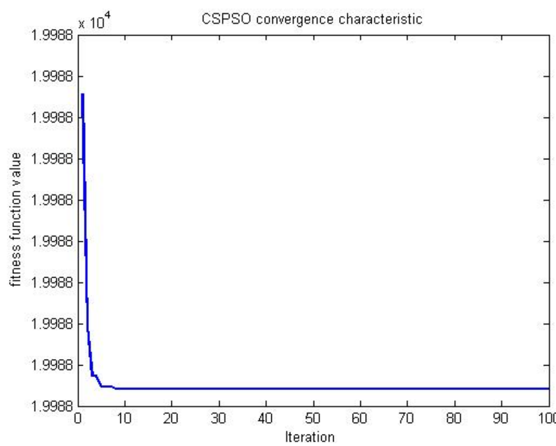




(a) CPSO when $\omega = 1$ for (5.47) **(b)** Evaluation of SPSO for (5.47)



(c) Evaluation of CSPSO for (5.47)



(d) Evaluation of CSPSO for (5.47)

Figure 5.2: Evaluation of the algorithms on Example 5.2.

CHAPTER 6

A CONVERGENT PARTICLE SWARM OPTIMIZATION METHOD WITH REPULSIVE FUNCTIONAL CONSTRAINTS FOR SOLVING UNIMODAL AND MULTIMODAL PROBLEMS

6.1 Introduction

Functional constraints refer to restrictions or conditions that must be satisfied by potential solutions. These constraints are typically expressed as mathematical equations or inequalities that involve the decision variables of the optimization problem to ensure that the solutions obtained are both optimal and feasible. In this chapter, we presented a reformulation of the classical particle swarm optimization (PSO) algorithm from a mathematical point of view by describing the influence of repulsive functional constraints on the convergence of the PSO algorithm.

6.2 Mathematical Algorithm Description and Formulation

In nonlinear analysis, particle swarm optimization plays an important role, particularly in optimization methods to solve complex problems in various disciplines (Elsheikh & Abd Elaziz, 2019). Moreover, its applications range from a zero-order, non- calculus-based method (no gradients are needed) to solve discontinuous, multimodal, non-convex problems by including some probabilistic features in the motion of particles problem (Elsheikh & Abd Elaziz, 2019; Gad, 2022). Let X_t^i and V_t^i be in $2D-$ dimensional particles randomly generated and velocity vectors for each particle, respectively. The classical constraints problem consists of finding $X \in S$ (swarm particles) such that $S \subset R^D$ where R^D denotes the set of positions (x_o, y_o) for i th particle at t th time unit and is defined as

$$f_t^i = \sqrt{(x_t^i - x_0^i)^2} + \sqrt{(y_t^i - y_0^i)^2}. \tag{6.1}$$

Problem (6.1) attracts much interest and many works have been published for various types of current fitness f_t^i (Jiang et al., 2007b). In particular, Goel & Panchal (2014) considered $D-$ dimensional multimodal nonlinear constraint functions. Recall that a constraint is called the velocity that determines the PSO algorithm convergence if the set of acceleration coefficients values $C = \{c_1,\ c_2\}$ is nonempty, m is a uniformly distributed random number, in [0, 1] and x^i is the position of the particles at iteration t to $t+1$ as described by Kennedy & Eberhart (1995) such that

$$V(x_{t+1}^i) = \begin{cases} V(x_t^i) - c_1 m \;\; if\ x_t^i > x^i \\ V(x_t^i) + c_1 m \;\; if\ x_t^i < x^i \end{cases} \tag{6.2}$$

or equivalently, the particle with the lowest fitness value, f_t^g (global best)

$$V(x_{t+1}^i) = \begin{cases} V(x_t^i) - c_2 m\ if\ x_t^i > x^g \\ V(x_t^i) + c_2 m\ if\ x_t^i < x^g \end{cases} \tag{6.3}$$

This class of fitness is quite general since it includes many special cases, such as position associated with $P^i = [x^i, y^i]$, and the position of the best particle in the current swarm, $P^i = [x^g, y^g]$, generalized hybrid and probabilistic PSO, as well as other nonlinear PSO algorithm arising in meta-heuristic optimization.

From the algorithms point of view, there exist numerous methods for solving (6.1) by Hassan (2004) and the references therein. In particular, Tilahun (2022) studied the convergence analysis of PSO iterative algorithm

$$\begin{aligned} V_{t+1}^i &= \omega V_t^i + c_1 r_t^1 (P_t^i - X_t^i) + c_2 r_t^2 (g_t - X_t^i), \\ X_{t+1}^i &= X_t^i + V_{t+1}^i, \end{aligned} \tag{6.4}$$

and proved its weak convergence under some suitable conditions on the involved parameters. Yang (2011) introduced another iterative procedure that incorporates no gradients and iterative algorithms. Choose $x_1 \in X$ and $x_1 = x_o + v_1$ and generate the sequence x_{t+1} where X is the position and V is the velocity of the individual particle. z is a random velocity vector and the subscripts t and $t+1$ in standard PSO with repulsive functional constraint (SPSO-RFC) (6.5) represent the number of recent and next iterations, respectively.

$$\begin{aligned} V_{t+1}^i &= \omega V_t^i + c_1 r_t^1 (P_t^i - X_t^i) + \omega c_2 r_t^2 (g_t - X_t^i) + \omega c_3 r^3 z, \\ X_{t+1}^i &= X_t^i + V_{t+1}^i, \end{aligned} \tag{6.5}$$

Under the assumption that cognitive $(P_t^i - X_t^i)$ and social $(g_t - X_t^i)$ components are bounded, and with some other suitable conditions on the involved parameters, strong convergence of (6.5) is established. Following this result, other authors generalized and extended it as can be seen in convergence of PSO for example.

Here, we present the PSO with current velocity-to-zero problem (PSO-VZP). Let X and Y be search spaces, and ZP1 and ZP2 two zero problems defined in X and Y, respectively. In addition, let $\Lambda \subset R$ be an index set and $f_t = X \rightarrow Y$ be a linear transformation for each $t \in \Lambda$. Given these data, the current velocity to zero problem is formulated as follows. find $x^i \in X$ that solves ZP1

$$V_{t+1}^i = c_1 r_t^1 (P_t^i - X_t^i) + c_2 r_t^2 (g_t - X_t^i) + \omega c_3 r^3 z, \tag{6.6}$$

such that "particle cost = cost function (particle position)"or $y^i = f_t(x^i)$ $\forall t$ iteration and y^i solves ZP2 for penalty multipliers α_i, running from $i = 1$ to number of convergence (N_{con})

$$F(X) = \Pi(X) + \sum_{i=1}^{N_{con}} \alpha_i (max[0, g_i])^2 \tag{6.7}$$

where the two right side functions are objective and penalty function, respectively (Hassan, 2004). When $f = f_t$ $\forall t$, the basic PSO algorithm reduced to PSO with velocity to zero for particle swarm position (g_{ht}) and $\triangle\tau$ is the time step value obtained from difference of the current (τ) and the initial (τ_o) to describe the dynamics of the PSO. In views of (6.6) and (6.7), we wish to introduce the generalized velocity to zero repulsive functional constraint problem (GVZRFCP),

$$V_{t+1}^i = c_1 r^1 \frac{(P_t^i - X_t^i)}{\triangle\tau} + c_2 r^2 \frac{(g_{ht} - X_t^i)}{\triangle\tau} + \omega c_3 r^3 z, \tag{6.8}$$

The accelerating factors $\varphi_t^1 = c_1 r^1$, $\varphi_t^2 = c_2 r^2$, & $\varphi_t^3 = c_3 r^3$, and g_h is best position of a randomly chosen other particle from within the swarm in (6.8).

$$V_{t+1}^i = \varphi_t^1 \frac{(P_t^i - X_t^i)}{\triangle\tau} + \varphi_t^2 \frac{(g_{ht} - X_t^i)}{\triangle\tau} + \omega \varphi_t^3 z, \tag{6.9}$$

These exhibit different features like self-organization capability, decentralization, and easy implementation of the swarm on the basis of two different components: exploitation and exploration. The exploration property asserts that the algorithm is escaping from the local optima and is searching the entire search space, while the exploitation property ensures the efficiency of the algorithm in carefully searching and approaching (converging) the optimal point (Jiang et al., 2007b; Kim & Xu, 2008).

Tuning the values of χ, c_1, c_2, and c_3, can prevent the swarm from being explosive and can lead to a better balance between exploration and exploitation within the search space (Kim & Xu, 2008; Marino et al., 2016; Tilahun, 2022). They showed that velocity does not grow to infinity if the parameters (χ, c_1, c_2 & c_3) satisfy the following equation, and we set the needed swarm size for the PSO algorithms based on specific problems (Mehmood et al., 2018). $\chi = \frac{2k}{|2 - \varphi_t - \sqrt{\varphi_t^2 - 4\varphi_t}|}$, where, $\varphi_t = \varphi_t^1 + \varphi_t^2 > 4$.

The two strategies are merged using weighting factors by $X^i_{t+1} = X^i_t + V^i_{t+1}$ as shown in (6.10):

$$X^i_{t+1} - (1 - \frac{\varphi_t}{\Delta\tau})X^i_t - \omega\varphi^3_t z = \frac{\varphi^1_t P^i_t + \varphi^2_t g_t}{\Delta\tau} = O^i_t \quad (6.10)$$

Motivated and inspired by the above equations, to the framework of GVZRFCP (6.9) by combining the probabilistic and stability technique of D. Hu et al. (2021) to accelerate the convergence rate. Moreover, a new step size rule is used, enabling us to avoid the precomputation /evaluation of the norms $||f_t||(t \in \{1, 2, \ldots, N\})$, which might be a hard task in practice.

6.3 Preliminaries

We introduce some notation and recall some useful definitions and results that are needed for the rest of the chapter (Kim & Xu, 2008).

Let Y be a stochastic process in a general state space S.

Definition 1. For every $i^l \in S$ $(l-1 \leq t)$, the sequence $\{Y^t\}_{t\geq 0}$ is a discrete stochastic process.

Definition 2. Let $\{Y^t,\ t \geq 0\}$ be a discrete stochastic process, $\forall t \geq 0$ and $i^l \in S$ $(l \leq t+1)$, if the probability $P\{Y^0 = i^0,\ Y^1 = i^1, \ldots, Y^t = i^t\} > 0$ and $P\{Y^{t+1} = i^{t+1} \mid Y^0 = i^0,\ Y^1 = i^1, \ldots, Y^t = i^t\} = P\{Y^{t+1} = i^{t+1} \mid Y^t = i^t\}$.

Definition 3. The state of particle $\kappa^i_t = (x^i_{t-1},\ x^i_t,\ p^i_t,\ g_t)$ at the t-th iteration for particle i in (6.8), and κ^i_t, the update probability of the state of the particle.

Definition 4. The state of swarm in (6.8), at iteration t, denoted as η_t, is defined as $\eta_t = (\kappa^1_t,\ \kappa^2_t, \ldots, \kappa^K_t)$.

The state of swarm space is a set of all possible states of swarm, denoted as ϖ.

Definition 5. Let Γ^n_1 denote the σ-field generated by particles state $\kappa^1_t,\ \kappa^2_t, \ldots, \kappa^n_t$, $(K \geq n)$ and define

$$\phi((\Gamma^n_1,\ \kappa^{n+1}_t) = sup\{\mid P(B\backslash A) - P(B) \mid : A \in \Gamma^n_1,\ B \in \sigma(\kappa^{n+1}_t)\}, \quad (6.11)$$

$$\phi = \underset{1\leq n\leq K-1}{sup} \phi(\Gamma^n_1,\ \kappa^{n+1}_t) \quad (6.12)$$

Due to the weak interdependent relationship among the particles, ϕ is approximately small.

Properties 2.1

1. If the accelerating factors in (6.9) satisfy $\varphi^1_t,\ \varphi^2_t,\ \&\ \varphi^3_t \in U(0, c)$, then the probability for particle i changes from the position x^i_t to the spherical region centered at x^i_{t+1} with radius $\varrho_t > 0$. The event $A^i = \{\kappa^i_{t+1} \mid \kappa^i_t\}$, defining the state of particle i at the t-th iteration is updated to the state at the $(t+1)$-th iteration with constant c, for each $i \in \{1, 2, \ldots, K\}$ can be computed as

$$P(A^i) = \frac{\varrho^3_t \Delta^3\tau}{\varphi^3_t\omega \mid\mid x^i_t - X^i_{t-1} \mid\mid\ \ c \mid\mid p^i_t - X^i_t \mid\mid c \mid\mid g_t - X^i_t \mid\mid}, \quad (6.13)$$

c is a constant within $U(0, c)$ and $\delta \to 0$, where

$$\varrho = c * \begin{cases} \| \, p_t^i - X_t^i \, \|, \ for \ f(x_t^i) - \delta \leq f(P_t^i) \leq f(x_t^i) + \delta \\ \| \, g_t - X_t^i \, \|, \ for \ f(x_t^i) - \delta \leq f(g_t) \leq f(x_t^i) + \delta \end{cases} \tag{6.14}$$

proof. The 1-step transition probability of the i th state of particle, P_{t+1}^i and g_{t+1} are determined by x_{t+1}^i for transferring κ_t^i to κ_{t+t}^i based on SPM-Single Particle Model in (Piotrowski et al., 2020).

2. The set of collection of swarm state $\{\eta_t\}_{t \geq 1}$ is a Markov chain.
   Proof. The proof follows by referring to equation of position.

3. The transition probability from η_t to η_{t+1} satisfies

$$| \, P(\eta_{t+1} \mid \eta_t) - \prod_{i=1}^{K} P(\kappa_{t+1}^i \mid \kappa_t^i) \, | \leq \mu \tag{6.15}$$

where μ can be made small enough, therefore, $\mu = (2^{K-1} - 1)\phi$.
Proof. Based on the Definition 5.

6.4 Main Results

Since the goal of this part is to prove rigorous results on the convergence of the algorithm PSO-RFC process and solving GVZRFCP (6.9) under some important conditions, assumptions, and sequence of steps. In order to address all issues identified in the previous section, a new form of the velocity update rule is proposed; it uses a balance point, probabilistic and stability convergence, trajectories, and theoretical analysis (similar to the CSPSO approach, by Kim & Xu (2008); Tilahun (2022), but with some important modifications).

6.4.1 Convergence Proof Towards a Local Optimum

A swarm S of K particles moves through the D-dimensional search space R^D. Let an objective function f: $R^D \to R$ in a D-dimensional domain be given that (without loss of generality) has to be minimized. For proving the convergence of a particle swarm, we first consider the following conditions based on swarm convergence for a point O

1. Initialization of particles positions in a possible area of global attractor O, satisfying $f(x) > f(O)$, for x in R^D.
2. The movement of the particles tends to zero ($\lim\limits_{t \to \infty} V_t = 0$).
3. Every particle moves towards O ($\lim\limits_{t \to \infty} X_t^i = O$).

It follows from the swarm convergence conditions $(1-3)$ above that $\lim_{t\to\infty} g_t = O$ almost surely and $\lim_{t\to\infty} P_t^i = O$ for every $i \in \{1, \ldots, K\}$.

Assumption 3.1

The convergence analysis in the literature of Gad (2022), usually makes the following assumptions about f,

(A1): At least the global attractor is constant forever,

(A2): The results of convergence proof hold under weaker assumptions of the convergence of the attractors,

(A3): The swarm is able to find a local minimum.

A positively running particle swarm occurs when the particles are consistently improving their positions and moving towards better solutions over time. The behavior of a particle swarm is influenced by numerous factors that interact in complex ways. Analyzing and addressing the specific causes behind the behavior is crucial to improving the performance of the particle swarm optimization process. Some of the factors that lead to a positive running particle swarm are well-tuned parameters, suitable fitness evaluation, favorable problem landscape, and sufficient diversity. However, a negatively running particle swarm could imply a behavior in which particles do not make progress or move away from better solutions over time (Goel & Panchal, 2014).

Conditions 3.1

For the algorithm's setup, we define the following conditions satisfying positively $(C1)-(C3)$ and negatively $(C4)-(C6)$ running particle swarm.

Let $d_0 \leq D$ be an arbitrary dimension. The swarm S is called positively running in the direction d_0 in iteration t, for every $i \in \{1, \ldots, K\}$:

(C1): $g_{t+1}^{i+1} = argmin\{P_{t+1}^i,\ g_t^i\}^f,\ 1 \leq i \leq K-1$

(C2): $P_{t+1}^i = argmin\{P_t^i,\ X_{t+1}^i\}^f,\ 1 \leq i \leq K-1$

(C3): $V_t^i \geq 0$, for every i.

In case of the above conditions, the larger the d_0' th entry of a position or attractor, the better its function value. If the state of running is maintained long enough, all particles will eventually overcome their own local attractors. From that point onward, as long as the swarm continues to running, the local attractors of the particles are identical to their current positions.

(C4): $g_{t+1}^{i+1} = argmax\{P_{t+1}^i,\ g_t^i\}^f,\ 1 \leq i \leq K-1$

(C5): $sign(P_t^{i,\, d_o} - X_t^{i,\, d_o}) \leq 0$.

(C6): $V_t^i \leq 0$ for every i.

Basically, one may think of running as the behavior a swarm depicts when it moves through some region that is monotone in one dimension d_0, while changes in any other dimension are insignificant.

Property 3.1. For some parameters in (8) and other $K,\ \chi,\ c_1,\ c_2,\ c_3$ and the swarm S running

i. positively if $(V_t^{i,d_o} + X_t^{i,d_O}) \to \infty$

ii. negatively if $(-V_t^{i,d_0} - X_t^{i,d_0}) \to \infty$

in d_o, for every i almost surely (Kennedy & Mendes, 2002). In particular, the swarm leaves every bounded set $B \subset R^D$ almost surely. To run a particle swarm optimization (GVZRFCP) algorithm, we need to follow a series of steps.

Table 6.1: PSO-RFC algorithm flow

STEP 1. Problem Formulation: Clearly define the optimization problem, including the objective function to be minimized or maximized and any constraints that must be satisfied.
STEP 2. Initialize a potential solution to the problem P_t^i, and x_t^i. Randomly assign X_t^i and Z.
STEP 3. Evaluate $f(x_{t+1}^i)$.
STEP 4. Assign P_t^i, and x_t^i. Compare $f(p_t^i)$ with $f(x_t^i)$. If $f(x_t^i) \geq f(p_t^i)$ update p_t^i to x_t^i.
STEP 5. Determine g_t and compare $f(p_t^i)$ with $f(g_t)$. If $f(g_t) \leq f(p_t^i),\ p_t^i$, update g_t accordingly.
STEP 6. Update (9) of V_{t+1}^i and compute X_{t+1}^i as follows $X_{t+1}^i = X_t^i + V_{t+1}^i$.
STEP 7. Determine if $\lvert f(g_t) - f(P_{t+1}^i)\rvert \leq$ error tolerance (level of convergence).
STEP 8. Set $t = t + 1$ and go to STEP 3.
STEP 9. Once the algorithm converges, extract the solution from the global best position. This solution should represent an optimized solution to the problem.

We apply the above conditions and assumptions, to determine selected objective function satisfies that

1. Velocity stays positive implies swarm will stay positively running forever and

2. Depending on the parameters $\omega\varphi_t^3,\ \varphi_t^2$ and φ_t^1, running swarm loses potential implies convergences

Example 1. To show the convergence of the objective function, consider a single dimensional particle swarm, $f(x) = -x$.

Proof: Assume that $\forall V_t^i > 0$ of X_t^i. Then by (C1) in direction $d_o = 1$ forever.
It is obvious that $X_t^{i,1}$ with the greatest $x-$value tends to the smallest value of $f(x)$ and therefore becomes the global attractor. It remains to prove that $\forall V_t^i > 0$. From (6.9) given the old velocity $V_t^{i,1}$, the new velocity $V_{t+1}^{i,1}$ is a positive linear combination of the three components $\varphi_t^1 \frac{(P_t^{i,1} - X_t^{i,1})}{\triangle\tau}$, $\varphi_t^2 \frac{(g_{ht}^1 - X_t^{i,1})}{\triangle\tau}$ and $\omega\varphi_t^3 z$. The value of $V_t^{i,1} > 0$ by assumption, $\varphi_t^1 \frac{(P_t^{i,1} - X_t^{i,1})}{\triangle\tau} \nleq 0$ and $\varphi_t^2 \frac{(g_{ht}^1 - X_t^{i,1})}{\triangle\tau} \nleq 0$ since $X_t^{i,1} \leq P_t^{i,1} \leq g_t^1$.

In that situation, a good behavior would be increasing (or at least nondecreasing) potential. The negative results presented by D. Hu et al. (2021) indicate that, under conditions (C4) - (C6), a running swarm loses potential and leads to convergence on the parameters mentioned above.

6.4.2 Moving Particles Convergence Speed Controller Rule

The movement of particles in PSO can be likened to the motion of a system of particles in physics. Hybridizing PSO with other algorithmic techniques can be beneficial in improving the behavior and convergence performance of SPSO within the overall algorithm to address the specific challenges and requirements of the VZRFC problem (Xiao & Yin, 2014).

The three components to describe the position of one moving particle at the step $t+1$ are : the current velocity z, the personal best position P_t^i and the global best position g_t among the informants of the particles.

Let $\mathbf{R}$ denote the position vector of the center of mass (the balance point) of the particles X_t^i, $\varphi_t^1 P_t^i$ and $\varphi_t^2 g_t$, where φ_t^1, φ_t^2 and φ_t^3 denote three positive real coefficients. The coordinates of the point $\mathbf{R}$ can be obtained as: $\mathbf{R} = \left[3X_t^i + \varphi_t^1(P_t^i - X_t^i) + \varphi_t^2(g_t - X_t^i)\right]/3$. Then, at iteration t a point $X_t^{i'}$ is randomly drawn in the Hypersphere, which is centered on the point $\mathbf{R}$ with a radius equal to $||\mathbf{R} - X_t^i||$ is $H(\mathbf{R}, ||\mathbf{R} - X_t^i||)$. We analyzed the performance of PSO-RFC under the effect of moving rule of particles and/or center of mass information, as described above. This results in the position and velocity update equations:

$$X_{t+1}^i = \omega\varphi_t^3 z + X_t^{i'}$$

and

$$V_{t+1}^i = \omega\varphi_t^3 z + X_t^{i'} - X_t^i.$$

Since $\mathbf{R}$ depends on φ_t^1 and φ_t^2 appropriate selection of the four parameters ω, φ_t^1, φ_t^2, and φ_t^3 can significantly improve the performance of the proposed algorithm.
The position vector of the center of mass $\mathbf{R}$, which can be adjusted to balance the trade-off between global and local exploration ability of the particle trajectory during the optimization process for two main purposes (Schmitt, 2015).

- investigate the deterministic trajectory $\mathbf{R} = X_t^{i'}$, enables particles to have a stronger exploration ability but a less exploitation ability.
- investigating the undeterministic trajectory $\mathbf{R} \neq X_t^{i'}$, allows particles to have a more precise exploitation ability.

However, the undeterministic nature of particle trajectories in PSO also implies that the algorithm's performance can vary across different runs. It is possible that a particular run of PSO-RFC may converge to a high-quality solution, while another run with the same parameters may converge to a suboptimal solution. This stochastic behavior motivates the multiple running of the PSO-RFC and considering statistical measures to assess its performance.

6.4.3 Probabilistic Convergence Analysis of PSO-RFC

In this subsection, we investigate convergence analysis for the version of the basic PSO (GVZR-FCP), by analogy to the method to analyze the convergence of the PSO specified by Schmitt & Wanka (2013), and we also incorporate the idea of swarm convergence conditions and assumptions. Our analysis has the advantage of providing proof of convergence of the PSO with repulsive functional constraints, in comparison to the original analysis (Schmitt, 2015). To carry on the convergence analysis of the PSO-VZRFC, we describe the dynamics of the PSO-RFC, and rewrite the velocity and position update formulas in (6.9) as follows:

$$V_{t+1}^i = \varphi_t^1 \frac{(P_t^i - x_t^i)}{\triangle\ \tau} + \varphi_t^2 \frac{(g_{ht} - x_t^i)}{\triangle\ \tau} + \omega\varphi_t^3 z, \tag{6.16}$$

$$X_{t+1}^i = X_t^i + V_{t+1}^i \tag{6.17}$$

By replacing (6.16) in to (6.17), we obtain the next three (6.18), (6.19) and (6.20):

$$X_1^i = X^i + \varphi_t^1 \frac{(P_t^i - x_t^i)}{\triangle\ \tau} + \varphi_t^2 \frac{(g_{ht} - x_t^i)}{\triangle\ \tau} + \omega\varphi_t^3 z \tag{6.18}$$

$$X_1^i = (1 - \varphi_*)x^i + \varphi_*^1 p^i + \varphi_*^2 g_h + \omega\varphi^3 z, \tag{6.19}$$

where, $\varphi_* = \frac{\varphi}{\triangle \tau}$,

$$V_1^i = -\varphi_* x^i + \varphi_*^1 p^i + \varphi_*^2 g_h + \omega\varphi^3 z. \tag{6.20}$$

We combine (6.19) and (6.20) to have the matrix form:

$$\begin{pmatrix} V_1^i \\ X_1^i \end{pmatrix} = \begin{pmatrix} \omega\varphi^3 & -\varphi_* \\ \omega\varphi^3 & 1 - \varphi_* \end{pmatrix} \begin{pmatrix} z \\ x^i \end{pmatrix} + \begin{pmatrix} \varphi_*^1 & \varphi_*^2 \\ \varphi_*^1 & \varphi_*^2 \end{pmatrix} \begin{pmatrix} P^i \\ g_h \end{pmatrix} \tag{6.21}$$

which can be thought of as a discrete dynamic system representation for the PSO-RFC in which $(X \ \ V)^T$ is the state subject to an external input $(P^i \ \ g_h)^T$, and the two terms on the right side of the equation correspond to the dynamic and input matrices, respectively (Kim & Xu, 2008; D. Hu et al., 2021). Suppose that there is no external excitation in the dynamic system, $[P^i, g_h]^T$ is constant. Then, convergent behavior could be maintained. If convergent, as τ closer to large value which, from the definition of dynamic system, implies that $(X_1^i \quad V_1^i)^T = (X^i \quad V^i)^T$. That is, the dynamic system becomes:

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \omega\varphi^3 & -\varphi_* \\ \omega\varphi^3 & 1-\varphi_* \end{pmatrix} \begin{pmatrix} z \\ x^i \end{pmatrix} + \begin{pmatrix} \varphi_*^1 & \varphi_*^2 \\ \varphi_*^1 & \varphi_*^2 \end{pmatrix} \begin{pmatrix} P^i \\ g_h \end{pmatrix} \tag{6.22}$$

which holds only when $V^i = 0$ and $X^i = P^i = g_h$, where the convergent point is an equilibrium point if there is no external excitation, but better points are found by optimization process with external excitation.

By substituting (6.16) in to (6.17), (6.17) is transformed into (6.23).

$$V_1^i = \omega\varphi^3 z + (\varphi_*^1 + \varphi_*^2)\left[p(i) - x^i\right] \tag{6.23}$$

Notice that, $p(i) = \frac{\varphi_*^1 P^i + \varphi_*^2 g_h}{\varphi_*^1 + \varphi_*^2}$. and setting $p(i) - x^i = y^i$, we get

$$V_1^i = \omega\varphi^3 z + \varphi_* y^i \tag{6.24}$$

$$Y_1^i = \omega\varphi^3 z + (1+\varphi_*)\, y^i \tag{6.25}$$

It follows from (6.24) and (6.25) that

$$\begin{pmatrix} V_i^1 \\ Y_i^1 \end{pmatrix} = \begin{pmatrix} \omega\varphi^3 & \varphi_* \\ \omega\varphi^3 & 1+\varphi_* \end{pmatrix} \begin{pmatrix} z \\ y^i \end{pmatrix} \tag{6.26}$$

which, implies a general forecasting model of the Markov chain; $\eta_t = PK_t^i$, where

$$(\eta_t)^T = \begin{bmatrix} V_1^i & Y_1^i \end{bmatrix} \tag{6.27}$$

$$P^T = \begin{bmatrix} \omega\varphi^3 & \omega\varphi^3 \\ \varphi_* & 1+\varphi_* \end{bmatrix},$$

and

$$(k_t^i)^T = \begin{bmatrix} z & y^i \end{bmatrix}$$

Now, by (C3), we have the probabilistic model (6.27) used to study the evolution of certain systems and the moving analysis of the proposed.

6.4.4 Stability Analysis on the Trajectory of the Proposed PSO-RFC

We further get insight into the dynamic system of closed form of trajectory for the case of $\mathbf{R} = X_t^{i'}$ in equation $X_{t+1}^i = \omega\varphi_t^3 z + \mathbf{R}$ in recursive method:

$$X^i(t+1) = (\omega\varphi_t^3 + 1 - \frac{\varphi_t}{3})x^i(t) - \omega\varphi_t^3 x^i(t-1) + \frac{(\varphi_t^1 P_t^i + \varphi_t^2 g_t)}{3}, \tag{6.28}$$

$$\begin{pmatrix} X^i(t+1) \\ X^i(t) \\ 1 \end{pmatrix} = \begin{pmatrix} \omega\varphi^3 + 1 - \frac{\varphi_t}{3} & -\omega\varphi^3 & \frac{\varphi_t^1 P_t^i + \varphi_t^2 g_t}{3} \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x(t) \\ x(t-1) \\ 1 \end{pmatrix} \tag{6.29}$$

First, we solve the characteristic equation of the dynamic system as follows: $|P^T - I\lambda| = 0$.

$$(1-\lambda)\left[\lambda^2 - (1 + \omega\varphi^3 - \frac{\varphi_t}{3})\lambda + \omega\varphi_t^3\right] = 0. \tag{6.30}$$

The trivial root of (6.30) is $\lambda = 1$ and the other two eigenvalues of (6.29) are obtained as follows: $\lambda_{1,2} = \frac{(1+\varphi_* - \omega\varphi^3 \pm \gamma)}{2}$, where $\gamma = \sqrt{(1 + \varphi_* - \omega\varphi^3)^2 - 4\omega\varphi^3}$ for $\lambda_1 \geq \lambda_2$.
The explicit form solution y recurrence relation (6.17) is then given by

$$Y_1^i(t) = r_1 + r_2\lambda_1^t + r_3\lambda_2^t \tag{6.31}$$

where r_1 , r_2, and r_3 are constants determined by the initial conditions of the system and $t \in \mathbb{N}$ represents the exponent. From updated velocity

$$V_1^i(t+1) = \frac{Y_1^i(t+1) - Y^i(t)}{\triangle\ \tau} \tag{6.32}$$

From (6.29), we have

$$V_1^i(t+1) = (r_2\frac{\lambda_1 - 1}{\triangle\ \tau})\lambda_1^t + (r_3\frac{\lambda_2 - 1}{\triangle\ \tau})\lambda_2^t \tag{6.33}$$

In view of (6.29) and (6.31), letting $k_1 = \frac{r_1(\lambda_1 - 1)}{\triangle\tau}$ and $k_2 = \frac{r_2(\lambda_2 - 1)}{\triangle\tau}$, we obtain

$$\lim_{t\to\infty} V_1^i(t+1) = \lim_{t\to\infty} k_1\lambda_1^t + \lim_{t\to\infty} k_2\lambda_2^t \tag{6.34}$$

Thus, (6.31) and (6.32) yields $max(||\lambda_1||,\ ||\lambda_2||) \leq 1,$ and hence,

Case 1: when $max(||\lambda_1||,\ ||\lambda_2||) < 1,\ X_1^i(t+1) \to x_1^i(t)$ as $t \to \infty$

Case 2: when $max(||\lambda_1||,\ ||\lambda_2||) = 1, V_1^i(t+1) \to (k_1\ or\ k_2\ or\ k_1 + k_2)$ as $t \to \infty$

and consequently (6.33) imply that the trajectory of the particle $X(t)$ in the PSO-RFC algorithm is convergent if $V \to 0$ and will converge to the point (Kim & Xu, 2008).

6.4.5 Trajectories Analysis of PSO-VZRFC Algorithm

Trajectory analysis in PSO provides valuable information on the behavior and performance of the optimization algorithm. We present the influence of the stochastic components in PSO-VZRFC, such as the random term (z) in the velocity update equation, provide a trade-off between exploration and exploitation, as well as the personal best position of the particle p^i and the best position of the entire swarm g_h, are kept constant (6.35) (Gad, 2022).

To balance the local and global search capability of PSO-RFC, we need to improve the global search capability for $\omega = 0$, in update (6.18), if $x^j(t) = p^j = g_h$, particle j will "moving" at the velocity zero and by conserving the current best position of the swarm g_h, and randomly initialize particle j's position x^j_{t+1}, and other particles are manipulated according to

$$X^i_{t+1} = (1 - \varphi_*)x^i_t + \varphi^1_* p^i + \varphi^2_* g_h \tag{6.35}$$

for each $i \in \{1, \ldots, s\}$, this means $P^j = x^j_{t+1}$ defined in the algorithm is given by

$$P^i_t = \begin{cases} x^i_{t+1} : if\ f(x^i_{t+1}) \leq f(P^i_t) \\ P^i_t \quad : otherwise; \end{cases} \tag{6.36}$$

$$g^0_h = arg\ min\{P^i\}^f,$$
$$g_h = arg\ min\{g_h,\ g^0_h\}^f.$$

Thus, experimentation and fine-tuning are often necessary to find the most effective strategies for a given optimization problem. For (6.35), some means of enhancing the manipulation of particles to improve the global search capability in PSO-VZRFC :

1. $p^j = g_h \Longrightarrow x^j_{t+1}$ and other particles are manipulated.
2. $p^j = g_h \nRightarrow x^j_{t+1}$ and other particles are manipulated.
3. If $p^j = g_h$, and changes g_h, there exists an integer $k1j$, which is satisfied $x_k(t+1) = p_k = g_h$, then particle k's position $x_k(t)$ needs to continue initialize randomly and other particles are manipulated according to (6.23),

Due to the particle's position need to uniformly sample from the domain when $x^j(t) = p^j = g_h$, the modified PSO-ZVRFC algorithm called stochastic PSO-ZVRFC. The closed form of the updated equation from part 1 of the manipulation of particles can be obtained using any suitable technique for solving non-homogeneous recurrence relations (NHRRs), assuming that g_h and p^i remain constant while t changes

$$x^i_t = P(i) + (x^i_o - P(i))(1 - \varphi_*)^t.$$

The exact time step at which the equation remains valid until a better position is discovered, this will occur depends on the fitness function, as well as the values of g_h and p^i.

Theorem 1. If $0 < \varphi_* < 2$

$$x_t^i \to g_h \; as \; t \to +\infty. \tag{6.37}$$

Proof. By (6.33) suppose $0 < \varphi_* < 2,\; x_t^i \to P(i)$ as $t \to +\infty$, and

$$X_{t+1}^i = x_t^i - \frac{\varphi^1 + \varphi^2}{\triangle \; \tau} x_t^i + \varphi_*^1 p^i + \varphi_*^2 g_h$$

when $t \to +\infty,\; \lim\limits_{t\to\infty} X^i(t+1) = \lim\limits_{t\to\infty} X^i(t)$.
$\lim\limits_{t\to\infty} \{-(\varphi_*^1 + \varphi_*^2)x_t^i + \varphi_*^1 p^i + \varphi_*^2 g_h\} = 0$. So $\;\; 0 = -(\varphi_*^1 + \varphi_*^2)x_t^i + \varphi_*^1 p^i + \varphi_*^2 g_h$. Because of $\varphi_*^1,\; \varphi_*^2$ are random variables, formula (6.35) is true if and only if $\lim\limits_{t\to\infty} x^i(t) = g_h = p^i$.

6.4.6 Theoretical Analysis on Convergence of PSO-VZRFC

We present theoretical generalizations of the VZRFCP (6.8). The positions of the local attractors and the global attractor converge to a single point O_t^i in the search space (Kim & Xu, 2008; Tilahun, 2022). Without loss of generality, we assume that all entries in O_t^i are zero and therefore $P_t^i = g_{ht} = 0$. Then for a single dimension and particle the dynamic movement in (6.5) and (6.8) degenerates to (6.38), when $r^1,\; r^2$ and r^3 are independent and uniformly distributed random variables on $[0, 1]$ and, $\Omega_{t+1} = (\frac{c_1 r^1}{\triangle\tau} + \frac{c_2 r^2}{\triangle\tau}) = (C_p r^1 + S_g r^2)$ for $C_p = \frac{c_1}{\triangle\tau}$ and $S_g = \frac{c_2}{\triangle\tau}$

$$\begin{aligned} V_{t+1}^i &= \omega c_3 r^3 (0 - V_t^i) - \Omega_{t+1} X_t^i \\ X_{t+1}^i &= \omega c_3 r^3 (0 - V_t^i) + (1 - \Omega_{t+1}) X_t^i \end{aligned} \tag{6.38}$$

The term $\omega c_3 r^3 z = \omega c_3 r^3 (0 - V_t^i)$ introduce a random perturbation that gradually approaches the velocity magnitude to zero for approximately scaled term $\omega c_3 r^3$. PSO converges to $P_t = g_t = 0$ iff $\lim\limits_{t\to\infty} X_t^i = 0$ and $\lim\limits_{t\to\infty} V_t^i = 0$ (Huang et al., 2019).

To analyze the convergence of PSO ZVRFCP, we need to evaluate the position and velocity of the next iteration $(t + 1)$, which can be modeled as (matrix-) multiplication with values of the previous iteration (t) in expected values and variances of positions of particles. To verify whether a random variable converges or not, we show (discuss) the measure of the expected value (E) and the variance (var) of the position and velocity based on the limit and closed form approach as presented in the next examples (Kim & Xu, 2008; Piotrowski et al., 2020).

1. The expected value and the variance of position

   Let all the variables are defined within the context of a common probability (P) space.

   Example 6.1. Let $\{X_t\}_{t\geq 0}$ be a random process such that $X_0 = 1$, $P[X_{t+1} = e \cdot X_t] = 1/2$

and $P[X_{t+1} = e^{-2} \cdot X_t] = 1/2$ where $(e \approx 2.718)$ we have that $\lim_{t\to\infty} P[x_t < \epsilon > 0] = 1$. Nevertheless, $\lim_{t\to\infty} E[X_t] = \frac{(e+e^{-2})}{2} \cdot E[X_{t-1}]= \infty$ and $\lim_{t\to\infty} Var[X_t] = \infty$

Example 6.2. If we analyze logarithmic scale $Y_t := ln(X_t)$ we have $Y_0 = 0$, $P[Y_{t+1} = Y_t + 1] = 1/2$ and $P[Y_{t+1} = Y_t - 2] = 1/2$. Here $E[Y_t] = (1-2)/2 + E[Y_{t-1}] = \lim_{t\to\infty}(-t/2) = -\infty$ and $E[Y_{t+1} - Y_t] < -1/2 < 0$.

This yields of Y_o to ∞ and also X_t to ∞.

2 The expected value and the variance of position and velocity.

A better statistic for analysis is the logarithm of position. To avoid numerical instabilities, we use the other logarithmic scale $\lim_{t\to\infty} ln(X_t^2 + V_t^2) = -\infty$ as $(V_t^i \to 0)$ (Chen et al., 2009).

From the definition of Ω_{t+1} and (6.38), we have the positions and velocities of iteration t and $t+1$ by the position x_{t-1} for $\omega = 0$,

$$\begin{cases} x_t = (1-\Omega_t)\cdot x_{t-1} \\ x_{t+1} = (1-\Omega_t)\cdot(1-\Omega_{t+1})\cdot x_{t-1} \end{cases} \qquad \begin{cases} v_t = -\Omega_t \cdot x_{t-1} \\ v_{t+1} = -(1-\Omega_t)\cdot\Omega_{t+1}\cdot x_{t-1} \end{cases} \tag{6.39}$$

$$E[\Omega_{t+1} - \Omega_t] = E[ln(X_{t+1}^2 + V_{t+1}^2) - ln(X_t^2 + V_t^2)]$$

$$E[\Omega_{t+1} - \Omega_t] = E[ln(1-\Omega_t)^2 + ln(\Omega_{t+1}^2 + (1-\Omega_{t+1})^2) - ln((1-\Omega_t)^2 + \Omega_t^2)]$$

Assume $\Omega_t \approx \Omega_{t+1}$,

$$E[\Omega_{t+1} - \Omega_t] = E[ln(1-\Omega_t)^2] = E[ln(1 - C_p r^1 - S_g r^2)^2] \tag{6.40}$$

Thus, (6.39) and (6.40) deliver

$$E[\Omega_{t+1} - \Omega_t] = \int_0^1 \int_0^1 ln(1 - C_p r^1 - S_g r^2)^2 dr^1 dr^2$$

For both variables of integration, after evaluating them from 0 to 1, we obtain

$$\begin{aligned} E[\Omega_{t+1} - \Omega_t] = \frac{1}{2C_pS_g}&[(1-C_p-S_g)^2(ln(1-C_p-S_g)^2 - 1) - (1-S_g)^2(ln(1-S_g)^2 - 1) \\ &- (1-C_p)^2(ln(1-C_p)^2 - 1)] - 3 \end{aligned} \tag{6.41}$$

To simplify (6.41), we consider the following cases

Case 1. If $CS = C_p = S_g$ (42) is negative iff $0 < CS < \delta < 2.4$

$$E[\Omega_{t+1} - \Omega_t] = \frac{1}{2CS^2}[(1-2CS)^2 \, ln(1-2CS)^2 - 2(1-CS)^2 \, ln(1-CS)^2] - 3 \tag{6.42}$$

Case 2. For $\omega = 0$ and $C_p = 0,\ S_g = CS$ or $C_p = CS,\ S_g = 0$, we obtain a slightly different result for evaluating r^1 from 0 to 1

$$E\Big[ln(1-C_pr^1-S_gr^2)^2\Big] = \int_0^1 ln(1-CSr^1)^2dr^1 = \left[\frac{(1-CSr)}{-CS}\,ln(1-CSr^1)^2+\frac{2}{CS}-2r^1\right]_0^1$$

$$= \frac{CS-1}{CS}\,ln(1-CS)^2-2$$

Case 3. By letting $C_p \to 0$, or $S_g \to 0$, and from evaluation of (6.41) it follows that the required value is negative and hence $\lim_{t\to\infty} ln(X_t^2+V_t^2) = -\infty$.

We can approach the convergence by measuring the variance of particles position as of

$$var[\Omega_{t+1}-\Omega_t] = E[\{\Omega_{t+1}-\Omega_t\}^2] - \{E[\Omega_{t+1}-\Omega_t]\}^2$$

$$= E[\{ln(1-\Omega_t)^2\}^2] - \{E[ln(1-\Omega_t)^2]\}^2$$

and considering all of the cases of expected value of particle position.

6.5 Numerical Simulations

We present a numerical example to illustrate the performance of the proposed algorithm in comparison with (Kim & Xu, 2008; Tilahun, 2022). In this experiment, we compare our basic algorithm with the proposed algorithm for different coefficient values of $c_1,\ c_2$ and c_3 on some test functions. For randomly generated starting points $x_0,\ x_1,\ x_2 \in \mathbb{R}^D$, the experimental results are reported in Tables 6.2 & 6.3 and Fig. 6.1 & 6.2.

Example 6.3. Unimodal function

$$\begin{gathered} min\ f(x_i) = \sum_{i=1}^{K} x_i^2 \\ subject\ to -10 \leq x_i \leq 10. \end{gathered} \tag{6.43}$$

Example 6.4. Multi modal function

$$\begin{gathered} min\ f(x) = \sum_{i=1}^{K=3}[x_i^2 - 10cos(2\pi x_i)+10] \\ subject\ to -10 \leq x_i \leq 10. \end{gathered} \tag{6.44}$$

Tables 6.2 & 6.3 show the algorithm components taken versus the best solution. The figure shows the best cost versus the number of iterations.

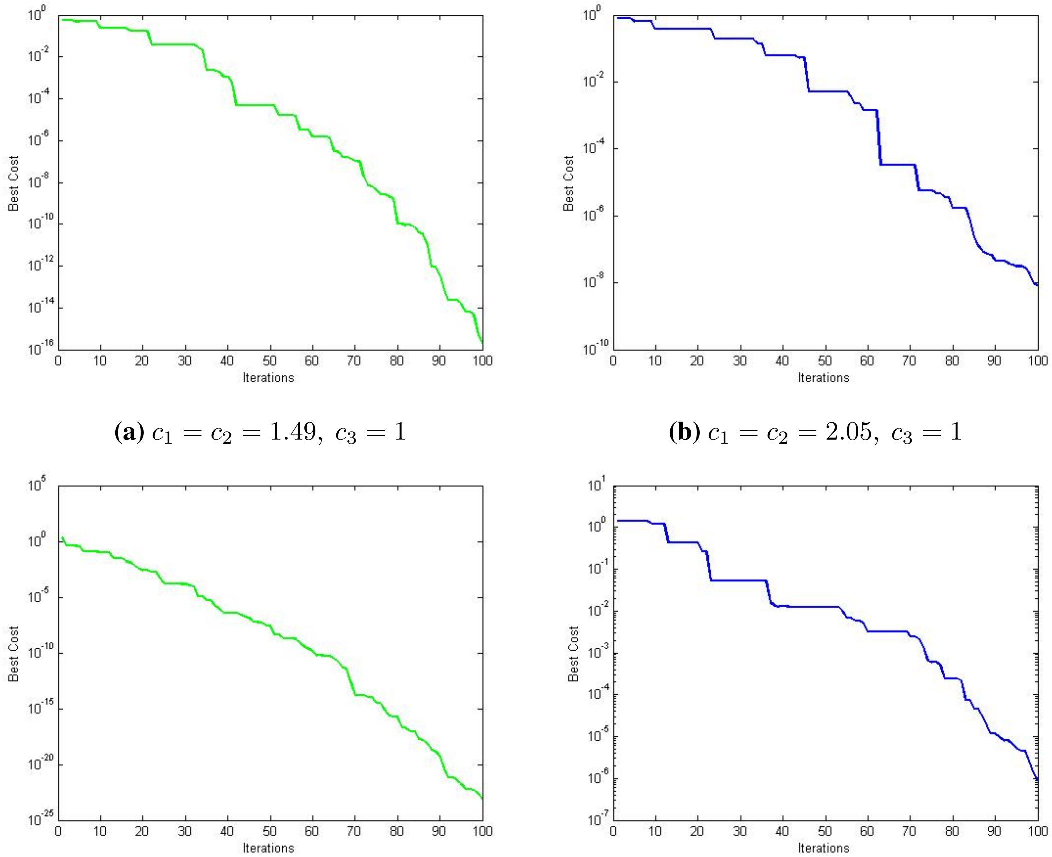
Table 6.2: Comparison of algorithms on optimization test functions

Algorithm components			Best soln.			
			Best cost.		position	
N$\underline{o}$. of runs	Algorithms	coefficients	mini.	maxi.		Figures
1.	Basic PSO	$c_1 = c_2 = 1.49$	$1.9343e-16$	0.5791	$[1.5007e-11-1.3692e-08-2.4412e-09]$	(a)
2.	SPSO	$c_1 = c_2 = 2.05$	$7.8605e-09$	0.7953	$[-8.1651e-06-2.5276e-05-8.4587e-05]$	(b)
3.	PSO-RFC	$c_1 = c_2 = c_3 = 1.49$	$7.0162e-24$	2.2416	$[-1.8108e-121.9301e-12-1.0947e-13]$	(c)
4.	PSO-RFC	$c_1 = c_2 = c_3 = 2.05$	$9.0924e-07$	1.4266	$[-6.4727e-05-2.4419e-04-9.1947e-04]$	(d)

Table 6.3: Comparison of algorithms on optimization test functions

Algorithm components			Best soln.			
			Best cost.		position	
N$\underline{o}$. of runs	Algorithms	coefficients	mini.	maxi.		Figures
1.	Basic PSO	$c_1 = c_2 = 2.05, c_3 = 1$	$7.3594e-12$	9.3504	$[-1.6511e-077.0782e-08-6.9422e-08]$	(a)
2.	SPSO	$c_1 = c_2 = 1.49, c_3 = 1$	0	23.5017	$[2.4279e-101.4731e-097.5424e-10]$	(b)
3.	PSO-RFC	$c_1 = c_2 = c_3 = 1.49$	0	28.8570	$[-1.2966e-097.5164e-105.5904e-10]$	(c)
4.	PSO-RFC	$c_1 = c_2 = c_3 = 2.05$	0.9960	20.3708	$[0.0020-0.9959-3.6091e-04]$	(d)

Fig.6.1 shows that for all visualized parameter sets the numerically evaluated limit distribution coincides with the experimentally evaluated PSO coefficients, all the figures $(a-d)$ meet the objective of the PSO-VZRFC algorithm for the optimization problem given in (6.43).



(a) $c_1 = c_2 = 1.49,\ c_3 = 1$

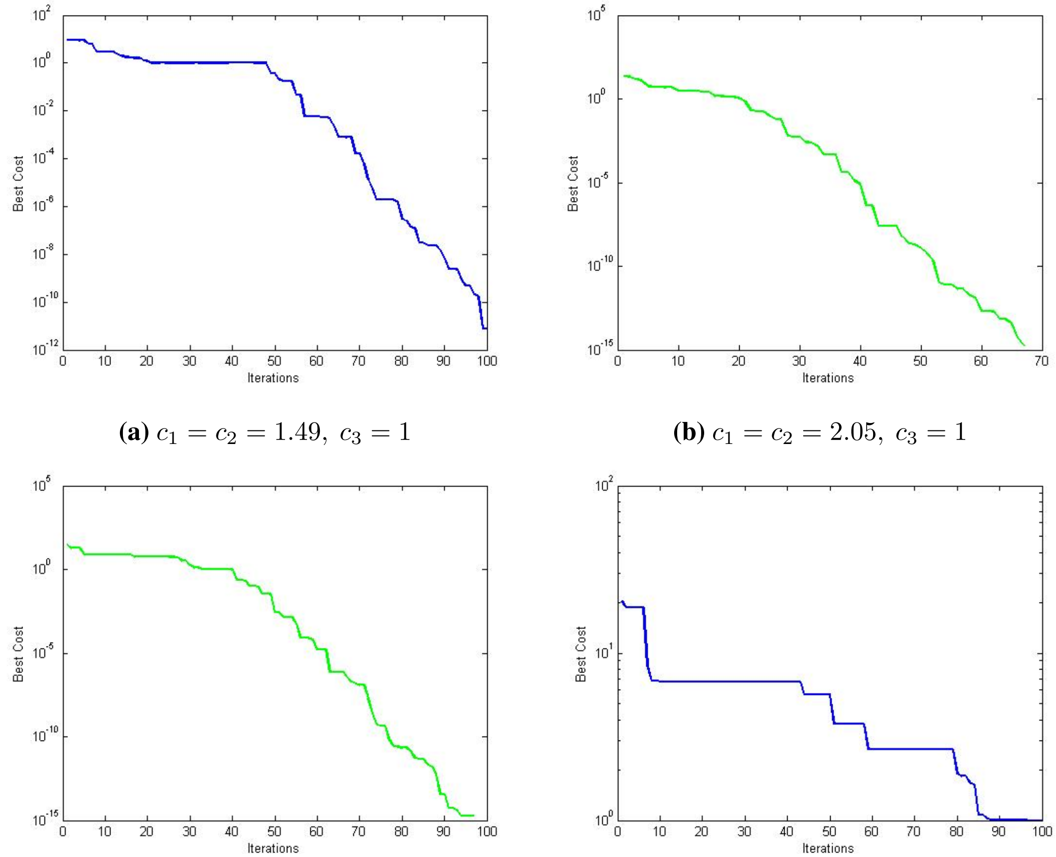
(b) $c_1 = c_2 = 2.05,\ c_3 = 1$

(c) $c_1 = c_2 = c_3 = 1.49$

(d) $c_1 = c_2 = c_3 = 2.05$

Figure 6.1: Evaluation of the algorithms on Example 6.3 .

The numerical results in Fig.6.2 (a) and Fig.6.2 (d) and in Tables 6.2 & 6.3 show that our algorithm PSO-RFC converges faster than the rest algorithms. Moreover, Tables 6.2 & 6.3 illustrate that, compared to the algorithms of the others, our algorithm PSO-RFC takes a very small number of iterations to reach the best solution required (optimum solution).



(a) $c_1 = c_2 = 1.49,\ c_3 = 1$

(b) $c_1 = c_2 = 2.05,\ c_3 = 1$

(c) $c_1 = c_2 = c_3 = 1.49$

(d) $c_1 = c_2 = c_3 = 2.05$

Figure 6.2: Evaluation of the algorithms on Example 6.4.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

In this dissertation, variants of PSO algorithms were proposed and the impact of algorithm parameters on convergence was considered. Qualitatively and quantitatively, these algorithms have been analyzed. An application of Markov chain theory is applied to determine the best optimal strategy. The cost-effective analysis is then implemented to describe the lowest-cost strategy to minimize the computation, or output stress required in PSO with limited resources.

In Chapters $1-3$, we studied some basic features of the PSO algorithm, historical background, and the various mathematical algorithms investigated by the authors.

In Chapter 4, we can draw the following important conclusions from the convergence analyzes of PSO related to its convergence properties, transformation invariance, component modification, coefficient adaptation, population size, topology, and hybridization.

- The first part of this chapter concentrated on limitations of PSO that have been theoretically investigated by classifying into two main groups: convergence and transformation invariance.
- The convergence properties of PSO was pointed out that by analyzing different behaviors of particles before convergence and convergence analysis including convergence to a fixed point, local convergence, and stagnation of other PSO variants.
- The importance of the transformation invariance property in convergence analyzes of PSO algorithms for developing robust and versatile optimization techniques.
- Apart from theoretical studies, the algorithm parameters in PSO are usually determined based on specific problems, application experiences, and numerous experiment tests related to modifications of components and PSO parameters. The topology of the swarm, the setting coefficients, and the population size are PSO parameters that were investigated in this study.
- More attention should be paid to the highly efficient PSO algorithm and put forward a suitable core update formula and an effective strategy to balance global exploration and local exploitation.

In view of the above-mentioned analyses, the convergence analyses of PSOs studied under certain rigid assumptions that simplify problem models, the advantages of algorithms could be summarized as follows:

- It is robust and can be used in different application areas with a small modification.
- It has a strong distributed ability.
- It can quickly converge to the optimal value.
- It is easy to combine with other algorithms to improve performance.

In Chapter 5, this study mainly concerned with the convergence and stability analysis of the CSPSO algorithm, and its performance improvement for different constriction coefficients. We first investigated the convergence of the SPSO algorithm by relating it to the Markov chain in which the stochastic process and Markov properties employ quantum behaviors to improve the global convergence and prove Markov chain transition probability showing that the CSPSO algorithm converges to the global optimum in probability. We also compared the proposed algorithm with basic PSO, SPSO and CPSO algorithms evaluating the optimal value (fitness value) based on the range of ω. The proposed algorithm is fast and efficient, and the CSPSO run plans for ω linearly decreasing from 0.9 to 0.4 are easy to implement. The CSPSO algorithm performs better because it regenerated those results to minimize test functions. On the other hand, the proposed heuristic algorithm did not seek solutions that minimized the delay time or cost function, and the adjustment process would be stopped if ω was not identified as regular. The CSPSO algorithm is verified to be a global convergent algorithm.

In Chapter 6, the proposed method integrates RFC and VZSPSO, referred to as PSO-RFC. However, theoretical guarantees regarding local optimality and convergence properties have not yet been established. This study proposes a mapping from VZRFC to the three coefficients of the PSO component. The mapping is designed to ensure the accelerated convergence of RFC-PSO and its improved speed compared to SPSO and other variants. The numerical examples demonstrate the theoretical and practical advantages of our algorithms. PSO-RFC proves to be an efficient optimization approach for benchmark functions.

Future Research

In this dissertation, the future research objective is to formally prove the mentioned aspects of the swarm's behavior in situations with more than one dimension as follows:

- This study focused exclusively on the convergence of PSO. However, the first hitting time could be strongly determined by the PSO convergence. Thus, the impact of the first hit time on the convergence of the PSO is the fundamental area of research to be investigated.
- A basic field of research to be investigated by including an additional parameter defined by the user in our algorithm for the optimization of standard particle swarm.
- An extension of the algorithms in this study to take into account how to obtain a strong convergence algorithm. So we can develop modified PSO algorithms in the configuration of constrained spaces.

Recommendations

Based on the above results and discussion we observed various parameter values applied on the basic PSO algorithm made the convergence to optimal solution with greater speed and accuracy than works in meta-heuristic PSO algorithm family/ group. In order to obtain best convergent PSO algorithm, we recommend the following based on our study

- Researchers in the field of optimization should focus on the most influential parameters, as experiments suggest that utilizing the smallest number of parameters can yield significant results, even with limited resources.
- Incorporate local search methods to refine solutions, helping particles converge more quickly to local optima after global exploration.
- Methods to maintain diversity within the swarm, such as introducing random perturbations or selecting diverse leaders, can prevent premature convergence.
- Combine PSO with other optimization algorithms to leverage their strengths for improved convergence.
- Implement techniques like constriction factors or acceleration coefficients that help particles converge more rapidly toward optimal solutions.

REFERENCES

Agrafiotis, D. K., & Cedeño, W. (2002). Feature selection for structure- activity correlation using binary particle swarms. *Journal of medicinal chemistry*, *45*(5), 1098–1107.

Alatas, B., Akin, E., & Ozer, A. B. (2009). Chaos embedded particle swarm optimization algorithms. *Chaos, Solitons & Fractals*, *40*(4), 1715–1734.

AlRashidi, M. R., & El-Hawary, M. E. (2008). A survey of particle swarm optimization applications in electric power systems. *IEEE transactions on evolutionary computation*, *13*(4), 913–918.

Anthony, C. (2001). An off-the-shelf pso. In *Proceedings of the workshop on particle swarm optimization, indianapolis, 2001.*

Antoniou, P., Pitsillides, A., Blackwell, T., & Engelbrecht, A. (2009). Employing the flocking behavior of birds for controlling congestion in autonomous decentralized networks. In *2009 ieee congress on evolutionary computation* (pp. 1753–1761).

Antoniou, P., Pitsillides, A., Blackwell, T., Engelbrecht, A., & Michael, L. (2013). Congestion control in wireless sensor networks based on bird flocking behavior. *Computer Networks*, *57*(5), 1167–1191.

Arabas, J., Michalewicz, Z., & Mulawka, J. (1994). Gavaps-a genetic algorithm with varying population size. In *Proceedings of the first ieee conference on evolutionary computation. ieee world congress on computational intelligence* (pp. 73–78).

Ardizzon, G., Cavazzini, G., & Pavesi, G. (2015). Adaptive acceleration coefficients for a new search diversification strategy in particle swarm optimization algorithms. *Information Sciences*, *299*, 337–378.

Armaghani, D. J., Hajihassani, M., Bejarbaneh, B. Y., Marto, A., & Mohamad, E. T. (2014). Indirect measure of shale shear strength parameters by means of rock index tests through an optimized artificial neural network. *Measurement*, *55*, 487–498.

Arumugam, M. S., Rao, M. V. C., & Tan, A. W. (2009). A novel and effective particle swarm optimization like algorithm with extrapolation technique. *Applied soft computing*, *9*(1), 308–320.

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

Banks, A., Vincent, J., & Anyakoha, C. (2007). A review of particle swarm optimization. part i: background and development. *Natural Computing*, *6*, 467–484.

Bansal, J. C. (2019). Particle swarm optimization. *Evolutionary and swarm intelligence algorithms*, 11–23.

Bansal, J. C., Singh, P., Saraswat, M., Verma, A., Jadon, S. S., & Abraham, A. (2011). Inertia weight strategies in particle swarm optimization. In *2011 third world congress on nature and biologically inspired computing* (pp. 633–640).

Beni, G. (2020). Swarm intelligence. *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*, 791–818.

Birbil, Ş. İ., Fang, S.-C., & Sheu, R.-L. (2004). On the convergence of a population-based global optimization algorithm. *Journal of global optimization*, *30*, 301–318.

Bonyadi, M. R. (2019). A theoretical guideline for designing an effective adaptive particle swarm. *IEEE Transactions on Evolutionary Computation*, *24*(1), 57–68.

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

Bonyadi, M. R., & Michalewicz, Z. (2014a). A locally convergent rotationally invariant particle swarm optimization algorithm. *Swarm intelligence*, *8*(3), 159–198.

Bonyadi, M. R., & Michalewicz, Z. (2014b). Spso 2011: analysis of stability; local convergence; and rotation sensitivity. In *Proceedings of the 2014 annual conference on genetic and evolutionary computation* (pp. 9–16).

Bonyadi, M. R., & Michalewicz, Z. (2015a). Analysis of stability, local convergence, and transformation sensitivity of a variant of the particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, *20*(3), 370–385.

Bonyadi, M. R., & Michalewicz, Z. (2015b). Stability analysis of the particle swarm optimization without stagnation assumption. *IEEE Transactions on Evolutionary Computation*, *20*(5), 814–819.

Bonyadi, M. R., & Michalewicz, Z. (2017). Particle swarm optimization for single objective continuous space problems: a review. *Evolutionary computation*, *25*(1), 1–54.

Bonyadi, M. R., Michalewicz, Z., & Li, X. (2014). An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal of Heuristics*, *20*, 417–452.

Brits, R., Engelbrecht, A. P., & van den Bergh, F. (2007). Locating multiple optima using particle swarm optimization. *Applied mathematics and computation*, *189*(2), 1859–1883.

Cai, Z.-Q., Huang, H., Zheng, Z.-H., & Luo, W. (2009). Convergence improvement of particle swarm optimization based on the expanding attaining-state set. *Journal of Huazhong University of Science and Technology (Nature Science Edition)*, *37*(6), 44–47.

Campana, E. F., Fasano, G., & Pinto, A. (2010). Dynamic analysis for the selection of parameters and initial population, in particle swarm optimization. *Journal of Global Optimization*, *48*, 347–397.

Chatterjee, A., & Siarry, P. (2006). Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & operations research*, *33*(3), 859–871.

Chen, J., Xin, B., Peng, Z., Dou, L., & Zhang, J. (2009). Optimal contraction theorem for exploration–exploitation tradeoff in search and optimization. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *39*(3), 680–691.

Cheng, Y. M., Li, L., & Chi, S. (2007). Performance studies on six heuristic global optimization methods in the location of critical slip surface. *Computers and Geotechnics*, *34*(6), 462–484.

Cleghorn, C. W., & Engelbrecht, A. P. (2014a). A generalized theoretical deterministic particle swarm model. *Swarm intelligence*, *8*, 35–59.

Cleghorn, C. W., & Engelbrecht, A. P. (2014b). Particle swarm convergence: An empirical investigation. In *2014 ieee congress on evolutionary computation (cec)* (pp. 2524–2530).

Cleghorn, C. W., & Engelbrecht, A. P. (2015). Particle swarm variants: standardized convergence analysis. *Swarm Intelligence*, *9*, 177–203.

Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the 1999 congress on evolutionary computation-cec99 (cat. no. 99th8406)* (Vol. 3, pp. 1951–1957).

Clerc, M. (2011). Standard particle swarm optimisation from 2006 to 2011. *Particle Swarm Central*, *253*.

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

Cui, Z., & Gao, X. (2012). *Theory and applications of swarm intelligence.* Springer.

Cui, Z., Gao, X.-Z., & Deb, S. (2019). *Theory and applications of soft computing methods* (Vol. 31). Springer.

Dasgupta, S., Das, S., Biswas, A., & Abraham, A. (2009). On stability and convergence of the population-dynamics in differential evolution. *Ai Communications*, *22*(1), 1–20.

Deb, K., & Deb, D. (2014). Analysing mutation schemes for real-parameter genetic algorithms. *International Journal of Artificial Intelligence and Soft Computing*, *4*(1), 1–28.

De Oca, M. A. M., Stutzle, T., Birattari, M., & Dorigo, M. (2009). Frankenstein's pso: a composite particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation*, *13*(5), 1120–1132.

De Oca, M. A. M., Stutzle, T., Van den Enden, K., & Dorigo, M. (2010). Incremental social learning in particle swarms. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *41*(2), 368–384.

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18.

Doerr, B. (2020). Probabilistic tools for the analysis of randomized optimization heuristics. *Theory of evolutionary computation: Recent developments in discrete optimization*, 1–87.

Dong, W. Y., & Zhang, R. R. (2019). Order-3 stability analysis of particle swarm optimization. *Information Sciences*, *503*, 508–520.

Dorea, C. (1983). Expected number of steps of a random optimization method. *Journal of Optimization Theory and Applications*, *39*(2), 165–171.

Dorigo, M., & Blum, C. (2005). Ant colony optimization theory: A survey. *Theoretical computer science*, *344*(2-3), 243–278.

Dorigo, M., & Stützle, T. (2019). *Ant colony optimization: overview and recent advances*. Springer.

Eberhart. (2001). *Swarm intelligence*. Elsevier.

Eberhart, & Kennedy. (1995). A new optimizer using particle swarm theory. In *Mhs'95. proceedings of the sixth international symposium on micro machine and human science* (pp. 39–43).

Eberhart, & Shi. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the 2000 congress on evolutionary computation. cec00 (cat. no. 00th8512)* (Vol. 1, pp. 84–88).

Eiben, Á. E., Hinterding, R., & Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on evolutionary computation*, *3*(2), 124–141.

Elsheikh, A., & Abd Elaziz, M. (2019). Review on applications of particle swarm optimization in solar energy systems. *International Journal of Environmental Science and Technology*, *16*, 1159–1170.

Engelbercht, A. (2005). Fundamentals of computational swarm intelligence. *University of Pretoria, South Africa*.

Engelbrecht. (2006). *Fundamentals of computational swarm intelligence*. John Wiley & Sons, Inc.

Engelbrecht. (2007). *Computational intelligence: an introduction*. John Wiley & Sons.

Engelbrecht. (2012). Particle swarm optimization: Velocity initialization. In *2012 ieee congress on evolutionary computation* (pp. 1–8).

Ezugwu, A. E., Agushaka, J. O., Abualigah, L., Mirjalili, S., & Gandomi, A. H. (2022). Prairie dog optimization algorithm. *Neural Computing and Applications*, *34*(22), 20017–20065.

Fan, Q., & Yan, X. (2014). Self-adaptive particle swarm optimization with multiple velocity strategies and its application for p-xylene oxidation reaction process optimization. *Chemometrics and Intelligent Laboratory Systems*, *139*, 15–25.

Feng, P., Xiao-Ting, L., Qian, Z., Wei-Xing, L., & Qi, G. (2013). Analysis of standard particle swarm optimization algorithm based on markov chain. *Acta Automatica Sinica*, *39*(4), 381–389.

Flori, A., Oulhadj, H., & Siarry, P. (2022). Quantum particle swarm optimization: An auto-adaptive pso for local and global optimization. *Computational Optimization and Applications*, *82*(2), 525–559.

Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. *Archives of computational methods in engineering*, *29*(5), 2531–2561.

García-Gonzalo, E., & Fernández-Martínez, J. L. (2014). Convergence and stochastic stability analysis of particle swarm optimization variants with generic parameter distributions. *Applied Mathematics and Computation*, *249*, 286–302.

Gholizadeh, S., & Salajegheh, E. (2009). Optimal design of structures subjected to time history loading by swarm intelligence and an advanced metamodel. *Computer Methods in Applied Mechanics and Engineering*, *198*(37-40), 2936–2949.

Goel, S., & Panchal, V. (2014). Performance evaluation of a new modified firefly algorithm. In *Proceedings of 3rd international conference on reliability, infocom technologies and optimization* (pp. 1–6).

Hansen, N. (2000). Invariance, self-adaptation and correlated mutations in evolution strategies. In *International conference on parallel problem solving from nature* (pp. 355–364).

Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of ieee international conference on evolutionary computation* (pp. 312–317).

Hansen, N., Ros, R., Mauny, N., Schoenauer, M., & Auger, A. (2008). *Pso facing non-separable and ill-conditioned problems* (Unpublished doctoral dissertation). INRIA.

Hansen, N., Ros, R., Mauny, N., Schoenauer, M., & Auger, A. (2011). Impacts of invariance in search: When cma-es and pso face ill-conditioned and non-separable problems. *Applied Soft Computing*, *11*(8), 5755–5769.

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 (cec)* (pp. 349–356).

Hasanipanah, M., Noorian-Bidgoli, M., Armaghani, D. J., & Khamesi, H. (2016). Feasibility of pso-ann model for predicting surface settlement caused by tunneling. *Engineering with Computers*, *32*(4), 705–715.

Hassan, R. (2004). Particle swarm optimization: Method and applications. *Presentation at http://ocw. mit. edu*.

He, J., & Yao, X. (2002). From an individual to a population: An analysis of the first hitting time of population-based evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *6*(5), 495–511.

Helwig, S., Branke, J., & Mostaghim, S. (2012). Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary computation*, *17*(2), 259–271.

Helwig, S., & Wanka, R. (2007). Particle swarm optimization in high-dimensional bounded search spaces. In *2007 ieee swarm intelligence symposium* (pp. 198–205).

Helwig, S., & Wanka, R. (2008). Theoretical analysis of initial particle swarm behavior. In *International conference on parallel problem solving from nature* (pp. 889–898).

Hossain, M. S., & El-Shafie, A. (2014). Evolutionary techniques versus swarm intelligences: application in reservoir release optimization. *Neural Computing and Applications*, *24*(7), 1583–1594.

Hsieh, S.-T., Sun, T.-Y., Liu, C.-C., & Tsai, S.-J. (2008). Efficient population utilization strategy for particle swarm optimizer. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *39*(2), 444–456.

Hu, D., Qiu, X., Liu, Y., & Zhou, X. (2021). Probabilistic convergence analysis of the stochastic particle swarm optimization model without the stagnation assumption. *Information Sciences*, *547*, 996–1007.

Hu, M., Wu, T., & Weir, J. D. (2012). An intelligent augmentation of particle swarm optimization with multiple adaptive methods. *Information Sciences*, *213*, 68–83.

Huang, H., Lv, L., Ye, S., & Hao, Z. (2019). Particle swarm optimization with convergence speed controller for large-scale numerical optimization. *Soft Computing*, *23*, 4421–4437.

Huang, H., Qin, H., Hao, Z., & Lim, A. (2012). Example-based learning particle swarm optimization for continuous optimization. *Information Sciences*, *182*(1), 125–138.

Jägersküpper, J. (2008). Lower bounds for randomized direct search with isotropic sampling. *Operations research letters*, *36*(3), 327–332.

Jain, N., Nangia, U., & Jain, J. (2018). A review of particle swarm optimization. *Journal of The Institution of Engineers (India): Series B*, *99*, 407–411.

Janson, S., & Middendorf, M. (2005). A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(6), 1272–1282.

Jiang, M., Luo, Y. P., & Yang, S. Y. (2007a). Particle swarm optimization-stochastic trajectory analysis and parameter selection. *Swarm intelligence, Focus on ant and particle swarm optimization*, 179–198.

Jiang, M., Luo, Y. P., & Yang, S. Y. (2007b). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information processing letters*, *102*(1), 8–16.

Kadirkamanathan, V., Selvarajah, K., & Fleming, P. J. (2006). Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation*, *10*(3), 245–255.

Kalatehjari, R. (2013). *An improvised three-dimensional slope stability analysis based on limit equilibrium method by using particle swarm optimization* (Unpublished doctoral dissertation). Universiti Teknologi Malaysia.

Kao, Y.-T., & Zahara, E. (2008). A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Applied soft computing*, *8*(2), 849–857.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of global optimization*, *39*(3), 459–471.

Karafotias, G., Hoogendoorn, M., & Eiben, Á. E. (2014). Parameter control in evolutionary algorithms: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, *19*(2), 167–187.

Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 congress on evolutionary computation-cec99 (cat. no. 99th8406)* (Vol. 3, pp. 1931–1938).

Kennedy, J. (2003). Bare bones particle swarms. In *Proceedings of the 2003 ieee swarm intelligence symposium. sis'03 (cat. no. 03ex706)* (pp. 80–87).

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of icnn'95-international conference on neural networks* (Vol. 4, pp. 1942–1948).

Kennedy, J., & Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm. In *1997 ieee international conference on systems, man, and cybernetics. computational cybernetics and simulation* (Vol. 5, pp. 4104–4108).

Kennedy, J., & Mendes, R. (2002). Population structure and particle swarm performance. In *Proceedings of the 2002 congress on evolutionary computation. cec'02 (cat. no. 02th8600)* (Vol. 2, pp. 1671–1676).

Kim, T.-H., & Xu, H.-K. (2008). Convergence of the modified mann's iteration method for asymptotically strict pseudo-contractions. *Nonlinear Analysis: Theory, Methods & Applications*, *68*(9), 2828–2836.

Kumar. (2021). Particle swarm optimization: The foundation. In *Applying particle swarm optimization: New solutions and cases for optimized portfolios* (pp. 97–110). Springer.

Kumar, A., Singh, B. K., & Patro, B. (2016). Particle swarm optimization: a study of variants and their applications. *International Journal of Computer Applications*, *135*(5), 24–30.

Kumar, D., Jain, N., & Nangia, U. (2021). Perfectly convergent particle swarm optimisation in multidimensional space. *International journal of Bio-inspired computation*, *18*(4), 221–228.

Kuok, K. K., Harun, S., & Shamsuddin, S. (2010). Particle swarm optimization feedforward neural network for modeling runoff. *International Journal of Environmental Science & Technology*, *7*(1), 67–78.

Lawler, G. F. (2018). *Introduction to stochastic processes*. Chapman and Hall/CRC.

Lee, J., & Mukerji, T. (2014). *Probabilistic particle swarm optimization (pro-pso) for using prior information and hierarchical parameters.* Department of Energy Resources Engineering, Stanford University.

Lehre, P. K., & Witt, C. (2013). Finite first hitting time versus stochastic convergence in particle swarm optimisation. In *Advances in metaheuristics* (pp. 1–20). Springer.

Li, & Qian. (2003). Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques. *Journal of circuits and systems*, *1*, 1–6.

Li, C., Yang, S., & Nguyen, T. T. (2011). A self-learning particle swarm optimizer for global optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *42*(3), 627–646.

Liang, J.-J., & Suganthan, P. N. (2005). Dynamic multi-swarm particle swarm optimizer. In *Proceedings 2005 ieee swarm intelligence symposium, 2005. sis 2005.* (pp. 124–129).

Liu. (2015). Order-2 stability analysis of particle swarm optimization. *Evolutionary computation*, *23*(2), 187–216.

Liu, H., Zhang, X.-W., & Tu, L.-P. (2020). A modified particle swarm optimization using adaptive strategy. *Expert systems with applications*, *152*, 113353.

Liu, J., Ma, X., Li, X., Liu, M., Shi, T., & Li, P. (2021). Random convergence analysis of particle swarm optimization algorithm with time-varying attractor. *Swarm and evolutionary computation*, *61*, 100819.

Liu, Q., Wei, W., Yuan, H., Zhan, Z.-H., & Li, Y. (2016). Topology selection for particle swarm optimization. *Information Sciences*, *363*, 154–173.

Løvbjerg, M., Rasmussen, T., & Krink, T. (2001). Hybrid particle swarm optimizer with breeding and subpopulations in: Proceedings of the genetic and evolutionary computation conference (gecco-2001).

Lu, Z.-s., Hou, Z.-r., & Du, J. (2006). Particle swarm optimization with adaptive mutation. *Frontiers of Electrical and Electronic Engineering in China*, *1*(1), 99–104.

Marino, G., Scardamaglia, B., & Karapinar, E. (2016). Strong convergence theorem for strict pseudo-contractions in hilbert spaces. *Journal of Inequalities and Applications*, *2016*, 1–12.

Mashhadban, H., Kutanaei, S. S., & Sayarinejad, M. A. (2016). Prediction and modeling of mechanical properties in fiber reinforced self-compacting concrete using particle swarm optimization algorithm and artificial neural network. *Construction and Building Materials*, *119*, 277–287.

Matyas, J., et al. (1965). Random optimization. *Automation and Remote control*, *26*(2), 246–253.

Medasani, S., & Owechko, Y. (2005). Possibilistic particle swarms for optimization. In *Applications of neural networks and machine learning in image processing ix* (Vol. 5673, pp. 82–89).

Mehmood, Y., Sadiq, M., Shahzad, W., & Amin, F. (2018). Fitness-based acceleration coefficients to enhance the convergence speed of novel binary particle swarm optimization. In *2018 international conference on frontiers of information technology (fit)* (pp. 355–360).

Mendes, R., Kennedy, J., & Neves, J. (2004). The fully informed particle swarm: simpler, maybe better. *IEEE transactions on evolutionary computation*, *8*(3), 204–210.

Mirjalili, S. (2018). *Evolutionary algorithms and neural networks: Theory and applications* (Vol. 780). Springer.

Moler, C., & Van Loan, C. (2003). Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review*, *45*(1), 3–49.

Mukherjee, S., & Chatterjee, A. (2006). Unified concept of bottleneck.

Muller, C. L., Baumgartner, B., & Sbalzarini, I. F. (2009). Particle swarm cma evolution strategy for the optimization of multi-funnel landscapes. In *2009 ieee congress on evolutionary computation* (pp. 2685–2692).

Najafzadeh, M., & Tafarojnoruz, A. (2016). Evaluation of neuro-fuzzy gmdh-based particle swarm optimization to predict longitudinal dispersion coefficient in rivers. *Environmental Earth Sciences*, *75*(2), 157.

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.

Neshat, M. (2013). Faipso: fuzzy adaptive informed particle swarm optimization. *Neural Computing and Applications*, *23*(1), 95–116.

Neshat, M., Sepidnam, G., & Sargolzaei, M. (2013). Swallow swarm optimization algorithm: a new method to optimization. *Neural Computing and Applications*, *23*(2), 429–454.

Ni, Q., & Deng, J. (2013). A new logistic dynamic particle swarm optimization algorithm based on random topology. *The Scientific World Journal*, *2013*.

Nickabadi, A., Ebadzadeh, M. M., & Safabakhsh, R. (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied soft computing*, *11*(4), 3658–3670.

Ozcan, E., & Mohan, C. K. (1999). Particle swarm optimization: Surfing the waves. In *Proceedings of the 1999 congress on evolutionary computation-cec99 (cat. no. 99th8406)* (Vol. 3, pp. 1939–1944).

Parrott, D., & Li, X. (2006). Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, *10*(4), 440–458.

Parsopoulos, K. E., & Vrahatis, M. N. (2002). Recent approaches to global optimization problems through particle swarm optimization. *Natural computing*, *1*, 235–306.

Parsopoulos, K. E., & Vrahatis, M. N. (2004). On the computation of all global minimizers through particle swarm optimization. *IEEE Transactions on evolutionary computation*, *8*(3), 211–224.

Ping Tian, D. (2013). A review of convergence analysis of particle swarm optimization. *International Journal of Grid and Distributed Computing*, *6*(6), 117–128.

Piotrowski, A. P., Napiorkowski, J. J., & Piotrowska, A. E. (2020). Population size in particle swarm optimization. *Swarm and Evolutionary Computation*, *58*, 100718.

Poli, R. (2008). Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications*, *2008*, 1–10.

Poli, R. (2009). Mean and variance of the sampling distribution of particle swarm optimizers during stagnation. *IEEE Transactions on Evolutionary Computation*, *13*(4), 712–721.

Poli, R., Bratton, D., Blackwell, T., & Kennedy, J. (2007). Theoretical derivation, analysis and empirical evaluation of a simpler particle swarm optimiser. In *2007 ieee congress on evolutionary computation* (pp. 1955–1962).

Poli, R., & Langdon, W. B. (2007). Markov chain models of bare-bones particle swarm optimizers. In *Proceedings of the 9th annual conference on genetic and evolutionary computation* (pp. 142–149).

Rao, S. S. (2019). *Engineering optimization: theory and practice*. John Wiley & Sons.

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

Ren, Z.-H., Wang, J., & Gao, Y.-L. (2011). The global convergence analysis of particle swarm optimization algorithm based on markov chain. *Control Theory & Applications*, *28*(4), 462–466.

Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on computer graphics and interactive techniques* (pp. 25–34).

Roy, R., & Ghoshal, S. (2008). A novel crazy swarm optimized economic load dispatch for various types of cost functions. *International Journal of Electrical Power & Energy Systems*, *30*(4), 242–253.

Sadoghi Yazdi, J., Kalantary, F., & Sadoghi Yazdi, H. (2012). Calibration of soil model parameters using particle swarm optimization. *International Journal of Geomechanics*, *12*(3), 229–238.

Sahu, A., Panigrahi, S. K., & Pattnaik, S. (2012). Fast convergence particle swarm optimization for functions optimization. *Procedia Technology*, *4*, 319–324.

Salomon, R. (1996). Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions. a survey of some theoretical and practical aspects of genetic algorithms. *BioSystems*, *39*(3), 263–278.

Schmitt. (2015). *Convergence analysis for particle swarm optimization*. FAU University Press.

Schmitt, & Wanka, R. (2013). Particle swarm optimization almost surely finds local optima. In *Proceedings of the 15th annual conference on genetic and evolutionary computation* (pp. 1629–1636).

Shakya, S., & Santana, R. (2012). *Markov networks in evolutionary computation* (Vol. 14). Springer.

Shi, Y., & Eberhart, R. (1998a). A modified particle swarm optimizer. In *1998 ieee international conference on evolutionary computation proceedings. ieee world congress on computational intelligence (cat. no. 98th8360)* (pp. 69–73).

Shi, Y., & Eberhart, R. C. (1998b). Parameter selection in particle swarm optimization. In *Evolutionary programming vii: 7th international conference, ep98 san diego, california, usa, march 25–27, 1998 proceedings 7* (pp. 591–600).

Shi, Y., & Eberhart, R. C. (1999). Empirical study of particle swarm optimization. In *Proceedings of the 1999 congress on evolutionary computation-cec99 (cat. no. 99th8406)* (Vol. 3, pp. 1945–1950).

Solis, F. J., & Wets, R. J.-B. (1981). Minimization by random search techniques. *Mathematics of operations research*, *6*(1), 19–30.

Spears, W. M., Green, D. T., & Spears, D. F. (2012). Biases in particle swarm optimization. In *Innovations and developments of swarm intelligence applications* (pp. 20–43). IGI Global.

Sudholt, D., & Witt, C. (2010). Runtime analysis of a binary particle swarm optimizer. *Theoretical Computer Science*, *411*(21), 2084–2100.

Sun, J., Feng, B., & Xu, W. (2004). Particle swarm optimization with particles having quantum behavior. In *Proceedings of the 2004 congress on evolutionary computation (ieee cat. no. 04th8753)* (Vol. 1, pp. 325–331).

Sun, J., Wu, X., Palade, V., Fang, W., Lai, C.-H., & Xu, W. (2012). Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Information Sciences*, *193*, 81–103.

Tang, Y., Wang, Z., & Fang, J.-a. (2011). Feedback learning particle swarm optimization. *Applied Soft Computing*, *11*(8), 4713–4725.

Tilahun, S. L. (2022). Swarm intelligence for a single source product distribution. *International Journal of Advanced Intelligence Paradigms*, *21*(1-2), 87–99.

Trelea, I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information processing letters*, *85*(6), 317–325.

Van Den Bergh, F. (2001). An analysis of particle swarm optimizers (pso). *Pretoria, University of Pretoria*, 78–85.

Van den Bergh, F., & Engelbrecht, A. P. (2002). A new locally convergent particle swarm optimiser. In *Ieee international conference on systems, man and cybernetics* (Vol. 3, pp. 6–pp).

Van den Bergh, F., & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization. *IEEE transactions on evolutionary computation*, *8*(3), 225–239.

Van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information sciences*, *176*(8), 937–971.

Van den Bergh, F., & Engelbrecht, A. P. (2010). A convergence proof for the particle swarm optimiser. *Fundamenta Informaticae*, *105*(4), 341–374.

Wan, S. (2013). Entropy-based particle swarm optimization with clustering analysis on landslide susceptibility mapping. *Environmental earth sciences*, *68*(5), 1349–1366.

Wang. (2013). Diversity enhanced particle swarm optimization with neighborhood search. *Information Sciences*, *223*, 119–135.

Wang. (2018). Particle swarm optimization algorithm: an overview. *Soft computing*, *22*, 387–408.

Wang, H., Moon, I., Yang, S., & Wang, D. (2012). A memetic particle swarm optimization algorithm for multimodal optimization problems. *Information Sciences*, *197*, 38–52.

Wilke, D. N., Kok, S., & Groenwold, A. A. (2007). Comparison of linear and classical velocity update rules in particle swarm optimization: Notes on scale and frame invariance. *International journal for numerical methods in engineering*, *70*(8), 985–1008.

Wilke, D. N., et al. (2007). *Analysis of the particle swarm optimization algorithm* (Unpublished doctoral dissertation). University of Pretoria.

Witt, C. (2009). Why standard particle swarm optimisers elude a theoretical runtime analysis. In *Proceedings of the tenth acm sigevo workshop on foundations of genetic algorithms* (pp. 13–20).

Xiao, X., & Yin, H. (2014). Moment convergence rates in the law of logarithm for moving average process under dependence. *Stochastics An International Journal of Probability and Stochastic Processes*, *86*(1), 1–15.

Xinchao, Z. (2010). A perturbed particle swarm algorithm for numerical optimization. *Applied Soft Computing*, *10*(1), 119–124.

Xu, G., & Yu, G. (2018). Reprint of: On convergence analysis of particle swarm optimization algorithm. *Journal of Computational and Applied Mathematics*, *340*, 709–717.

Yang, X.-S. (2011). Metaheuristic optimization: algorithm analysis and open problems. In *International symposium on experimental algorithms* (pp. 21–32).

Yuan, Q., & Yin, G. o. (2014). Analyzing convergence and rates of convergence of particle swarm optimization algorithms using stochastic approximation methods. *IEEE Transactions on Automatic Control*, *60*(7), 1760–1773.

Zhan, Z.-H., Zhang, J., & Liu, O. (2009). Orthogonal learning particle swarm optimization. In *Proceedings of the 11th annual conference on genetic and evolutionary computation* (pp. 1763–1764).

Zhang, Y., Gallipoli, D., & Augarde, C. (2013). Parameter identification for elasto-plastic modelling of unsaturated soils from pressuremeter tests by parallel modified particle swarm optimization. *Computers and Geotechnics*, *48*, 293–303.

Zheng, Y.-l., Ma, L.-h., Zhang, L.-y., & Qian, J.-x. (2003). Empirical study of particle swarm optimizer with an increasing inertia weight. In *The 2003 congress on evolutionary computation, 2003. cec'03.* (Vol. 1, pp. 221–226).

Zheng, Y.-L., Ma, L.-H., Zhang, L.-Y., & Qian, J.-X. (2003). On the convergence analysis and parameter selection in particle swarm optimization. In *Proceedings of the 2003 international conference on machine learning and cybernetics (ieee cat. no. 03ex693)* (Vol. 3, pp. 1802–1807).

RESEARCH WORK

Research articles

1. **Dereje Tarekegn Nigatu, Tekle Gemechu Dinka and Surafel Lulseged Tilahun** .
   Convergence analysis of particle swarm optimization algorithms for different constriction factors., Front.Appl.Math.Stat. sec.Optimization,volume 10-2024, doi.103389/fams 20241304268.

2. **Dereje Tarekegn Nigatu, Surafel Lulseged Tilahun and Tekle Gemechu Dinka**.
   " Convergence Analysis of Particle Swarm Optimization.", 20 August 2023, International Journal of Swarm Intelligence Research 14(1), DOI:10.4018/IJSIR.328092.

3. **Dereje Tarekegn Nigatu, Surafel Lulseged Tilahun and Tekle Gemechu Dinka** .
   A convergent particle swarm optimization method with repulsive functional constraints for solving unimodal and multimodal problems., Springer Journals, Computational Optimization and Applications (COAP). Under review