Particle Swarm Optimisation in Dynamically Changing Environments - An Empirical Study

by

Julien G.O.L. Duhain

Submitted in partial fulfillment of the requirements for the degree
Master of Science (Computer Science)
in the Faculty of Engineering, Built Environment and Information Technology
University of Pretoria, Pretoria

August 2011

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Publication data:

Julien Georges Omer Louis Duhain. Particle Swarm Optimisation in Dynamically Changing Environments - An Empirical Study. Master's dissertation, University of Pretoria, Department of Computer Science, Pretoria, South Africa, August 2011.

Electronic, hyperlinked versions of this dissertation are available online, as Adobe PDF files, at:

http://cirg.cs.up.ac.za/

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Particle Swarm Optimisation in Dynamically Changing Environments - An Empirical Study

by

Julien Georges Omer Louis Duhain
E-mail: s22160282@tuks.co.za

Abstract

Real-world optimisation problems often are of a dynamic nature. Recently, much research has been done to apply particle swarm optimisation (PSO) to dynamic environments (DE). However, these research efforts generally focused on optimising one variation of the PSO algorithm for one type of DE. The aim of this work is to develop a more comprehensive view of PSO for DEs. This thesis studies different schemes of characterising and taxonomising DEs, performance measures used to quantify the performance of optimisation algorithms applied to DEs, various adaptations of PSO to apply PSO to DEs, and the effectiveness of these approaches on different DE types.

The standard PSO algorithm has shown limitations when applied to DEs. To overcome these limitations, the standard PSO can be modified using personal best re-evaluation, change detection and response, diversity maintenance, or swarm sub-division and parallel tracking of optima. To investigate the strengths and weaknesses of these approaches, a representative sample of algorithms, namely, the standard PSO, re-evaluating PSO, reinitialising PSO, atomic PSO (APSO), quantum swarm optimisation (QSO), multi-swarm, and self-adapting multi-swarm (SAMS), are empirically analysed. These algorithms are analysed on a range of DE test cases, and their ability to detect and track optima are evaluated using performance measures designed for DEs. The experiments show that QSO, multi-swarm and reinitialising PSO provide the best results. However, the most effective approach to use depends on the dimensionality, modality and type of the DEs, as well as on the objective of the algorithm. A number of observations are also made regarding the behaviour of the swarms, and the influence of certain control parameters of the algorithms evaluated.

Keywords: Particle Swarm Optimisation, Dynamically Changing Environment, Computational Intelligence, Re-evaluating PSO, Reinitialising PSO, Charged PSO, Atomic PSO, Quantum Swarm Optimisation, Multi-swarm, Self-adapting Multi-swarm.

Supervisors : Prof A.P. Engelbrecht
Department : Department of Computer Science
Degree : Master of Science

"The grand aim of all science is to cover the greatest number of empirical facts by logical deduction from the smallest number of hypotheses or axioms."

– Albert Einstein

"Sometimes a scream is better than a thesis."

– Ralph Waldo Emerson

Acknowledgements

I would like to express my sincere thanks to the following people for their assistance during the production of this work:

- Prof Andries P. Engelbrecht for his mentorship and invaluable guidance during the progression of work completed for this thesis.
- Ms Anna Rakitianskaia and the late Mr Oluşegun Olorunda for their valuable input in several parts of this work.
- The members of CIRG who helped with their opinions and technical knowledge.
- Mr Pierre Duhain and Mrs Françoise Duhain for their support.
- The financial assistance of the National Research Foundation towards this research is hereby acknowledged. Opinions expressed in this thesis and conclusions arrived at, are those of the author and not necessarily to be attributed to the National Research Foundation.

Contents

List of Figures

List of Graphs

List of Algorithms

List of Tables

Chapter 1

Introduction

> *"Take the first step, and your mind will mobilize all its forces to your aid. But the first essential is that you begin. Once the battle is startled, all that is within and without you will come to your assistance."*
>
> – Robert Collier

Computational intelligence (CI) researchers have shown an increasing interest into solving optimisation problems where the objective function changes over time. This thesis refers to these problems as dynamic optimisation problems, or dynamic environments (DE). *Evolutionary algorithms* (EA) and *particle swarm optimisation* (PSO) have been applied to dynamic problems in recent years. This work focuses on PSO, a technique that is known to perform well on numerous static optimisation problems [17, 18, 19, 30, 39, 42, 92] but shows limitations when applied to a changing objective function [29].

A number of PSO algorithms have been developed for dynamically changing environments, and there exist several types of DEs. The main objective of this work is to conduct an empirical analysis of the various adaptations of PSO to solve dynamic optimisation problems and to evaluate the effectiveness of these approaches for different types of DEs. In reaching this goal, existing classifications of DEs are analysed and an alternative characterisation scheme is proposed, the PSO algorithms designed for DEs are reviewed, and after critically examining existing performance measures, performance measures specialised for DEs are selected and designed.

Section 1.1 presents an overview of the thesis, section 1.2 and 1.3 describe respectively the objectives and the contributions of this work, and section 1.4 outlines the various chapters.

1.1 Overview

Many efficient optimisation algorithms have been developed to solve complex optimisation problems. This is specifically the case for *computational intelligence* (CI) research fields such as EAs and swarm based algorithms. However, most of these algorithms were developed for solving static optimisation problems where the objective function, and therefore the search landscape, does not change over time.

Many real-world optimisation problems are, however, dynamic in that the optimal solution varies over time. Timetable optimisation, routing in a telecommunication network, and traffic control optimisation, are examples of such problems. Recently, algorithms have been developed to solve dynamic optimisation problems. These include EAs [15], PSO [29], and *ant colony optimisation* (ACO) [37]. These algorithms have shown various levels of success for specific DE types.

The focus of this thesis is on PSO algorithms and their applicability to solve dynamic optimisation problems. In a static environment, an optimisation algorithm must only attempt to find the best solution as fast as possible. However, when the environment is dynamic, the algorithm must not only find the optimal solution but must also *keep track* of the optimum as it moves through the search space as well as *detect* new optimal solutions as they appear. The original PSO algorithm struggles to optimise dynamic functions. To overcome the standard PSO's limitations, several approaches have been developed and variations of the PSO algorithm have been designed [6, 8, 9, 10, 11, 12, 27, 32, 38, 52, 54, 57, 59, 70, 71].

However, these research efforts generally focused on optimising one particular algorithm for one particular type of DE. Little effort has been made so far to develop a comprehensive view of the variations of PSO specialised for DEs and their applicability to solve different types of dynamic optimisation problems. The modified PSO algorithms make use of several approaches to adapt the standard PSO for DEs. Because these approaches were developed to optimise diverse dynamic problems, they should show a different level of efficiency depending on the type of DE the algorithm is applied to. Unlike previous research, this work does not aim at the optimisation of a specific algorithm for a specific problem but attempts to provide a better understanding of the effectiveness of the various approaches with regard to the various types of problems. This thesis is concerned with the definition and characterisation of dynamic optimisation problems, the measuring of the performance of PSO in DEs, the various approaches that can be considered for applying particle swarm algorithms to DEs, the effectiveness of these approaches, and the swarm behaviours that are observed for the different types of DEs.

1.2 Objectives

The main objective of this work is to provide an empirical analysis and comparison of a panel of PSO algorithms developed to solve dynamic optimisation problems. The purpose of this empirical study is to obtain a better understanding of dynamically changing environments and of the strengths and weaknesses that the various PSO algorithms demonstrate when applied to a variety of DEs. In reaching this goal, the following sub-objectives are identified:

- To study the existing classifications of DEs and to develop a characterisation system that can be used to define the main types of DEs.
- To identify the limitations that the standard PSO algorithm exhibits when applied to dynamic optimisation problems.
- To investigate how to optimise the standard PSO algorithm for maximum performance in DEs.
- To survey the various approaches taken to modify the PSO algorithms to overcome the PSO's limitation.
- To select, describe and implement a representative sample of PSO algorithms developed for DEs.
- To determine what qualities an algorithm should possess to perform well in DEs.
- To examine existing performance measures for swarm algorithms optimising dynamic functions.
- To select the most appropriate performance measures and design new performance measures if necessary.
- To define various test cases of DEs to efficiently assess the modified PSO algorithms.
- To define an experimental procedure that can highlight the strengths and weaknesses of the algorithms.
- To analyse the behaviours and performance that the standard PSO algorithm exhibits when applied to various DEs.
- To analyse the behaviours and performance that the PSO algorithms exhibit when applied to various DEs.
- To identify which algorithm is better suited for which type of DE based on the experiments conducted.

1.3 Contribution

The novel contributions of this work include the following:

- A new characterisation system for DEs is proposed.
- New performance measures are proposed to evaluate the various qualities of PSO algorithms applied to DEs.
- Implementations of the selected PSO algorithms, the selected performance measures, and the moving peak benchmark in CIlib [74].
- A descriptive list of approaches taken to adapt the PSO algorithms to dynamically changing environments is provided.
- An analysis of the behaviours and performances that a number of modified PSO algorithms demonstrate when applied to different kinds of DEs is given.
- A comparison between the dynamic algorithms tested which identifies the strength and weaknesses of each algorithm with regard to a variety of dynamically changing environments is provided.
- Suggestions regarding the most effective approach to take to optimise the various types of dynamic problems are given.

1.4 Thesis Outline

Chapter 2 begins the theory part of the thesis by discussing DEs and their characterisation. First the concepts related to dynamic function optimisation are explained. Then the existing categorisations for dynamically changing environments are investigated and a new system of characterisation is proposed. The base function that generates the test environments used to evaluate the algorithms, the *moving peaks benchmark* (MPB), is described in detail.

Chapter 3 describes PSO. The original PSO algorithm is explained and the various parameters influencing the behaviour and performance of the algorithm are discussed.

After identifying the limitations shown by the PSO algorithm when applied to DEs, **chapter 4** investigates several techniques researchers have used to overcome these limitations. A number of variations of the PSO algorithm designed to be effective in dynamically changing environments are also described in this chapter.

Chapter 5 covers the performance measures that are used in the experimental part of this work. Firstly, the limitations of the performance measures used to evaluate PSO algorithms applied to static environments are discussed. Secondly, the performance measures used in the past to assess evolutionary and swarm based algorithms applied to DEs are critically analysed. Thirdly, a guideline for measuring the ability of an algorithm to be efficient in solving dynamic optimisation problems is presented. Finally, a number of performance measures that can evaluate the various qualities of a dynamic algorithm are proposed.

Chapter 6 describes the experimental procedure used in this thesis. The test environments and evaluation methods are selected and described in detail. Additionally, values are assigned to these parameters that are common to all the algorithms evaluated during the experiments.

Chapter 7 provides an evaluation of the standard PSO and the re-evaluating PSO against the test environments to serve as benchmarks against which the modified PSO algorithms can be assessed. The behaviours that the swarms of these algorithms demonstrate on the various test environments are also analysed in this chapter.

Chapter 8 assesses the variations of the PSO algorithms that were described in chapter 4. Each algorithm is evaluated against the test environments described in chapter 6, using the stated experimental procedure and evaluation method. The algorithms are then compared with each other to identify the strengths and weaknesses of the various algorithms.

Chapter 9 brings this thesis to a conclusion with a summary of the major experiments and findings. This chapter also suggests a number of related research topics and experiments that can be conducted in the future.

The following appendices are also included:

Appendix A provides the pseudocode for creating a rotation matrix. This rotation matrix is used to generate rotating DEs.

Appendix B lists XML code snippets that were used to generate simulations within CIlib.

Appendix C lists the p-values obtained from Mann-Whitney U-test that determine if two sets of results are significantly different.

Appendix D summarises the acronyms used in this work.

Appendix E lists the symbols used in this work.

Part I

Background and Theory

"Know your enemy and know yourself and you can fight a hundred battles without disaster."

– Sun Tzu

Chapter 2

Dynamic Environments

"Nothing remains great without a capacity to change and to accommodate the conditions of a changing world."

– John Ashcroft

The first step towards solving a problem is to understand it. The opening chapter of this work is therefore dedicated to a theoretical overview of function optimisation and dynamically changing environments.

2.1 Introduction

The world is ever changing and what is optimal today can be sub-optimal tomorrow. Hence, solutions to real-world problems often need to be adapted as time goes by. But before looking for ways to solve dynamically changing problems, it is important to understand how dynamic real-world problems can be represented, defined, and characterised.

The first objective of this chapter is to describe how a static or dynamic problem can be represented as a mathematical function, and how, through function optimisation, a solution to this problem can be found. The second objective is to describe the nature and characteristics of the various types of DEs. Identification of the DEs' characteristics can lead to a categorisation of these environments. Proposing a characterisation system for DEs is the third objective of this chapter. Such categorisation opens the door to the design of pertinent test cases of dynamic optimisation problems. These test cases can allow a better evaluation of the algorithms designed to optimise dynamic problems. The fourth objective is to provide a description of an instance of a dynamic function that can be parameterised to represent a large range of DEs.

Section 2.2 defines the concepts of function optimisation and dynamic function optimisation. Section 2.3 then investigates existing ways of describing and classifying DEs. Section 2.4 introduces behavioural classes of DEs and proposes a more complete system of characterisation for these environments. Finally, section 2.5 gives a description of the MPB, a dynamic function that is used to define the test environments of chapter 6.

2.2 Function Optimisation

The simplified model of a computer-driven car can be used as an example to explain how a dynamic problem can be represented as a function.

The software in this example receives information from cameras located on the vehicle and can send commands to the wheel and pedals of the car. As the car moves, the software continuously attempts to keep the vehicle on the optimal position on the road, based on safety and fastest authorised speed. While driving the vehicle, the computer adjusts the direction of the wheels as well as the pressure on the accelerator or on the brakes depending on the shape of the road, the behaviour of other vehicles, road signs etc. At any given time, the objective for the software is to evaluate the situation and to select the best combination of values that should be assigned to the control parameters of the vehicle (direction and acceleration or deceleration). Each potential solution to this problem can therefore be represented as a point in a multi-dimensional space. In this space, one dimension corresponds to the acceleration/deceleration, i.e. pressure on the accelerator or brakes, and another dimension represents the angle of the wheels. The position of the point gives the combination of values characterising the potential solution to the problem.

To represent how optimal the solution is, a function over all dimensions is used. Because the visible road changes shape as the car moves forward, and because other vehicles use the same road, the formula (function) that is used to evaluate the quality of the position of the car changes over time. The problem can therefore be represented as an unknown changing mathematical function. Trying to find the optimum of this dynamic function is the objective of the software, hence *dynamic function optimisation*.

This section introduces the concepts necessary to understand static and dynamic function optimisation. Section 2.2.1 defines static function optimisation, the types of static optimisation problems are described in section 2.2.2, section 2.2.3 presents the types of solution to optimisation problems, and finally section 2.2.4 defines dynamic optimisation problems.

2.2.1 Definition of Static Function Optimisation

In mathematics and computer science, the problem of finding the best solution from a set of feasible solutions is called an optimisation problem. That is, a problem whose objective is to minimise or maximise a function by systematically choosing the values of variables within an allowed set. Formally, a maximisation problem is defined as

$$\begin{aligned} &\text{maximise} \\ &\qquad f(\mathbf{x}), \quad \mathbf{x} = (x_1, \ldots, x_{n_x}) \\ &\text{subject to} \\ &\qquad x_j \in \mathcal{F} \end{aligned} \tag{2.1}$$

The function f is the *objective function* or *base function*. The domain, $\mathcal{S}$, of f is referred to as the *search space*. The elements of $\mathcal{S}$ are the *candidate solutions* to the optimisation problem. The search space is limited by *domain constraints*. $\mathcal{S}$ therefore consists of the union of the *feasible space* $\mathcal{F}$ and the *infeasible space* $\mathcal{I}$ which is the set of *infeasible solutions*. Infeasible solutions are those candidate solutions that are excluded from the search space by constraints (domain constraints or others) stated in the problem's definition. $\mathcal{F}$ is the set of *feasible solutions*, that is the set of valid solutions according to the problem's definition. A feasible solution that optimises the objective function is an *optimal solution*.

2.2.2 Types of Static Optimisation Problems

A static optimisation problem is an optimisation problem where the function to optimise does not change during the optimisation process. A number of characteristics can be used to classify static optimisation problems:

- The **number of variables** influencing the objective function: *Univariate* problems have only one variable to be optimised and *multivariate* problems have more than one variable to be optimised.
- The **type of variables**: Optimisation problems can be classified according to the type of values the variables of the problem can take. The variables of a *continuous* problem have continuous, floating-point values, the variables of an *integer* problem have integer values, the variables of a *discrete* problem have discrete (i.e. non-continuous) values, the variables of a *mixed-integer* problem can have both continuous or integer values, and solutions to a *combinatorial* optimisation problem are permutations of integer-valued variables.
- The **degree of nonlinearity of the objective function**: Depending on the linear or quadratic nature of the objective function, problems can be *linear*, *quadratic*, or *nonlinear* (when other nonlinear functions are used).

- The **constraints used**: In addition to the domain's constraint, some problems define an extra set of constraints which exclude certain candidate solutions. Such problems are called *constrained problems*. Problems that use only boundary constraints are generally referred to as *unconstrained problems*.
- The **number of optima**: A problem is *unimodal* if there is only one optimum or *multimodal* otherwise.
- The **number of optimisation objectives**: A problem that specifies only one objective that needs to be optimised is called a *uni-objective* or *single-objective* problem. If more than one sub-objective is specified, the problem is *multi-objective*.

This thesis focuses on multivariate, continuous, nonlinear, uni-objective, unconstrained problems. In this text, the term optimisation problem refers to this type of problems unless otherwise specified. Both unimodal and multimodal problems are used in the experiments.

2.2.3 Types of Solutions to Optimisation Problems

Solutions to an optimisation problem can be local or global optima. These types are formally defined as follows, considering a maximisation problem, and illustrated in figure 2.1:

- The solution $\mathbf{x}^* \in \mathcal{F}$ is the *global maximum* of the objective function, f, if

$$f(\mathbf{x}^*) > f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{F} \tag{2.2}$$

- The solution $\mathbf{x}^* \in \mathcal{F}$ is a *strong local maximum* of the objective function, f, if

$$f(\mathbf{x}^*_{\mathcal{N}}) > f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{N} \tag{2.3}$$

  where $\mathcal{N} \subseteq \mathcal{F}$ is a set of feasible points in the neighbourhood of $\mathbf{x}^*_{\mathcal{N}}$.

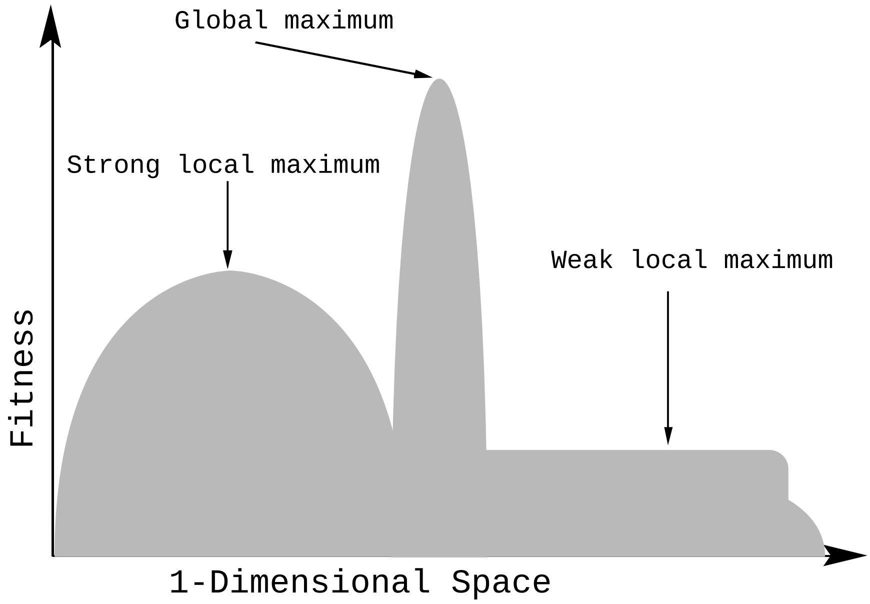


Figure 2.1: Types of solutions to optimisation problems

- The solution $\mathbf{x}^* \in \mathcal{F}$ is a *weak local maximum* of the objective function, f, if

$$f(\mathbf{x}^*_{\mathcal{N}}) \geq f(\mathbf{x}), \forall \mathbf{x} \in \mathcal{N} \tag{2.4}$$

  where $\mathcal{N} \subseteq \mathcal{F}$ is a set of feasible points in the neighbourhood of $\mathbf{x}^*_{\mathcal{N}}$.

All problems used in this work are maximisation problems. The words optimum and maximum are therefore interchangeable in this text.

2.2.4 Definition of Dynamic Optimisation Problems

A dynamic optimisation problem is an optimisation problem which has an objective function that changes over time, formally defined as [29]:

$$
\begin{aligned}
&\text{maximise} \\
&\qquad f(\mathbf{x}, w(t)), \quad \mathbf{x} = (x_1, \ldots, x_{n_x}), w(t) = (w_1(t), \ldots, w_{n_w}) \\
&\text{subject to} \\
&\qquad g_m(\mathbf{x}) \leq 0, \quad m = 1, \ldots, n_g \\
&\qquad h_m(\mathbf{x}) = 0, \quad m = n_g + 1, \ldots, n_g + n_h \\
&\qquad x_j \in dom(x_j)
\end{aligned}
\tag{2.5}
$$

where g_m represents the m^{th} inequality constraint, n_g is the number of inequality constraints, h_m represents the m^{th} equality constraint, n_h is the number of equality constraints, $dom(x_j)$ is the domain of x in dimension j, and $w(t)$ is a vector of time-dependent objective function control parameters. The objective is to find

$$\mathbf{x}^*(t) = \mathbf{x} \quad | \quad f(\mathbf{x}) = \underset{\mathbf{x}}{max}\{f(\mathbf{x}, w(t))\} \tag{2.6}$$

where $\mathbf{x}^*(t)$ is the optimum found at time step t.

DEs and their characteristics are discussed in the following section.

2.3 Existing Classifications and Characterisations of Dynamic Environments

Dynamic optimisation problems can be categorised into different classes of problems based on a number of characteristics. It is important to perform a categorisation of DEs for several reasons. Firstly, a complete classification system defines a wide range of dynamic problems and evaluating an optimisation algorithm against these various kinds of DEs allows to better identify the algorithm's strengths and weaknesses. Secondly, after identifying in which category a problem fits, the algorithms known to perform well for environments of that category can already be expected to perform well for this problem. Finally, algorithms can demonstrate a specific behaviour for a specific type of environment. For instance, an algorithm can consistently show certain limitations or achieve a certain performance level when applied to a particular environment type. It is therefore possible to gather information about the type of an unknown DE by studying the behaviour of a known algorithm in that environment.

Section 2.3.1 describes the characteristics specific to dynamic problems and section 2.3.2 to 2.3.5 review the various existing classification systems for DEs.

2.3.1 Characteristics of Dynamic Environments

In addition to the characteristics presented in section 2.2.2, DEs can be characterised by the way in which the base function is modified over time. The characteristics of DEs are listed below:

- The **direction of change** defines whether a change can modify the location of an optimum, its value, or both.
- The presence or absence of a **pattern in the changes** to the base function can be used to characterise a DE. If a pattern is present in the changes, the environment can also be categorised based on the shape of this pattern, i.e. the **trajectory of the optima**. If the pattern repeats itself over time, the number of environmental states between two consecutive encounters of the same state defines a **cycle length** [15] which is another characteristic of DEs.
- The **homogeneity of movement** describes the coherence of the changes across the search space. In a *homogeneous* environment every optimum undergoes the same transformation when a change occurs and all peaks exhibit the same behaviour at any given time. In a *heterogeneous* environment, the optima undergo different transformations when a change occurs and different peaks can have completely different behaviours.
- The frequency at which an environment changes is referred to as the **temporal severity**. Changes can occur *continuously*, *periodically*, or at *random* time intervals. Regular periodic modifications can be described in terms of an *update*

frequency [1], which can be measured by the number of iterations, the number of function evaluations, or even the time (in the case of a real-world application) between successive changes. In multimodal heterogeneous environments, the temporal severity should be evaluated by considering all changes regardless of what portion of the search space is affected. The lowest possible temporal severity occurs when the environment changes only once. The highest temporal severity occurs when modifications to the landscape take place at each time step causing the environment to continuously change. A low temporal severity allows the environment to remain static for a period of time, and gives to the optimisation algorithm the opportunity to find a good solution before a change occurs. A higher temporal severity forces the optimisation algorithm to frequently adjust the solution(s) found in response to the changes that take place.

- **Spatial severity** refers to the magnitude of the modification experienced by the location or the value of an optimum when a change occurs. The spatial severity is *regular* when the intensity of the modification to the optimum is always the same every time a change occurs. Otherwise, the spatial severity can *follow a pattern* if the intensity of the modifications is predictable change after change. The severity of the changes can also vary *randomly* within a certain range. In those environments where the severity of the alteration is different each time (non-regular spatial severity), the spatial severity of an environment can be measured in terms of the average spatial severity over the entire time period. In environments with multiple optima, the spatial severity of the environment can be evaluated by considering only the largest change in fitness or location. Alternatively, the spatial severity can be measured by following the changes to a single, randomly selected peak [16] or by taking an average of the changes experienced by all optima. Typically, the more severe the change, the harder it is for the optimisation algorithm to recover from the change. After a small change occurred, the new optimum can be found in the area surrounding the old optimum; it is not the case after a severe change takes place.

Branke *et al.* [16] have proposed a number of measures to help characterise DEs by quantifying the spatial severity of the changes and the homogeneity of the changes across the search space. These measures are listed below:

- **Change severity** is the normalised Euclidian distance between the current location of the optimum and the location of the optimum before the change divided by the maximal distance between any two points of the search space.
- **Estimated change severity** is the Euclidian distance between the best solution before and after a change.
- **Fitness correlation** measures the correlation coefficient of the fitnesses of a number of randomly selected position samples before and after a change.
- **LHC Fitness correlation** is the correlation of fitness obtained by performing *local hill climbing* (LHC) before a change and performing LHC again after a change. LHC is a mathematical optimisation technique [78]. The LHC algorithm starts with a random solution, and iteratively makes small changes to the solution, improving the solution during each iteration. When the algorithm cannot detect any improvement anymore, it terminates.
- **Fitness change correlation of similar points** measures the correlation with distance d by picking n random pairs of positions in the search space separated by distance d, and observing the correlation of fitness change between these n pairs. This measure evaluates how homogeneous the changes are throughout the search space.
- **Estimated value of last-stage local optima** compares the fitness obtained by performing LHC from the best position sample of one of the previous environmental states (referred to as stage) to the fitness obtained by LHC from a random sample.
- **Value of past optima** keeps the optima of the m previous stages and checks how much the solution found by LHC from a previous optimum improves compared to starting LHC from a random solution.

- **Estimated value of past optima** replaces the optimum in the previous measure by the point obtained from LHC on the best sample.

There are other aspects of a problem one could consider when describing a DE such as the necessity to change the representation of the problem after a change, the influence of the algorithm on the environment [15], the presence of noise in the data, or the fact that the system is notified of when a change occurs or not. However, these characteristics are more related to the interaction between the algorithm and the environment than to the environment itself.

2.3.2 Eberhart et al.'s Classification of Dynamic Environments

Three types of unconstrained, single objective, DEs were identified by Eberhart *et al.* [27, 40]. These types classify the different ways in which an optimum can be modified when the base function changes shape, i.e. the direction of change. The different categories are outlined below, and discussed in reference to equation (2.7):

$$f(\mathbf{x}, w(t)) = \sum_{j=1}^{n_x} (x_j - w_1(t))^2 + w_2(t) \tag{2.7}$$

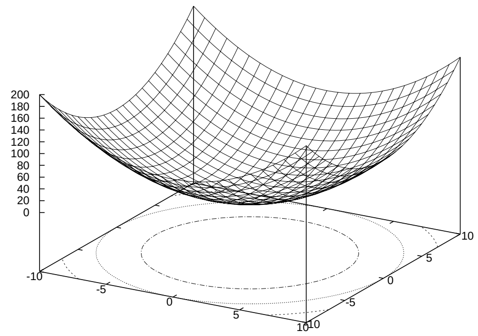
where n_x is the number of dimensions and the control parameters w_1 and w_2 can be used to change the shape of the function. According to Eberhart *et al.*, DEs can be classified as follows :

- **Type I** environments, where the position of the optimum changes but its value remains the same. Type I environments can be obtained by *transforming the coordinates* of the landscape components. That is, peaks, plateaus and hills that compose the landscape are translated to a different location. An example of a type I environment is illustrated in figure 2.2(b). The function from equation (2.7) behaves like a type I environment if $w_1 \neq 0$ and $w_2 = 0$. For this kind of environments, algorithms need only to be able to track the position of the optimum after finding it. This is because no new optimum appears elsewhere in the search space, neither can an existing local optimum become better than the global optimum.

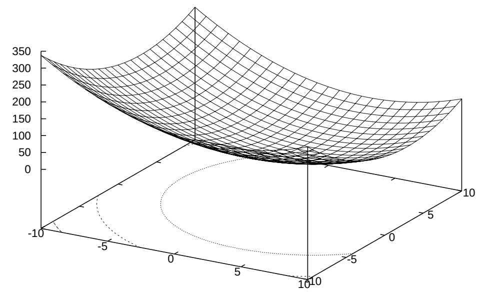
- **Type II** environments, where the position of the optimum stays unchanged but its value varies over time. Type II environments can be obtained through *fitness rescaling* of the landscape components [101]. In an environment where multiple peaks are present, the decrease of the global maximum's value or the growth of a local maximum may lead to a change of global optimum. Algorithms applied to this type of environments must therefore be able to detect the emergence of new optima. The function from equation (2.7) behaves like a type II environment if $w_1 = 0$ and $w_2 \neq 0$. An example of a type II environment is illustrated in figure 2.2(c).

- **Type III** environments, which are a combination of type I and II environments where both the location and the value of an optimum are subject to change. The function from equation (2.7) behaves like a type III environment if $w_1 \neq 0$ and $w_2 \neq 0$. An example of a type III environment is illustrated in figure 2.2(d).

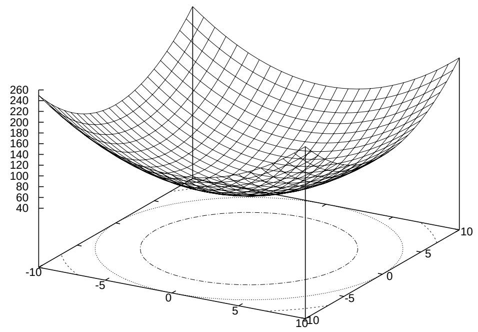
Unimodal problems and homogeneous multimodal problems can easily be characterised by this classification system because all optima in the search space behave in the same way. In the case of a heterogeneous multimodal problem, the different peaks can have different behaviours. If some peaks behave in a type I manner and others in a type II manner, or if any peak behaves in a type III manner, the environment is of type III because both the location and the value of the optimum can change. Eberhart *et al.*'s classification system provides an unambiguous way of dividing the DEs between type I, II and III but focuses solely on the direction of the changes the environment undergoes. The other characteristics listed in section 2.3.1 also influence the difficulty of the problems and a more complete classification system for DEs should incorporate the other characteristics to obtain a finer characterisation.



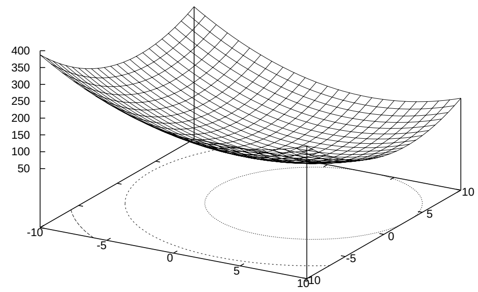
(a) Before change



(b) Type I after one change
$w_1 = 3$, $w_2 = 0$



(c) Type II after one change
$w_1 = 0$, $w_2 = 50$



(d) Type III after one change
$w_1 = 3$, $w_2 = 50$

Figure 2.2: Types of DEs according to the Eberhart *et al.* classification

2.3.3 Angeline's Classification of Dynamic Environments

The presence or absence of a pattern in the changes that occur in the environment is an alternative approach to classify DEs, as proposed by Angeline [1]. The trajectory of the optima defines the *dynamics* of an environment. According to Angeline's classification, problems are categorised into different classes of DEs based on the trajectories of the optima over a number of iterations [1]. Angeline's classification scheme identifies the following environment types [1, 2, 15]:

- Environments with **linear** peak trajectories where the optima always move in a straight line as illustrated in figure 2.3(a) for the moving parabola,

$$f(\mathbf{x}, t) = \sum_{j=1}^{n_x} (x_j + \triangle_j(t))^2 \tag{2.8}$$

where t denotes the index for the environmental state; t is increased by one every time the environment changes. Equation (2.8) has linear dynamics when $\triangle_j(t)$ is defined as

$$\triangle_j(t) = \begin{cases} 0 \quad \forall j \in 1, \ldots, n_x & \text{if } t = 0 \\ \triangle_j(t-1) + s_j S & \text{if } t > 0 \end{cases} \tag{2.9}$$

where S represents the spatial severity and s_j is a constant defined for each dimension j which gives the direction of the linear trajectory.

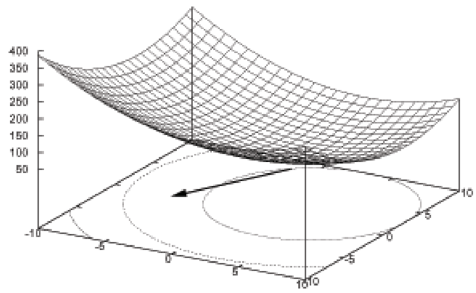
- Environments with **circular** trajectories where the optima move on the circumference of a circle with radius S and the base function is translated in a circular manner as illustrated in figure 2.3(b). The parabola defined in equation (2.8) has circular dynamics when

$$\triangle_j(t) = \begin{cases} 0 & \text{if } t = 0 \text{ and } j \text{ is odd} \\ S & \text{if } t = 0 \text{ and } j \text{ is even} \\ \triangle_j(t-1) + S \sin \frac{2\pi t}{T} & \text{if } t > 0 \text{ and } j \text{ is odd} \\ \triangle_j(t-1) + S \cos \frac{2\pi t}{T} & \text{if } t > 0 \text{ and } j \text{ is even} \end{cases} \tag{2.10}$$

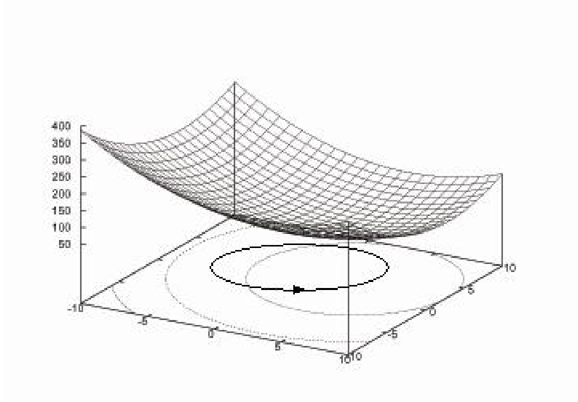
where $\frac{t}{T}$ is the fraction of the circumference that has been travelled through so far. If T remains constant through the simulation, the base function returns to its original position after T modifications. Spatial severity is therefore defined using S, t and T.

- Environments with **random** dynamics, where the base function is translated randomly as illustrated in figure 2.3(c). The function defined in equation (2.8) has random dynamics when noise is added to the offset $\triangle_j(t)$ as follows:

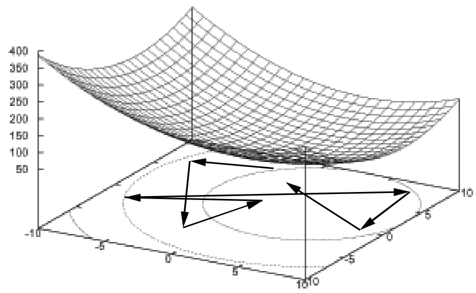
$$\triangle_j(t) = \begin{cases} 0 \quad \forall j \in 1, \ldots, n_x & \text{if } t = 0 \\ \triangle_j(t-1) + s_j S N(0,1) \end{cases} \tag{2.11}$$



(a) Linear trajectory



(b) Circular trajectory



(c) Random trajectory

Figure 2.3: Types of DEs according to the Angeline classification

where $N(0,1)$ is a Gaussian random variable with 0 mean and a deviation of 1, and s_j multiplied by S gives the spatial severity for dimension j.

Environments where the optimum *alternates* between several positions and where the landscape *periodically* returns to previous states [101] can be classified as circular environments. Even though the trajectory depicted by the optimum might not be an exact circle in these cases, alternating and periodic environments can be said to have circular dynamics since the optimum is expected to be found repetitively at specific locations. So, if S is replaced by s_j in equation (2.10) the environment's dynamics remain classified as circular.

More environment types could be added to this classification scheme to include non-random, non-circular trajectories that do not follow an exact straight line and are therefore non-linear. But these extra types of dynamics can be hard to detect and make the classification of environments more ambiguous. Any environment with non-linear and non-circular pattern in the changes can therefore be classified as an environment with random dynamics although the pattern in the change is not random per say.

Multimodal, heterogeneous environments should always be considered to have random dynamics since the entire landscape never shifts in a linear manner nor describes a circular trajectory as a whole. Although, the different peaks can be said to have linear or circular trajectories when considered individually.

Algorithms that use memory of the position of old optima or a hall of fame could potentially take advantage of the presence of a pattern by "expecting" an optimum in areas where the optimum has been found previously (in the case of circular trajectories). Predicting the next position of the optimum based on its past moves (linear or circular trajectories) might also be possible although a swarm based algorithm with such capacity has not been encountered during the writing of this thesis.

As with Eberhart *et al.*'s classification, Angeline's types do not overlap and clearly separate DEs in three distinct categories, although, Angeline's classification focuses on a single characteristic of the environment, namely the trajectory of the optimum.

This said, Angeline's and Eberhart *et al.*'s classification focus on different characteristics of the environment. When categorising a DE, it is therefore relevant to provide the environment's type according to Angeline *and* its type according to Eberhart *et al.* as each of these classifications offers information about a different feature of the changes the environment undergoes. However, neither Angeline's nor Eberhart *et al.*'s classification gives explicit information about the severity – either spatial or temporal – of the changes which allows for the classification of a DEs to be further refined.

2.3.4 Weicker's Classifications of Dynamic Environments

Weicker [15, 101, 102] proposed a classification framework for DEs where an environment is characterised based on

- the stationary or dynamic nature of the environment;
- the spatial severity of change that is either constant or variable;
- whether the environment is *periodic*, that is the environment returns to previous states, or not;
- whether the environment is *alternating*, that is the global optimum can be relocated to a different element (peak) in the environment, or not; and
- whether the environment is homogeneous or heterogeneous.

This characterisation provides more information about the environment than either Angeline's or Eberhart *et al.*'s classifications as Weicker takes into consideration the presence of a circular pattern in the changes, the constant or variable nature of the spatial severity, the homogeneity of movements, and the alternation of the global optimum which gives an indication on the direction of change. However, the alternating nature of the environment is redundant if the problem is known to be unimodal or homogeneous as these environments cannot be alternating. Also, in a type I or type III environment, the global optimum can be relocated while remaining on the same peak as the peaks themselves are moving. For these reasons, Eberhart *et al.*'s classification is more descriptive regarding the direction of change. Regarding the presence of a pattern in the change, Weicker only identifies circular (periodic) or random (non-periodic) trajectories while Angeline's classification also includes linear trajectories. Also, Weicker does not characterise the spatial or temporal severity of the changes.

2.3.5 De Jong's Classification of Dynamic Environments

De Jong [25] proposed a classification scheme for DEs where the categories are motivated by real-world problems with similar characteristics. Referring to environments as landscapes, De Jong classifies DEs as follows:

- Drifting landscapes, where the environment does not change much over time. The optima gradually move though the search space. The severity of change is low in these environments, but changes should occur frequently enough to allow a progressive change of the landscape.
- Landscapes that are undergoing significant morphological changes over time, where regions of high fitness emerge from previously uninteresting areas while the fitness of other regions decreases. The severity of the changes is higher in these environments than in drifting landscapes.
- Landscapes that exhibit cyclic patterns, where the landscape repeatedly visits a number of states.
- Abrupt and discontinuous problems, where the environment's landscape is fundamentally modified after a change, i.e. the severity of the changes is higher than in significantly changing landscapes.

The first, second and fourth categories De Jong described are distinguished by the spatial severity of the changes. Drifting landscapes have a low spatial severity, problems with significant morphological changes have a medium spatial severity, and abrupt and discontinuous problems have changes with high spatial severity. Unfortunately, the distinction between the three severity levels appears blurry. Furthermore, an environment can change significantly after a number of subsequent minor changes *or* after one single severe change, but this classification scheme does not differentiate between these two types of significantly changing environment. Also, as illustrated in figure 2.4, the temporal severity can only be inferred for the drifting landscapes category and it appears

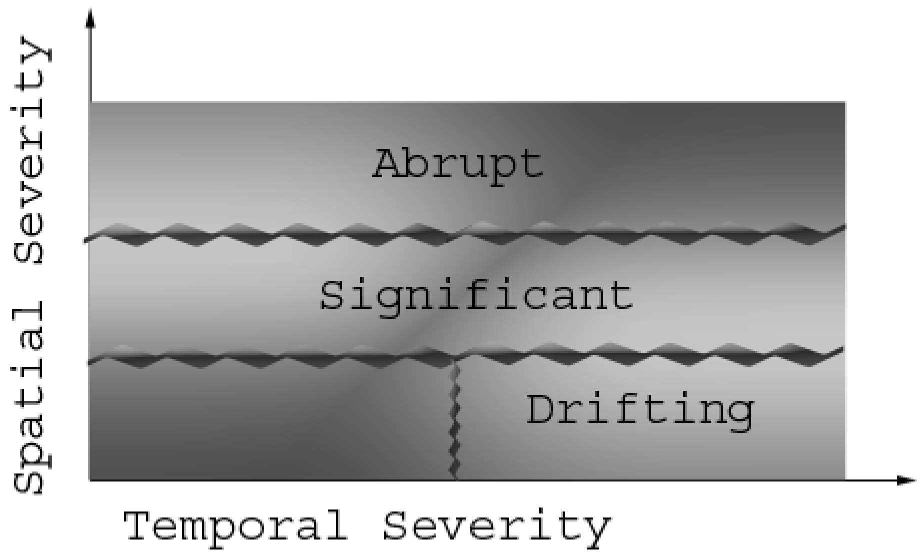


Figure 2.4: De Jong's categories could be refined based on the temporal severity.

that De Jong's classification could be further refined based on the temporal severity. It might therefore be pertinent to consider both spatial and temporal severity together when characterising the behaviour of DEs.

The category of problems with cyclic patterns is similar to Angeline's circular type and is based on the trajectory of the change instead of the spatial severity. This category and the other three categories of De Jong are therefore not mutually exclusive. For instance, a DE that would undergo significant morphological changes following a cyclic pattern could be classified simultaneously in the second and the third of De Jong's categories.

The fact that this scheme was inspired by real-world dynamic problems emphasises the importance of spatial severity in these problems. However, temporal severity is also an important aspect and the above analysis of De Jong's categories suggests that a better description of DEs could be obtained by considering spatial and temporal severity jointly.

Table 2.1: Summary of classification schemes for DEs

	Direction of change	Trajectory of change	Homogeneity of movement	Spatial and Temporal Severity
Eberhart *et al.*	Type I, type II or type III			
Angeline		Linear, circular or random		
Weicker	Alternating or non-alternating	Periodic or non-periodic	Homogeneous or heterogeneous	Constant or variable
De Jong		Cyclic or non-cyclic		Drifting, significant or abrupt
Behavioural Classes				Progressive, abrupt or chaotic

2.4 Proposal of a More Complete Classification System

As illustrated in table 2.1, the existing classification systems described in the previous section only focus on specific characteristics of DEs. Describing a DE using all classification systems would be cumbersome and redundant. However, a more accurate characterisation of DEs can be obtained by combining the existing classification systems that provide the most descriptive types of environment and do not overlap with each other. New classes of dynamic problems that consider the frequency and severity of the changes the environment undergoes can also be designed to refine the characterisation of these problems.

The purpose of this section is to propose a more complete classification system for DEs. Section 2.4.1 defines new classes of DEs based on the *behaviour* of the environment which, in this thesis, refers to the combination of the temporal and spatial severity. Section 2.4.2 presents a classification system that combines Eberhart *et al.* and Angeline's types to the new behavioural classes.

2.4.1 New Behavioural Classes

In addition to the direction and trajectory of the optima, the severity and the frequency of change are also important characteristics of a DE. As observed in section 2.3.5, it is relevant to consider both spatial and temporal severity together to characterise the different types of behaviours a DE can have. This thesis proposes a new set of environment classes based on the temporal and spatial severity of the alterations that the environment undergoes.

Figure 2.6 illustrates the categorisation proposed in this paper. Each class is described in more detail below:

- **Static** and **quasi-static environments**: If either the spatial or temporal severity is null, the environment's behaviour is **static** (non-dynamic). If modifications to the environment are irrelevant compared to the scale of the problem and do not affect the performance of the algorithm for the duration of the simulation, the environment can also be classified as static or **quasi-static**.
- **Progressively changing environments**: If the alterations are frequent but small, the environment changes progressively and smoothly. An algorithm can take advantage of this fact by keeping track of the locations of previous optima and focusing the search in the surroundings of these locations. Also, if a new optimum emerges in an environment with such behaviour, the algorithm has a certain amount of time to find this new optimum before it becomes the global optimum.

  Figure 2.5(a) shows the progression of the optimum's fitness in a progressively changing environment defined in the domain [-20,20] as

  $$f(\mathbf{x}, t) = 40 - \triangle(t) - \sum_{j=1}^{n_x} x_j \qquad (2.12)$$

  where n_x, the number of dimensions is set to two, t is incremented by one after every change and equation (2.12) is updated every iteration with $\triangle(t)$, defined as

  $$\triangle(t) = 6 + N(0, 1) \qquad (2.13)$$

where $N(0,1)$ is a Gaussian random variable, with 0 mean and a deviation of 1.

- **Abruptly changing environments**: If the changes are rare but severe, the environment stays static for a while, giving time for the algorithm to find a good solution, and then changes abruptly, relocating the optima to possibly distant locations and/or changing their fitness significantly.

  An example of an abruptly changing environment is obtained by updating equation (2.12) every 25 iterations in the domain [-20, 20] with

  $$\triangle(t) = 20 + N(0,40) \tag{2.14}$$

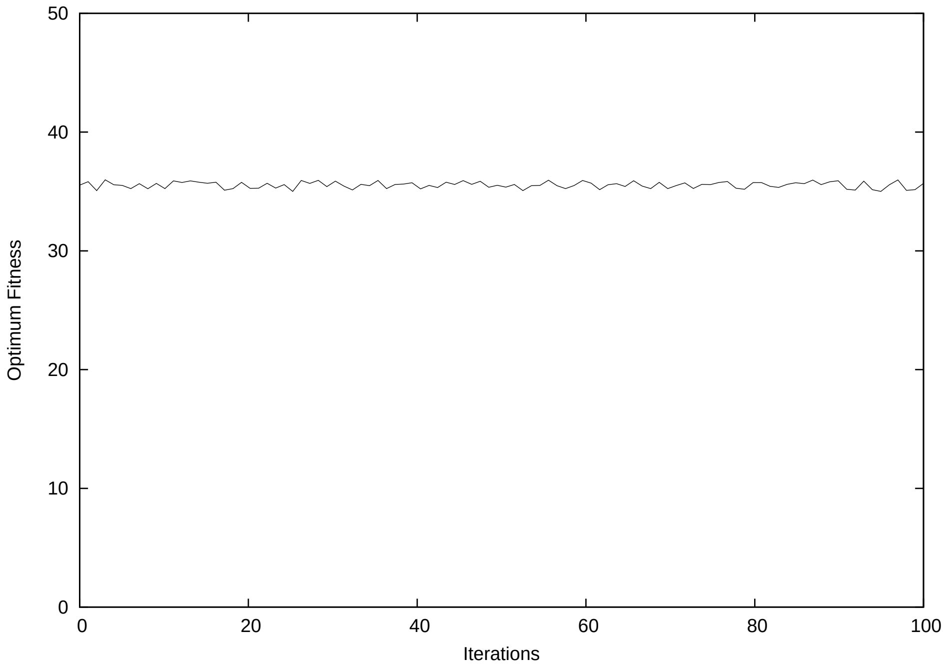
  Figure 2.5(b) illustrates the progression of the optimum's fitness in the above abruptly changing environment.

- **Chaotic environments**: If both the spatial and temporal severity are high, the behaviour of the environment is chaotic. In this context, "chaotic" does not refer to the randomness of the changes – which is described by Angeline's type – but to the chaotic nature of the behaviour, frequently and severely changing. In extreme cases where the optimum is randomly repositioned in a different area of the search space at every iteration, the optimisation process may prove to be extremely difficult. In a chaotic environment, an algorithm must be able to simultaneously adapt to frequent and severe changes.

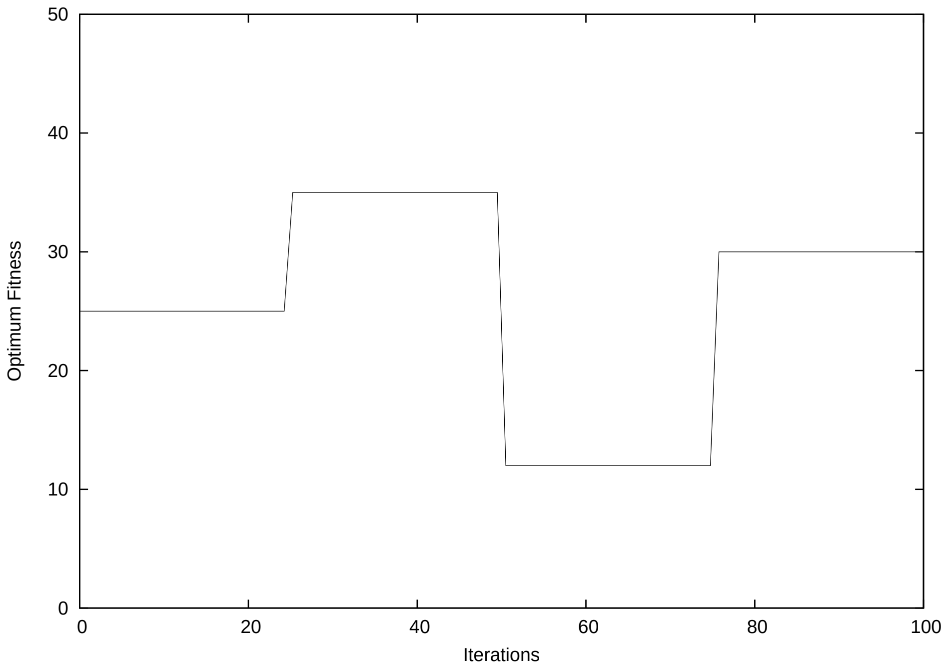
  An example of a chaotically changing environment is obtained by updating equation (2.12) every iteration in the domain [-20, 20] with

  $$\triangle(t) = 20 + N(0,40) \tag{2.15}$$

  Figure 2.5(c) illustrates the progression of the optimum's fitness in a chaotically changing environment.

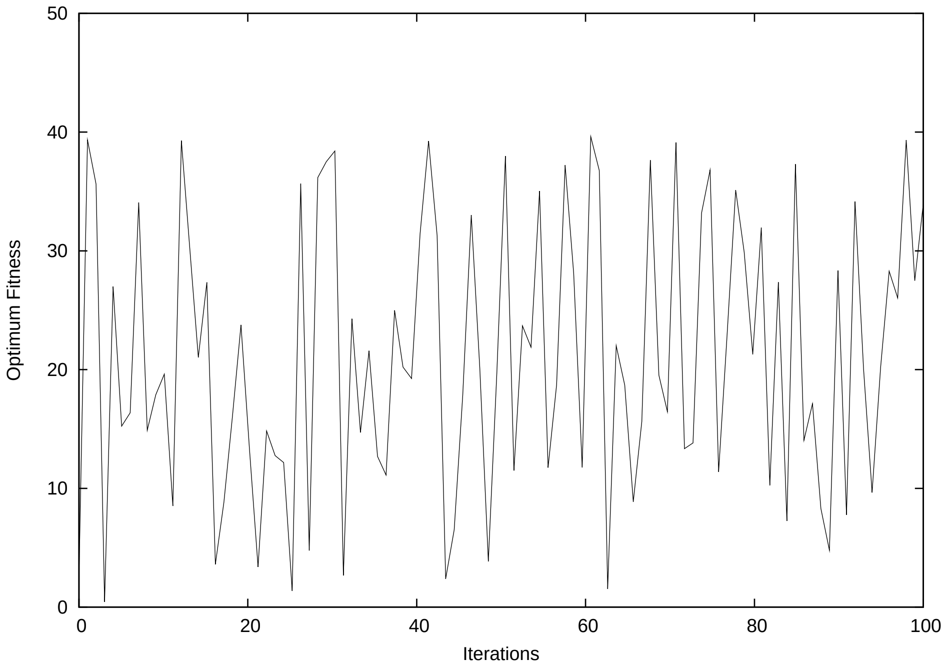


(a) Progressively changing environment



(b) Abruptly changing environment

Figure 2.5: Progression of the global optimum's fitness in DEs



(c) Chaotically changing environment

Figure 2.5: (continued)

The size of the search space should be taken into consideration when measuring the spatial severity since the perceived severity of a change is relative to the size of the environment. Similarly, the duration of the simulation should be considered when differentiating between frequently and infrequently changing environments. Unfortunately, there is no absolute value that can be used to unambiguously distinguish a severe change from a non-severe change, and there is also no obvious way to decide if an environment changing every 50 iterations, for instance, should be considered as frequently changing or not. A crisp classification of the temporal or spatial severity is therefore not possible, as it is problem dependent.

This said, the behaviours of specific problems can be compared in terms of the degree to which they belong to any of the classes. In [58], progressively, abruptly and chaotically changing environments have been used as experimental test cases. In the context of an experiment with a given dynamic problem test case whose spatial and temporal severity

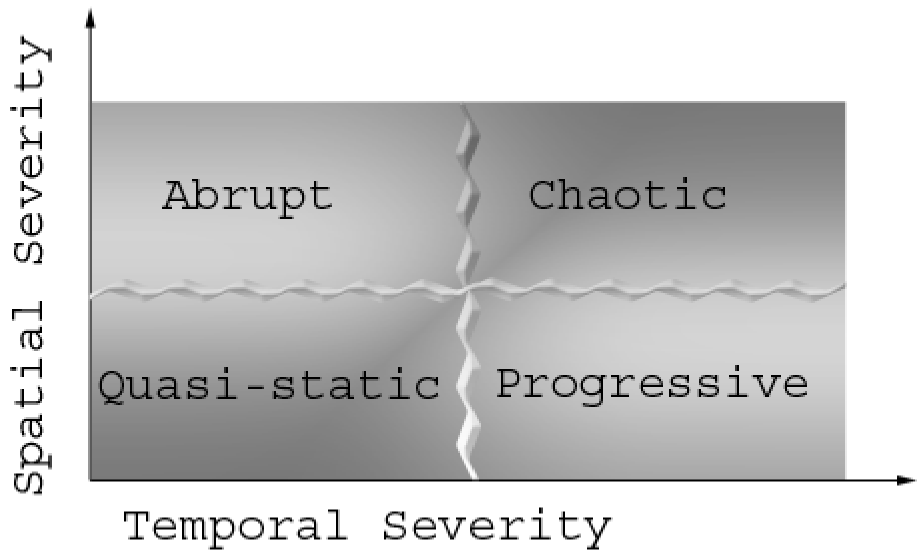


Figure 2.6: Behavioural classes for DEs

can be set, variations of the problem belonging to the different classes can be defined. Testing an algorithm on these different variations of the problem allows the analysis of the influence of the environment's behaviour on the performance level of the algorithm.

The behavioural classes provide more information about the environment compared to De Jong's categories as not only the spatial but also the temporal severity of the changes is described by the classes. These classes do not overlap with Eberhart *et al.* or Angeline's types. These three classification schemes can therefore be used simultaneously to provide information about the direction, trajectory, frequency, and severity of the changes in a given DE.

2.4.2 More Complete Classification System

As it has been shown, there are several approaches to characterise DEs. A truly complete classification system, if possible at all, would be complex and therefore hard to use. However, from the review of the existing classification systems examined in section 2.3, it appears that a more comprehensive characterisation of DEs can be proposed.

The categories of environment should be descriptive, and mutually exclusive. Eberhart *et al.*'s classification is the most descriptive of the direction of change, Angeline's is the most descriptive of the patterns in the changes and the behavioural classes proposed describe both the temporal and spatial severity of the changes. De Jong's cyclic class is replaced by Angeline's circular type and the behavioural classes propose an alternative way of classifying the spatial severity to De Jong's drifting, significant and abrupt classes. As discussed in section 2.3.4 Eberhart *et al.*'s and Angeline's types are more descriptive than Weicker's regarding respectively the direction and trajectory of change. By combining Eberhart *et al.* and Angeline's types of environments to the behavioural classes from section 2.4.1, a more complete classification system is obtained. For instance, an environment can be characterised as an "abruptly changing environment of type I with circular dynamics" or, for short, "abrupt, circular, type I". This characterisation defines 27 kinds of DEs based on the direction, the trajectory and the severity of the changes they undergo. The categories from Weicker's classification that are not redundant can be used, if relevant, to further describe the environment: the constant or variable nature of the severity can be mentioned, and for multimodal problems, the homogeneous or heterogeneous nature of the environment can also be stated. If the experiments are to be reproducible, the frequency of change and the spatial severity should be mentioned along with the other parameters of the problem.

This is a simple and intuitive system that can characterise variations of a dynamic problem and help define a diverse set of test cases.

2.5 The Moving Peaks Benchmark

In order to evaluate the performance of optimisation algorithms on the different classes of DEs, a problem generating function that can be used to generate problem instances of the different environment classes is necessary. A number of benchmarks that have been used to evaluate optimisation algorithms in DEs can be found in [13, 15]. As previously stated, this thesis focuses on multivariate, continuous, nonlinear, uni-objective, unconstrained problems. The list of potential benchmarks is therefore reduced to simple functions such as those described by equations (2.7) and (2.8) but these functions do not cater for multimodal problems. Branke [13, 14, 15] developed the MPB as a candidate problem generator.

The MPB has one or several independent peak(s) moving through a multi-dimensional landscape in a fashion that is controlled by a number of parameters. The fitness at any given location is the maximum over all peak functions. The MPB function is defined as

$$F(\mathbf{x},t) = \max\{B(\mathbf{x}), \max_{b=1\ldots n_p}\{P(\mathbf{x}, p_{hb}(t), p_{wb}(t), \mathbf{p}_{\mathbf{p}_b}(t))\}\} \tag{2.16}$$

where $F(\mathbf{x},t)$ is the moving peak function, $B(\mathbf{x})$ is the time invariant "basis" landscape, n_p is the number of peaks, P is the function defining a peak's shape, p_h is the peak's height, p_w is the peak's width and $\mathbf{p_p}$ is the peak's location. P can be a number of different peak functions, the cone function that is used in the experimental part of this work is defined as:

$$P(\mathbf{x}, p_{hb}(t), p_{wb}(t), \mathbf{p}_{\mathbf{p}_b}(t)) = p_{hb}(t) - p_{wb}(t)\sqrt{\sum_{j=1}^{n_x}(\mathbf{p}_{\mathbf{p}_b}(t) - x_j)^2} \tag{2.17}$$

A single peak is fully characterised by its

- height, $p_{hb}(t) = p_{hb}(t-1) + \mathit{height\_severity} \cdot \sigma$,
- width, $p_{wb}(t) = p_{wb}(t-1) + \mathit{width\_severity} \cdot \sigma$,

- and location, $\mathbf{p}_{\mathbf{p}_b}(t) = \mathbf{p}_{\mathbf{p}_b}(t-1) + \mathbf{p}_{\mathbf{v}_b}(t)$,

where $\sigma \sim N(0,1)$ and the shift vector $\mathbf{p}_{\mathbf{v}_b}(t)$ is defined as

$$\mathbf{p}_{\mathbf{v}_b}(t) = \frac{s}{|\mathbf{p}_\mathbf{r} + \mathbf{p}_{\mathbf{v}_b}(t-1)|}((1-\lambda)\mathbf{p}_\mathbf{r} + \lambda \mathbf{p}_{\mathbf{v}_b}(t-1)) \tag{2.18}$$

where the random vector $\mathbf{p}_\mathbf{r}$ is created by generating random numbers for each dimension and normalizing its length to s. The variable λ determines if the peak's move is random ($\lambda = 0.0$) or if the peak tends to keep moving in the same direction as previously ($\lambda \neq 0.0$). If $\lambda = 1.0$, the peak moves in a straight line until it bounces off like a billiard ball when it reaches the limit of the search space. The spatial severity is therefore controlled by s, $height\_severity$ and $width\_severity$.

The location of each peak is modified at a regular change frequency measured in number of iterations (the original formula [15] implemented a change after a specified number of function evaluations). An instance of MPB is illustrated in figure 2.7.

Li and Yang and Li *et al.* [53, 55] have criticised the MPB for not being capable of generating binary or combinatorial problems. This limitation is not an issue here since the scope of the thesis only includes real space problem. But to be a suitable benchmark function to evaluate optimisation algorithms, the MPB must be able to simulate all types of DEs described in section 2.4.2. The MPB can behave as

- a type I environment when $height\_severity = 0$ and $s \neq 0$,
- a type II environment when $height\_severity \neq 0$ and $s = 0$,
- a type III environment when $height\_severity \neq 0$ and $s \neq 0$,
- a linear environment when $\lambda = 1.0$ and $s \neq 0$,
- a circular environment when $s = 0$ and the function is rotated on its centre,
- a random environment when $\lambda = 0.0$ and $s \neq 0$,

- a progressively changing environment if s, $height\_severity$ and $width\_severity$ are low relative to the size of the search space and the change frequency is high relative to the maximum number of iterations,

- an abruptly changing environment if s and/or $height\_severity$ and/or $width\_severity$ are high relative to the size of the search space and the change frequency is low relative to the maximum number of iterations,

- a chaotically changing environment if s and/or $height\_severity$ and/or $width\_severity$ are high relative to the size of the search space and the change frequency is high relative to the maximum number of iterations.

Therefore, all of Eberhart *et al.*'s and Angeline's types and all behavioural classes can be simulated using the MPB. It can be seen from the list above that the parameter settings to generate Eberhart *et al.*'s types, the behavioural classes, and Angeline's random type are not conflicting. The settings for the behavioural classes and Angeline's circular and linear types are also not mutually exclusive. Circular type I and III can be generated even though $s = 0$ since the rotation relocates the peak. A linear type II can be generated by having the peaks always grow (or always reduce in size), and a circular type II can be obtained by having the peaks grow and shrink in a repetitive sequence of states. Consequently, the MPB can generate environments characterised by any combination of Eberhart *et al.*'s types, Angeline's types, and the behavioural classes. Furthermore, the number of peaks can be set to generate both unimodal and multimodal environments. Homogeneous and heterogeneous multimodal environments can be generated by programming all peaks to have the exact same behaviour or not. The MPB can therefore simulate all dynamic types of environments described in the classification system proposed in this thesis.

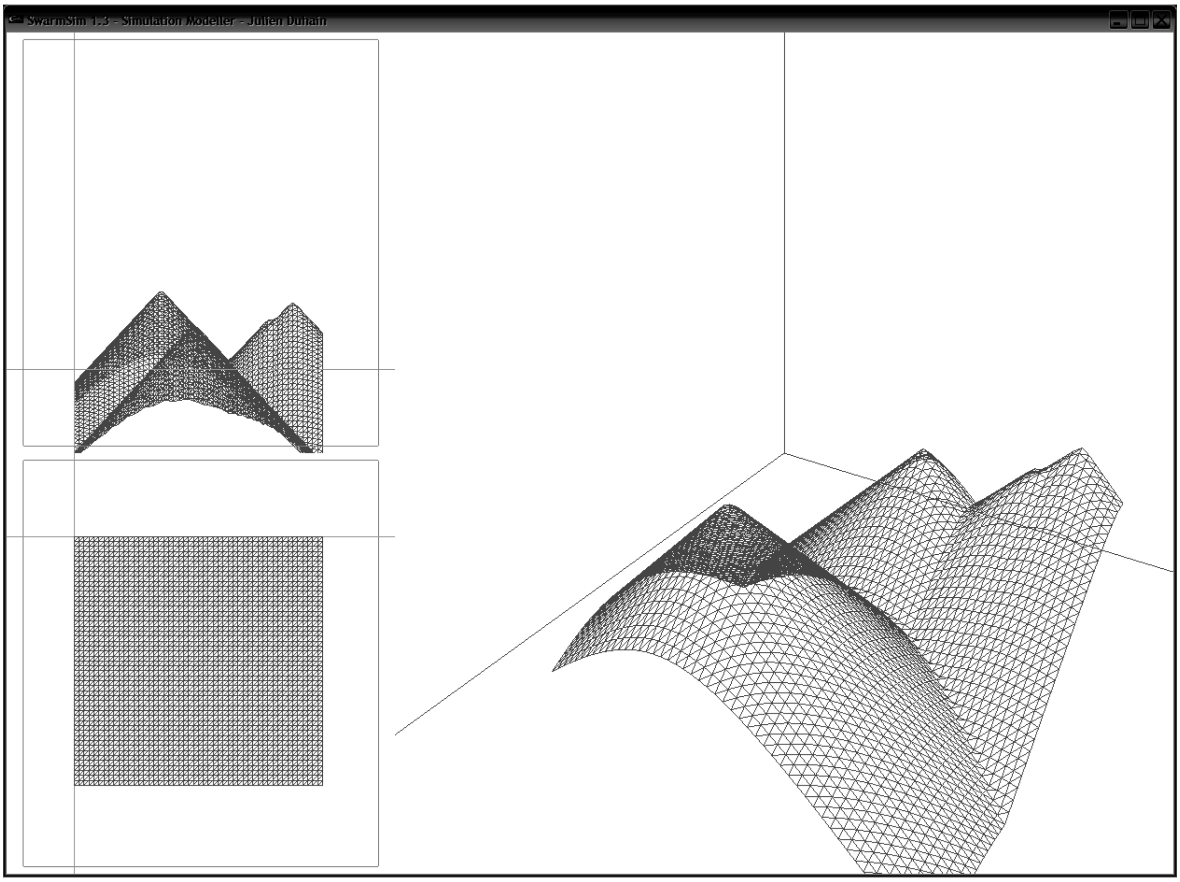


Figure 2.7: Two-dimensional MPB with four peaks in the domain [0, 100]

2.6 Summary

This chapter provided background on dynamically changing environments. A formal definition of dynamic optimisation problems was given, different classification schemes for DEs have been discussed, and an alternative classification scheme was proposed. The MPB that is used in the experimental part of this work was described as a means to generate the dynamic optimisation problems considered. The next chapter provides background on PSO.

Chapter 3

Particle Swarm Optimisation

"Men are like sheep, of which a flock is more easily driven than a single one."

– Richard Whately

The previous chapter described how mathematical functions can represent static and dynamic problems. This chapter presents the original algorithm for PSO, an algorithm capable of optimising static functions.

3.1 Introduction

PSO is a swarm-based search algorithm where a swarm of interconnected particles travels through a search space looking for an optimal solution. This chapter presents PSO and explains the concepts relevant to the understanding of the experiments conducted in part II. The original PSO algorithm is discussed in section 3.2, while section 3.3 describes the various parameters and strategies applicable to PSO.

3.2 Original PSO Model

A large group of students go out on a Friday night, all hoping to have a great time. Some go to the club, some go to a house party, some go to the pub etc. From time to time one of the students gives a phone call to some of his best friends in the group to find out who is having more fun. People finding out that a better party is going on elsewhere are tempted to join it. At the same time, the students tend to go to their usual hangout which always remains an attractive option. As the night goes, the grapevine is likely to gather everyone at the same place, hopefully the party with the best atmosphere and the cheapest drinks.

This metaphor illustrates the functioning of a PSO algorithm where individuals with only limited information about the search space interact to try to find the best solution to the optimisation problem. PSO is a simple and efficient optimisation algorithm whose concept evolved from an attempt of Eberhart and Kennedy [49] to graphically simulate the choreography of a bird flock. During the execution of a PSO algorithm, a population or *swarm* of interconnected *particles* "flies" in a hyper-dimensional search space looking for an optimal solution.

Every particle represents a potential solution. If the problem to solve has n variables, the particles evolve in an n-dimensional search space. The position of a particle in a dimension defines the value of a variable of the potential solution this particle represents. Each particle has a *fitness* value which indicates the quality of the solution. The fitness value can be calculated in various ways depending on the problem to solve. In the case of function maximisation, the fitness is simply the value obtained by passing the particle's position as argument to the function. The particle of the swarm with the highest fitness is referred to as the *global best* (*gbest*).

When the algorithm starts, each particle is positioned randomly and its velocity is set to null. Then, each particle's velocity is determined by a combination of the *cognitive component* and the *social component*. The cognitive component is influenced by the best position ever visited by the particle, referred to as the *personal best* (*pbest*), a constant and a random vector. Each particle keeps track of its *pbest* position. The fitness at *pbest* is compared with the fitness at the particle's current position during each iteration of the algorithm, and the *pbest* is updated if the current position's fitness is higher. The social component is influenced by the best position occupied by a connected particle, a constant and a random vector. The neighbourhood of a particle defines to which other particles in the swarm that particle is connected. Typically, the neighbourhood is based on the numerical indices of the particles and not on the proximity of the particles. In the original *gbest* PSO, all particles are interconnected.

Iteration after iteration, the particles tend to return to previously successful regions of the search space while being attracted to the best occupied position in their neighbourhood and the swarm progressively converges. *Convergence* refers to the fact that the magnitude of the changes in particle positions becomes smaller and the swarm progresses towards a stable state also called an *equilibrium state*. At convergence, velocities tend to zero. Theoretical work [90, 105] has proven that the original *gbest* PSO has all its particles converge on a weighted average of the *pbest* and the *gbest* positions.

The PSO algorithm is summarised in algorithm 3.1, where $\hat{\mathbf{y}}_i(t)$ is the *neighbourhood best* (*nbest*), of particle i at iteration t, that is the best position found in the neighbourhood $\mathcal{N}_i$. The *nbest* of i is defined as

$$\hat{\mathbf{y}}_i(t+1) \in \{\mathcal{N}_i \mid f(\hat{\mathbf{y}}_i(t+1)) = \max\{f(\mathbf{x})\}, \forall \mathbf{x} \in \mathcal{N}_i\} \tag{3.1}$$

where $f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ is the fitness function. If $\mathbf{x}_i(t)$ denotes the position of particle i at iteration t and $\mathbf{v}_i(t+1)$ represents the velocity of that particle, the **position** of the particle at iteration $t+1$ is

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1) \tag{3.2}$$

If $v_{ij}(t)$ denotes the **velocity** of particle i in dimension j at iteration t then

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)] \tag{3.3}$$

where c_1 and c_2 are constants, $\mathbf{r_1}$ and $\mathbf{r_2}$ are random vectors with values in the range $[0-1]$ sampled from a uniform distribution, y_{ij} is the position of particle i's *pbest* in

Create and initialise an n_x-dimensional swarm, S of $S.n_s$ particles;
repeat:
  for *each particle* $i = 1, \ldots, S.n_s$ **do**
    // set the *pbest* position
    if $f(S.\mathbf{x}_i) > f(S.\mathbf{y}_i)$ **then**
      $S.\mathbf{y}_i = S.\mathbf{x}_i$;
    end
    // set the *nbest* position
    if $f(S.\mathbf{y}_i) > f(S.\hat{\mathbf{y}}_i)$ **then**
      $S.\hat{\mathbf{y}}_i = S.\mathbf{y}_i$;
    end
  end
  for *each particle* $i = 1, \ldots, S.n_s$ **do**
    update the velocity using equation (3.3);
    update the position using equation (3.2);
  end
until stopping condition is true;

Algorithm 3.1: PSO algorithm with synchronous update

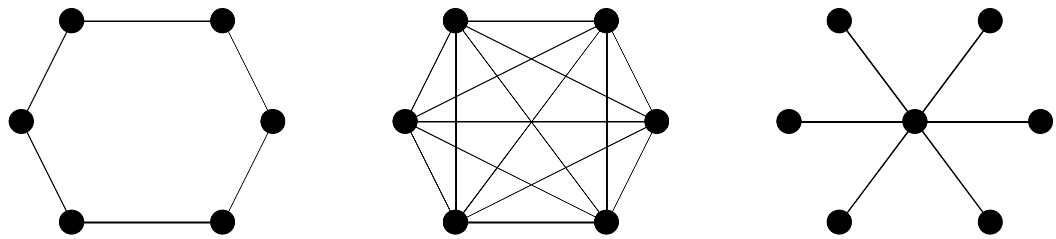
dimension j, $\hat{y}_{ij}$ is the position of particle i's *nbest* in dimension j and x_{ij} is the current position of particle i in dimension j.

If $\mathbf{y}_i(t)$ denotes the ***pbest*** of the particle i at iteration t then

$$\mathbf{y}_i(t+1) = \begin{cases} \mathbf{y}_i(t) & \text{if } f(\mathbf{x}_i(t+1)) \leq f(\mathbf{y}_i(t)) \\ \mathbf{x}_i(t+1) & \text{if } f(\mathbf{x}_i(t+1)) > f(\mathbf{y}_i(t)) \end{cases} \tag{3.4}$$

3.3 Parameters of the PSO Algorithm

A number of parameters affect the swarm's behaviour and efficiency [66, 84, 86, 87, 88, 93]. These parameters have interconnected influences on the algorithm's performance.



(a) Ring topology (b) Star topology (c) Wheel topology

Figure 3.1: Examples of social network structure

The best value for each of parameter is problem dependant. The roles of each parameter are discussed below:

- **Neighbourhood structure**

  Particles are grouped into overlapping neighbourhood of any number of particles.

  There exist various *neighbourhood topologies* or *social structures* as illustrated in Figure 3.1. An example of tightly connected social structure is the *star* topology where all particles are connected. The *ring* topology is an example of loosely connected social structure where a particle is only connected to two others in a ring like fashion. PSO algorithms are sometimes defined by the social structure they use. For instance, the *gbest* PSO is defined by its use of the star topology and the *lbest* PSO is defined by its use the ring topology. Algorithm 3.1 defines a *gbest* PSO if the neighbourhood size is equal to the swarm size and an *lbest* PSO otherwise. In the case of a *gbest* PSO, the same $\hat{\mathbf{y}}_i(t)$ is shared by all particles.

  The speed at which information travels through the population depends on how the particles are connected to each other [47]. A very connected swarm converges faster towards a good solution but might leave some areas of the search space unexplored and is therefore more likely to settle in a local optimum. Since the neighbourhoods are overlapping, a swarm with fewer connections is also likely to converge, although at a slower pace as the particles "spend more time" to explore the search space. A fast converging swarm which concentrates the search on a promising area to refine a candidate solution is said to focus on *exploitation*. A swarm that visits more regions of the search space to find a good optimum before converging focuses on *exploration*, followed by an exploitation step to refine the found optimum.

- **Acceleration coefficients**

  In equation (3.3), c_1 and c_2 are the acceleration coefficients also called *trust parameters* because they represent how much trust a particle has in its *pbest* (c_1) and in its *nbest* (c_2). The coefficient c_1, referred to as *nostalgia*, is modified by the random vector $\mathbf{r}_1$ and determines the influence of the cognitive component on the velocity of the particles. The coefficient c_2, referred to as *envy*, is modified by the random vector $\mathbf{r}_2$ and determines the influence of the social component.

  Although the best values for the acceleration coefficients are problem dependant, good performances are often achieved when the values of nostalgia and envy are close [29]. An algorithm where c_1 is much larger than c_2 leads the particles to excessive wandering and an algorithm with c_2 much larger than c_1 leads to premature convergence of the swarm.

  The trajectory of the particles is smoother and the particles explore far from the good regions before being pulled back if both acceleration coefficients have low values [29]. High values cause higher acceleration and more abrupt movement around the good regions. Eberhart and Shi [26] empirically found $c_1 = c_2 = 1.49618$ to be a good parameter choice in general. This was later confirmed by Van den Bergh [92].

- **Swarm size**

  Having more particles in the swarm increases the computational complexity per iteration, but permits a higher initial diversity and allows more areas of the search space to be investigated at the same time. Although empirical studies have shown that even a small swarm can find optimal solution [18, 93], smooth search spaces require less particles to locate an optimal solution than rough surfaces and the minimal number of particles required to find an optimal solution is problem dependant.

- **Termination conditions**

  The stopping conditions are the conditions that have to be met for the PSO algorithm to terminate. If the **number of iterations** is the stopping condition, the algorithm runs until that number is reached, and then stops. The swarm does not have the time to find and to exploit a good solution if the value is too small and for static environments, too large values unnecessarily add computational complexity. In DEs, the swarm has to be exposed to a sufficient number of environmental states for an experiment to be meaningful. Hence the maximum number of iterations has to be the stopping condition with a limit set reasonably high. Other stopping conditions include to terminate when an acceptable solution is reached, when there is no improvement in the fitness of the best solution found, when the radius of the swarm is too small, or when the objective function slope approximates zero [29].

- **Velocity clamping**

  If unchecked, particles tend to have larger and larger position updates. This acceleration can lead the particles to diverge and leave the search space [28]. The velocity of a particle can therefore be clamped to a *maximum velocity* if it exceeds a certain value thereby limiting step sizes. Let $V_{max,j}$ denote the maximum velocity allowed for dimension j:

$$v_{ij}(t+1) = \begin{cases} v'_{ij}(t) & \text{if } v'_{ij}(t+1) \leq V_{max,j} \\ V_{max,j} & \text{if } v'_{ij}(t+1) > V_{max,j} \end{cases} \tag{3.5}$$

  where v'_{ij} is calculated using equation (3.3).

  $V_{max,j}$ controls the granularity of the search and its value must therefore be chosen carefully. A large $V_{max,j}$ encourages exploration, while a small $V_{max,j}$ favours exploitation. If the value is too large, particles may jump over an optimum and if the value is too small, particles may take longer to find a good solution or become trapped in a local optimum.

Velocity clamping also has some disadvantages. If through clamping, the particle's velocity is reduced by different amounts for several dimensions, the direction in which the particle moves is affected. Also, if all velocities are equal to the maximum velocity, particles only search on the boundaries of a hypercube define by $[\mathbf{x}_i(t) - \mathbf{V}_{max}, \mathbf{x}_i(t) + \mathbf{V}_{max}]$ [29]. Introducing an inertia weight or reducing $V_{max,j}$ over time can solve this problem.

- **Inertia weight**

  The inertia weight introduced by Shi and Eberhart [86] helps controlling the exploration and exploitation capacity of the swarm by adjusting the importance of the previous velocity on the new velocity.

  The introduction of an inertia weight modifies equation (3.3) as follows:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)] \tag{3.6}$$

  where w is the inertia weight. Giving a large value to w promotes exploration, but if w is higher than 1, the particles accelerate towards the maximum velocity and the swarm diverges. A small value encourages exploitation but limits the exploration ability of the swarm.

  The value of the inertia weight can be varied over time. In static environments, the weight would usually start with a higher value and decrease progressively to improve exploration first and exploitation later. There are several approaches to weight adjustment, including random adjustment [75], linear [87] or non linear decreasing [65, 76, 98, 99], fuzzy adaptive inertia [89] and increasing inertia [104, 105].

  Eberhart and Shi [26] empirically found that the value 0.7298 works very well in general. However, the value of w should be chosen in conjunction with the value of c_1 and c_2 as they have interconnected effects on the behaviour of the swarm. Van den Bergh [92] demonstrated that the trajectory of the particles converges if

$$w > \frac{(c_1 + c_2)}{2} - 1 \tag{3.7}$$

- **Constriction coefficient**

  Using a constriction coefficient [22, 23] is an alternative to the inertia weight that removes the need for velocity clamping, although using constriction with velocity clamping can lead to a faster convergence rate [26]. If constriction is used, equation (3.3) is modified as follows:

$$v_{ij}(t+1) = \chi[v_{ij}(t) + \phi_1(y_{ij}(t) - x_{ij}(t)) + \phi_2(\hat{y}_{ij}(t) - x_{ij}(t))] \tag{3.8}$$

  where

$$\chi = \frac{2\kappa}{|2 - \phi - \sqrt{\phi(\phi - 4)}|} \tag{3.9}$$

  with $\phi = \phi_1 + \phi_2$, $\phi_1 = c_1 r_1$, and $\phi_2 = c_2 r_2$. A swarm with a constriction coefficient is guaranteed to converge if $\phi \geq 4$ and $\kappa \in [0, 1]$.

- **Synchronous and asynchronous updates**

  The *pbest* and *nbest* positions can be updated synchronously or asynchronously. With synchronous updates the *pbest* and *nbests* are updated once per iteration, separately from the position updates, as shown in the algorithm 3.1. With asynchronous updates, the best positions are re-calculated after each particle's position update which allows immediate feedback.

- **Neighbourhood best update strategy**

  There are different ways in which the *nbest* of a particle can be selected. With a *memory based* update strategy, the *nbest* of a particle is selected based on the fitness of the *pbest* of the neighbouring particles. With *iteration based* update strategy, the *nbest* is selected based on the fitness at the positions currently occupied by the neighbouring particles). Since the *pbest* positions changes less frequently than the particles' positions, the particles can be attracted towards the same location for several consecutive iterations when a memory based update strategy is used. A memory based update strategy therefore favours exploitation more than an iteration based strategy, which favours exploration.

- **Velocity model**

Equation (3.3) can be modified in several ways to create new velocity models [46].

The *cognition-only model* excludes the social component from the original velocity formula, replacing equation (3.3) by

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] \tag{3.10}$$

This model promotes exploration as it allows the particles to climb different optima independently. However, Kennedy [46] reported that the cognition-only model is more vulnerable to failure and slower to reach a good solution than the full model.

The *social-only model* excludes the cognitive component from equation (3.3) as follows:

$$v_{ij}(t+1) = v_{ij}(t) + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)] \tag{3.11}$$

With this model, the particles are only attracted towards their *nbest* and converge faster towards a single peak. Kennedy [46] has also shown that the social-only model can outperform the full and cognition only models in static environments.

The *selfless model* is a social-only model where a particle is not allowed to pick itself as *nbest*. The selfless model has shown to be faster than the social-only model on some problems [46].

3.4 Summary

This chapter described the original PSO algorithm and explained the influences of the various parameters on the performance of the algorithm. The original PSO and its variants were developed for static optimisation problems, and the next chapter discusses consequences for applying PSO to DEs.

Chapter 4

Particle Swarm Optimisation and Dynamic Environments

"Adapt or perish, now as ever, is nature's inexorable imperative."

– H. G. Wells

Chapters 2 and 3 have introduced PSO and DE, the two major concepts necessary to the understanding of this work. These concepts can now be put together by investigating the outcomes of applying PSO algorithms to DEs. This chapter presents a number of observations and findings from previous research and describes the variations of the PSO algorithm designed for DEs that are used in the experimental part of this thesis.

4.1 Intoduction

In spite of its capacity to solve optimisation problems, the original PSO has shown some limitations when working in DEs. The objectives of this chapter are to describe these limitations and to introduce a number of approaches that have been used to overcome these issues. Section 4.2 presents the strengths and weaknesses exhibited by the PSO algorithm when PSO is applied to dynamic optimisation functions, section 4.3 investigates how to optimise the parameters of the PSO algorithm to obtain better performance in DEs, and sections 4.4 to 4.7 describe various strategies that can be used to modify PSO in order to overcome the limitations of the algorithm. Finally, section 4.8 describes a number of variations of the PSO algorithm that have been designed to optimise dynamic problems.

4.2 Strengths and Limitations

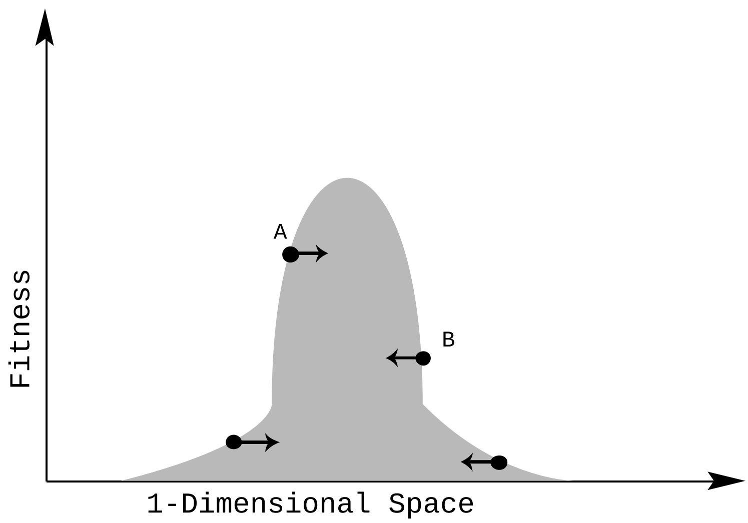
PSO has a natural capacity to adapt to minor modification of the landscape [20, 27, 40]. As a swarm converges, particles oscillate around the global optimum, getting progressively closer to the global best position [68, 69, 92, 94]. In a type I environment, if the optimum moves to a nearby position within the converging swarm, a particle may "land" near the new optimum and attract the other particles towards the new best position. However, if the optimum moves to a remote location, away from the converging swarm, the PSO algorithm is less likely to detect the new optimum as no particle is exploring that section of the search space. PSO demonstrates the same limitation in a type II environment when a new peak emerges. A new peak appearing far from the converging swarm is less likely to be detected than a peak appearing in an area surrounded by particles. As the swarm converges, the degree of dispersion of the particles in the swarm (*diversity*) decreases [92], which reduces the area where a new optimum can be detected. **Diversity loss** is therefore a major limitation of the PSO algorithm applied to DEs. To adapt to a changing environment, diversity is extremely important and must either be maintained at all time or re-introduced after a change.

In type I or type III environments, the *pbest* of a particle may attract this particle towards an area of the search space that only contains poor solutions because the optimum that used to be located in this area has now moved to another location. This phenomenon is known as **outdated memory** and is the other major limitation of PSO in DEs. In a type II environment, the performance of the algorithm is not affected by outdated memory if the value of the optimum is increasing, since the optimum remains in the same location and the previously attractive positions have weaker values than that of the optimum. But, if the value of the optimum is decreasing, the cognitive component can lead the particles towards a formerly good position instead of towards the new optimum. Figure 4.1 illustrates what happens: the particles are converging towards the maximum when a change occurs, bringing the maximum's value down. Particle A's *pbest* is now higher than the current maximum. A's inertia weight may push A further up the hill but its *pbest* cannot possibly be updated to a position of higher fitness which leads A to be attracted towards the location of the outdated *pbest*. For the other particles, the global best is A's *pbest* (if a memory based neighbourhood best update strategy is used), and no matter where they go, no particle can ever find a better position. Even though particle B is closer to the maximum, the particles are destined to oscillate between their outdated *pbest* and outdated *nbest*. Figure 4.2 shows a swarm in an abruptly changing environment of two dimensions and five peaks. Initially, the swarm converges towards an optimum but after the environment has changed, the swarm remains in place because of outdated memory. The *pbest* of the particles stays unchanged and the particles keep oscillating between their *nbest* and their *pbest*. Figure 4.2(a) shows that before the first change occurs, the swarm exploits the peaks. After a change, the particles remain in the same area of the search space and exploitation stops. Figure 4.2(b) shows that 200 iterations after the change, the particles have stayed in the same area.

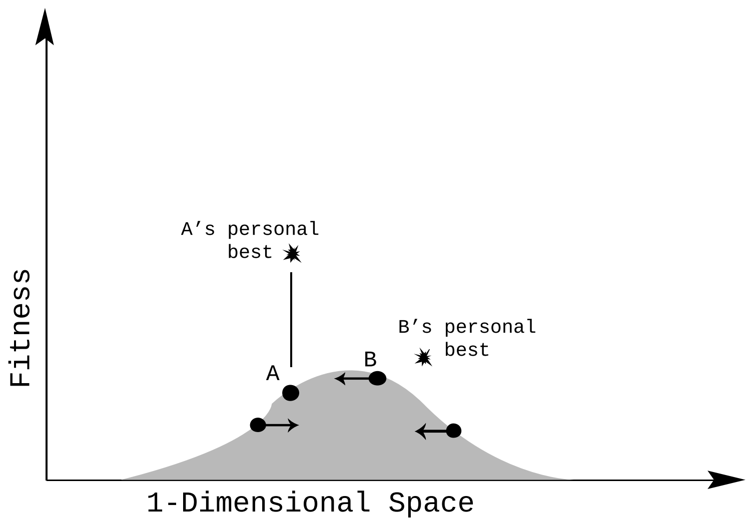
4.3 Parameter Optimisation

The limitations of PSO can be partially overcome by adjusting some of the parameters described in section 3.3. The following list describes how the standard PSO algorithm can be optimised in order to achieve a better performance level in DEs:

- The selection of the **neighbourhood best update strategy** influences outdated



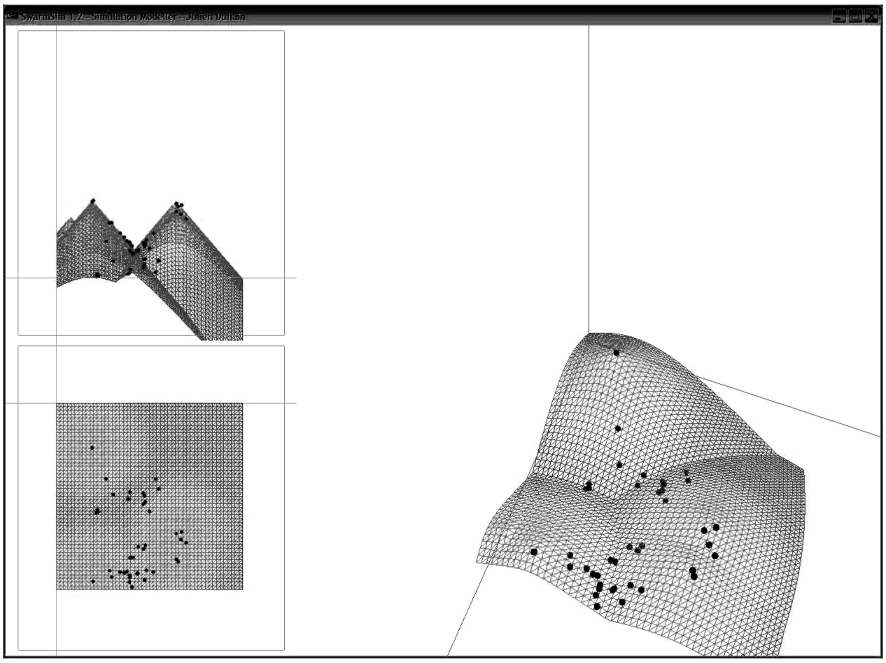
(a) Before change



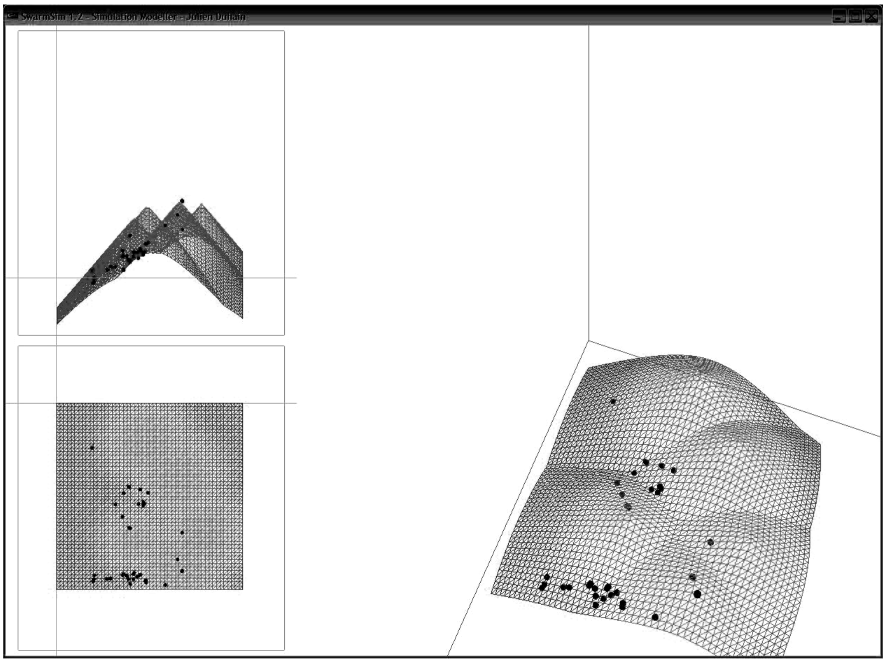
(b) After change

Figure 4.1: Outdated memory limitation

memory. Indeed, if the iteration based strategy defined in section 3.3 is used, each particle updates its *nbest* with the position in the neighbourhood that currently has the highest fitness. Therefore the *nbest* of the particles is not affected by outdated



(a) Before first change



(b) 200 iterations after change

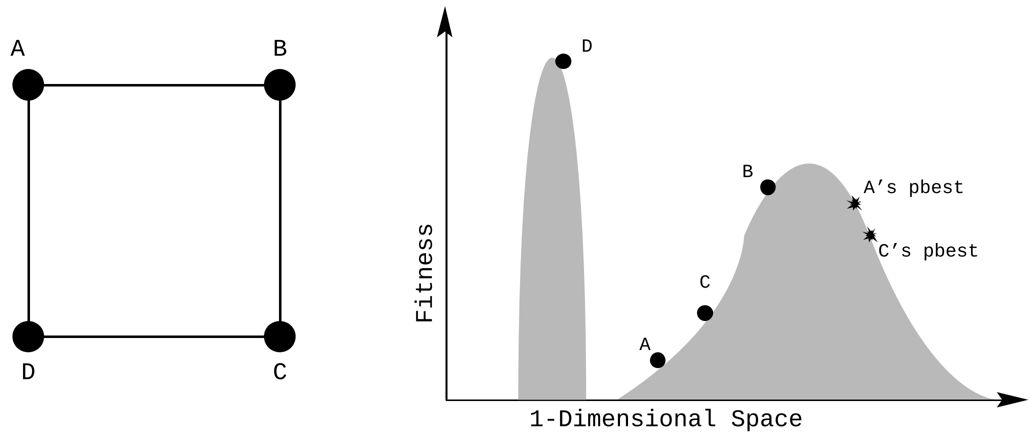
Figure 4.2: Effect of outdated memory on swarm convergence

memory. However, because the *pbests* are still potentially outdated, particles can nonetheless be attracted to poor solutions that formerly had a high fitness.

- Carlisle and Dozier [20] have studied the influence that the various **velocity models** from section 3.3 have on the performances of PSO in DEs. They have shown that the selfless and cognition-only models do not perform well in changing environments. Carlisle and Dozier also showed that the social-only model outperforms the full model in environments with low temporal severity, although the performances of the social-only model deteriorate faster than that of the full model as the speed at which the optimum moves through the search space is increased. The better performance of the social-only model can be explained by the fact that this model does not have a cognitive component, hence does not experience outdated memory problems. Outdated memory can therefore be avoided but only at the cost of losing the information contained in the *pbest* of the particles.

- For static problems, the **inertia weight** can be initialised to a large value and decreased over time to promote exploration first and exploitation later [29]. However, in DEs, new peaks may appear after a change and optima may go undetected if the diversity of the swarm has dropped too much. Therefore w should be reinitialised to its original larger value every time a change is detected. As an alternative solution, Eberhart and Shi [27] proposed using a randomly selected inertia weight in the range $[0.5, 1]$ to ensure a good mix of exploration and exploitation. In their study Eberhart and Shi used $c_1 = c_2 = 1.494$. However, to select the optimal value for w in DEs, the value of c_1 and c_2 should be taken into consideration due to their dependencies and influence on convergence behaviour. A swarm having a constant or randomly selected inertia weight does not converge as fast as a swarm having a decreasing inertia weight, but as long as convergence takes place, diversity loss cannot be avoided.

- **Velocity clamping** with small $V_{max,j}$ reduces exploration which can lead the swarm to be trapped in an old optimum. The value of $V_{max,j}$ should therefore be carefully chosen. Removing velocity clamping completely can also be considered as it promotes exploration [27]. Although, as previously stated, slowing down the



(a) Topology (b) Particles' position

Figure 4.3: Limited propagation of information

convergence of the swarm cannot fully overcome the issue of diversity loss.

- Selection of the **social network structure** also influences diversity loss. In environments with multiple optima, the *pbest* and the *nbest* of a particle can be located on different peaks. This can lead particles to jump back and forth between unattractive positions until either a particle's *pbest* or *nbest* is updated. Such particles can prevent the propagation of information from one neighbourhood to the next if a topology with loosely coupled particles is used [31]. This phenomenon is illustrated in figure 4.3 where particles B and D can exploit different peaks because they are separated by A and C. This has the benefit of allowing different parts of the swarm to exploit different peaks simultaneously as shown in figure 4.4 where the swarm in a static environment has not converged after 500 iterations and particles are visible on four of the peaks.

  A *fine-grained PSO* is a PSO algorithm where the particles are mapped onto a two dimensional grid. On the grid, the last particle of a row is connected to the first particle of that row, and similarly the first and last particles of a column are connected. Each particle's neighbourhood, known as a *Von Neumann neighbourhood*, therefore includes four other particles as illustrated in figure 4.5. Kennedy

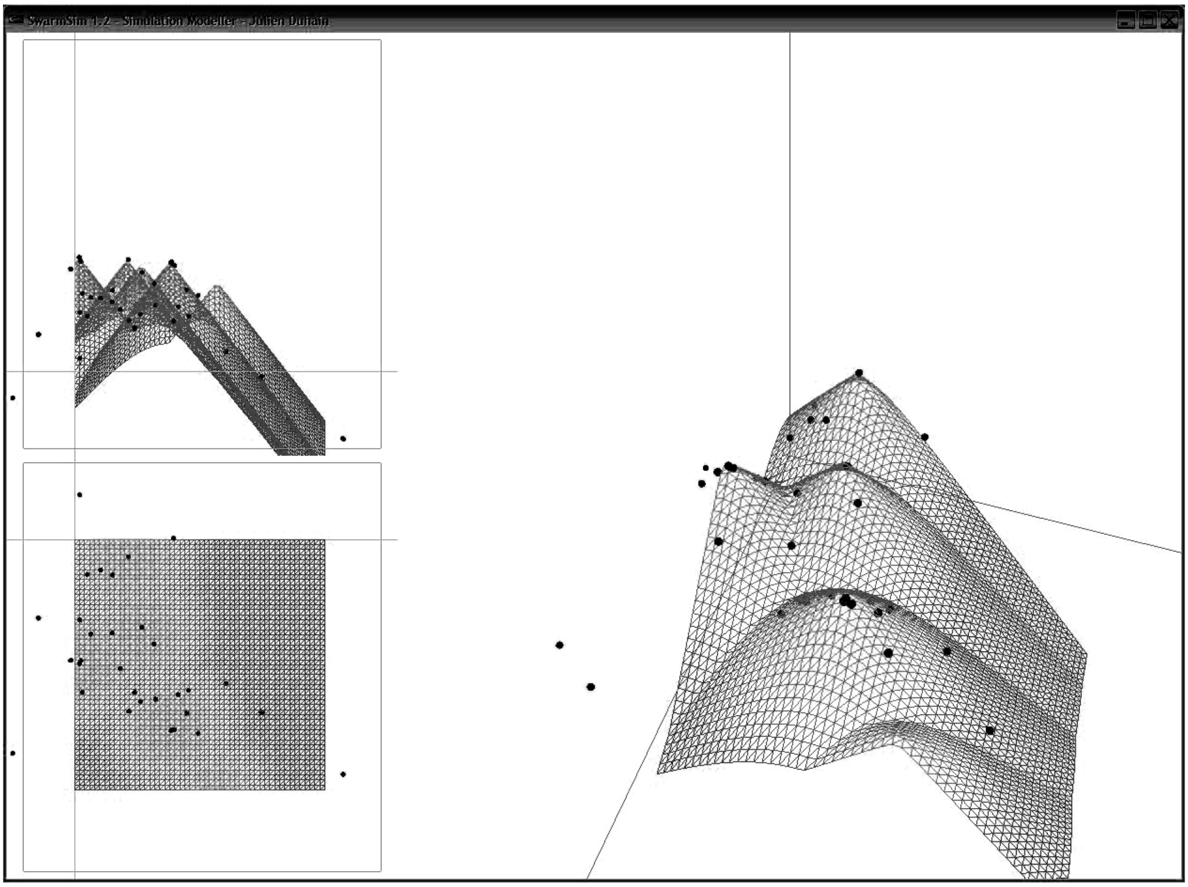


Figure 4.4: Exploitation of several peaks with a loosely coupled topology (particles are allowed to leave the search space)

and Mendes [50] have shown that the Von Neumann topology outperforms other neighbourhood topologies for a large number of static problems. Li and Dam's experiments in DEs [58] demonstrated that the Von Neumann social topology maintains better diversity than the more connected topologies since a particle's influence can only propagate gradually through the swarm.

Jansen and Middendorf [43] also reported improvement over the standard PSO when using a *hierarchical* neighbourhood structure. In this network structure, the particles are arranged in a hierarchical tree where each node contains a single particle, and the *nbest* of a particle is the position of the particle directly above. If a particle at a child node is better than the *pbest* of the particle at the parent

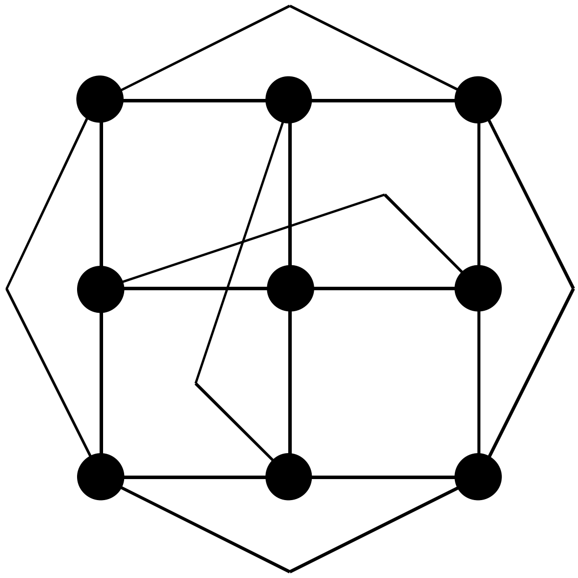


Figure 4.5: Von Neumann neighbourhood topology

node, the two particles are exchanged.

Because they maintain diversity longer, the fine-grained PSO and the hierarchical PSO are more efficient in DEs than PSO's using other, more connected social network structures. Yet, the neighbourhood topologies used by these algorithms convey information through the swarm at a faster pace than the ring topology, therefore offering a good trade-off between exploitation and exploration.

As described above, research has shown that through parameter optimisation, diversity loss can be slowed down but not fully avoided. It was also shown that, at the cost of losing the information contained in the cognitive component, outdated memory can be eliminated. The following sections describe ways to modify the standard PSO algorithm to further improve the performance of PSO in DEs.

4.4 Re-evaluation of the *pbest*

In this thesis, *pbest* re-evaluation refers to the fact that, before the *pbest* is updated, the fitness of the *pbest* of a particle is re-calculated using the current state of the environment. When using the full velocity update model, re-evaluating the *pbest* of the particles is imperative to avoid outdated memory. If fitness evaluation is computationally inexpensive, the *pbest* of each particle can be re-evaluated every iteration. Although, if the time of change is known, the *pbest* should only be re-evaluated after a change occurs to avoid unnecessary computations. If fitness evaluation is computationally expensive and if the time of change is unknown, a detection and response strategy (described in the next section) can rather be used to decide when a re-evaluation should take place. A periodic re-evaluation is also an option if no detection mechanism is used.

4.5 Detection and Response

The first approach that can be taken to overcome diversity loss is to reintroduce diversity when a change is detected. This section investigate this approach.

In abruptly changing environments, a detection and response strategy can save computation time by only re-evaluating the *pbest* of the particles when a change occurs. Detection and response strategies also allow the algorithm to change behaviour to focus on exploitation when the environment is static and on exploration when the landscape changes. In environments with high temporal severity, changes happen too frequently for a detection and response strategy to be effective as the response is triggered too often. To avoid this problem, a threshold can be set so that the detection mechanism ignores small changes.

Section 4.5.1 discusses change detection and describes various detection mechanisms, and section 4.5.2 discusses how swarm algorithms can respond to an environmental change.

4.5.1 Change Detection

If the algorithm is informed of when the changes occur, or if the changes are periodic and the system can calculate the time of the next change, no detection mechanism is needed. If the algorithm has no prior knowledge about the time of change, different techniques can be used to detect a modification in the environment.

Deterioration of the performance of the current best particle is an indicator that the landscape has changed [24, 95, 96, 97], but that is assuming that the quality of the old solution decreases which may not be the case.

An approach used with genetic algorithms (GA) is to re-evaluate several individuals every generation [14]. If at least one fitness has changed, the environment has been modified. Hu and Eberhart [40] proposed to monitor changes in fitness at the position of the global best particle based on the assumption that a change in the optimum's location leads to a change in fitness of the current best solution. The second-best particle (and possibly the third-best, etc) can also be monitored to increase accuracy and prevent false alarms [41]. In a unimodal environment, this approach is likely to detect all changes, but it may fail to do so in a multimodal, heterogeneous environment. Monitoring particles' position allows only to detect changes that take place inside the radius of the swarm and as the swarm converges, the area covered by the population becomes smaller and smaller. The appearance of a new peak, for instance, could go undetected if the environment stays static in the area where the swarm has converged. This option should only be considered for algorithms constantly maintaining a high level of diversity, i.e. algorithms that do not rely on a detection and response mechanism to reintroduce diversity.

A better option is to use a number of sentry points [19, 21]. Sentries store a copy of the fitness value at a random location. At the start of every iteration, the stored value is compared to the new fitness at that point. Using a large number of sentries can be computationally expensive – depending on the cost of evaluation – but is more accurate since several parts of the environment are monitored. Sentries can be re-evaluated at every iteration or periodically if the computational cost of fitness calculation is high. It is also possible to set a threshold to only detect severe changes or, in progressively changing environments, to wait until the landscape has considerably changed before triggering a response. It is important to uniformaly distribute the sentries across the environment to assure a good monitoring of the search space. Alternatively, the sentries can periodically be randomly relocated.

If a threshold is used to avoid responding to insignificant changes, the sentries (or particles, if particles are used as sentries) should not update their stored fitness until a response has been triggered. In this way, an accumulation of small changes to the landscape can be noticed.

Other techniques can also be used when dealing with real-world problems. Fogarty *et al.* [33] suggested using a validation module to evaluate the algorithm's performance. Maintaining a model of the environment and regularly checking its consistency with the real environment is yet another possibility [35, 44, 45, 77].

4.5.2 Response to Change

When an environmental change is detected, the algorithm can respond to the change in several ways. If the detection and response strategy is used only to save unnecessary re-evaluation of the *pbest*, the algorithm does not change its behaviour when a change is detected but only re-evaluates the particles' *pbest*. In other cases where the objective of the response is to overcome diversity loss, diversity can be re-introduced in the swarm, typically by randomly reinitialising the position of some particles or even the whole swarm [27]. The details of such a reinitialisation strategy are discussed in section 4.8.1. Various other approaches to re-introduce diversity are discussed in the remainder of this chapter.

4.6 Diversity Maintenance

The second approach that can overcome diversity loss is to maintain diversity at all time.

To overcome premature convergence in static optimisation problem, Krink *et al.* [51] designed a *PSO model with Spatial Particle Extension* (SEPSO) where particles have a volume (spatial extension) so that two particles cannot share the same space. If two particles collide, the particles bounce away and thus the clustering does not occur. To make already discovered optima unattractive to the swarm, Parsopoulos and Vrahatis [73] used *function stretching*, a technique which transforms the original objective function.

Using *inter-particle repulsion*, that is, having particles repel each other based on their proximity is a repulsion technique that has been used to maintain diversity in DEs [6, 8, 9, 10, 32]. Applying repulsion strategies can be computationally expensive if the distance between all particles must be measured. Also, repulsion methods may require fine tuning of the parameters to prevent particles from leaving the search space and allow exploitation while maintaining a sufficient level of diversity.

Alternatively, Blackwell and Branke [11] have proposed to have a number of particles being randomly re-located within a radius centered on the *gbest* instead of having their position updated using equation 3.2. This approach avoids the need for calculating inter-particle distance and allows control over the dispersion of the particles.

4.7 Swarm Sub-division and Parallel Tracking of Optima

The third approach to overcome diversity loss is to track every local optimum individually with a subset of particles from the swarm. In this way the algorithm already monitors areas of interest in case a local optimum becomes the global optimum after a change.

Speciation or *niching* was first developed to find all optima in multimodal problems [3, 4, 18, 48, 56, 72, 80, 81, 82] and later adapted and applied to DEs for its capacity to track multiple peaks simultaneously [54, 57, 70, 71, 83]. When speciation is used, the particles are allowed to leave the main swarm and form sub-swarms (also called *species*, *niches*, or *clusters*) based on inter-particle proximity. Each niche exploits a separate peak while the other particles continue exploring the search space. Niches are also allowed to merge or further divide.

An alternative approach to niching is to use multiple populations. Multiple populations (multi-swarm) also permits simultaneous tracking of several optima, but typically uses fixed-size sub-swarms instead of dynamically forming them [57].

Both speciation and multiple populations require that some level of diversity is kept within the sub-swarms to allow peak tracking, as well as between the sub-swarms to allow peak detection. To maintain diversity within the sub-swarm, diversity maintenance techniques can be used.

There exists other strategies to sub-divide the swarm and track the optima in parallel. One approach is to divide the search space into sub-parts or *cells*, and to distribute the particles among the cells [38]. Another approach is to have different types of sub-swarms collaborate [52, 59]. This way, a sub-swarm focuses on the detection of new peaks while one or more other sub-swarm(s) exploit(s) each peak independently.

4.8 Swarm Algorithms Designed for Dynamic Environments

A number of variations of the PSO algorithm have been developed to solve dynamic optimisation problems. These variations have proven to perform better in DEs than the standard PSO [6, 8, 9, 10, 11, 12, 27, 32, 38, 52, 54, 57, 59, 70, 71]. Discussing the efficiency of each variation of every algorithm is outside the scope of this work. However, it is important to select specific instances of algorithms and problems that can give a general idea of the entire range of existing algorithms. In the experimental part of this work, it is then possible to match the selected algorithms against a range of DE test cases and to analyse the results of these experiments.

A representative panel of PSO algorithms, namely the reinitialising PSO, charged PSO (CPSO), APSO, QSO, multi-swarm, and SAMS, are described in this section. These algorithms make use of the approaches presented in section 4.4 to 4.7 to improve their efficiency on dynamic optimisation problems. An algorithm that uses a detection and response strategy is presented in section 4.8.1, section 4.8.2 and 4.8.3 describe algorithms that use diversity maintenance. Sections 4.8.4 and 4.8.5 describe algorithms that use swarm sub-division and parallel tracking of optima. All the algorithms described implement re-evaluation of the *pbest* of the particles.

4.8.1 Reinitialising PSO

Eberhart and Shi [27] suggested reinitialising all or part of the swarm after an environmental change to reintroduce diversity. When the environment is static, the reinitialising PSO behaves like the standard PSO. When a change occurs, all or a percentage of the particles of the swarm are randomly repositioned in the search space in order to detect the new position of the optimum. If only part of the swarm is reinitialised, the particles that are reinitialised are selected randomly among the particles of the swarm. To keep track of the best known position(s), Eberhart and Shi also suggested to reinitialise all but the best particle or to only keep in place the particles that are the best in their neighbourhood. Since the aim of the reinitialisation is to increase diversity, the reinitialised particles should also have their velocity reset and their *pbest* set to the particle's current position to avoid being immediately attracted towards their former position. The values of $V_{max,j}$, w, c_1 and c_2 are constant and remain unaffected by the reinitialisation process. In addition, the *pbest* of the other particles should be re-evaluated. The reinitialising PSO algorithm is summarised in algorithm 4.1.

Reinitialisation introduces more diversity into the swarm, which prevents the swarm from converging within a small area. Among the algorithms presented in this section, the reinitialising PSO is the only one making use of a detection and response strategy. The reactive nature of the algorithm allows the swarm to use the standard PSO's capacity to exploit while overcoming the problem of diversity loss. One of the downsides of the reinitialising PSO is its dependence on the detection mechanism. If some of the changes go undetected, the reinitialising PSO may be subject to the limitations of the standard PSO. Also, if the environment has a high temporal severity, frequent reinitialisations can reduce the exploitation capacity of the algorithm.

4.8.2 Charged PSO and Atomic PSO

Blackwell and Bentley [6, 9, 10] proposed using inter-particle repulsion to maintain diversity at all time and developed the CPSO [9, 10] based on an analogy of electrostatic energy. The swarm of a CPSO is made of *charged particles*, a new type of particles that repel each other if they come too close to one another. Alternatively, the swarm can consist of a combination of charged particles and *neutral particles* that behave like standard particles without experiencing or exerting a repulsive force. The *atomic* PSO (APSO) [10] contains both charged and neutral particles. The neutral particles exploit the area around the global best, while the charged particles continue to explore the search space. The CPSO and APSO algorithms are outlined in algorithm 4.2, where the CPSO is the special case where all particles have a non-zero charge. Charged particles use equation (4.1) to update their velocity:

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)[y_{ij}(t) - x_{ij}(t)] + c_2 r_{2j}(t)[\hat{y}_{ij}(t) - x_{ij}(t)] + a_{ij}(t) \tag{4.1}$$

where $a_{ij}(t)$ is the acceleration which determines the magnitude of inter-particle repulsion for particle i in dimension j. The acceleration vector $\mathbf{a}_i(t)$ is defined as

$$\mathbf{a}_i(t) = \sum_{o=1, i \neq o}^{n_s} \mathbf{a}_{io}(t) \tag{4.2}$$

Create and initialise an n_x-dimensional swarm, S;
repeat:
    for *each particle* $i = 1, \ldots, S.n_s$ **do**
        // set the *pbest* position based on the current fitness at the *pbest*'s position
        if $f(S.\mathbf{x}_i) > f(S.\mathbf{y}_i)$ **then**
            $S.\mathbf{y}_i = S.\mathbf{x}_i$;
        end
        // set the *nbest* position
        if $f(S.\mathbf{y}_i) > f(S.\hat{\mathbf{y}}_i)$ **then**
            $S.\hat{\mathbf{y}}_i = S.\mathbf{y}_i$;
        end
    end
    for *each particle*, $i = 1, \ldots, S.n_s$ **do**
        update the velocity using equation (3.3);
        update the position using equation (3.2);
    end
    if change is detected **then**
        for *a subset of the particles*, $i = 1, \ldots, S.n_s$ *not including the gbest,* **do**
            randomly select a new position
            set velocity to zero
            set the *pbest* to current position
        end
    end
until stopping condition is true;

Algorithm 4.1: Reinitialising PSO

with the repulsion force between particle i and o defined as

$$\mathbf{a}_{io}(t) = \begin{cases} \left(\frac{Q_i Q_o}{\|\mathbf{x}_i(t)-\mathbf{x}_o(t)\|^3}\right)(\mathbf{x}_i(t) - \mathbf{x}_o(t)) & \text{if } 0 < R_c \leq \|\mathbf{x}_i(t) - \mathbf{x}_o(t)\| \leq R_p \\ \left(\frac{Q_i Q_o (\mathbf{x}_i(t)-\mathbf{x}_o(t))}{R_c^2 \|\mathbf{x}_i(t)-\mathbf{x}_o(t)\|}\right) & \text{if } \|\mathbf{x}_i(t) - \mathbf{x}_o(t)\| \leq R_c \\ 0 & \text{if } \|\mathbf{x}_i(t) - \mathbf{x}_o(t)\| > R_c \end{cases} \tag{4.3}$$

where Q_i is the charged magnitude of particle i, R_c is the core radius, and R_p is the perception limit of each particle. The range $[R_c, R_p]$ defines the range within which charged particles experience inter-particle repulsion. The purpose of the lower bound, R_c, is to limit the acceleration of the particles too close to one another. No repulsion takes place between particles further apart than the upper bound, R_p.

When a non-fully connected neighbourhood topology is used, the charged and the neutral particles should be randomly positioned in the topology so that the charged and neutral particles are interconnected and can exchange information more easily.

Blackwell and Bentley [10] found that atomic swarms outperform the charged swarms and empirical evaluations have found the CPSO and APSO to be very efficient in DEs [6, 7, 10].

The main drawback of CPSO and APSO is their computational complexity of $O(n^2)$ since the calculation of the inter-particle repulsion requires that the distance between all charged particles be computed.

4.8.3 Quantum Swarm Optimisation

In the quantum model of the atom, orbiting electrons are replaced by a quantum cloud which is a probability distribution governing the position of the electron upon measurement [85]. QSO was developed by Blackwell and Branke [11, 12] as a simplified and less computationally expensive version of the CPSO. As an atomic swarm, a quantum swarm contains neutral particles and charged particles, also called *quantum particles*. Instead of repelling each other, the quantum particles are randomly placed within a multi-dimensional sphere, B_n, of radius r_{cloud} centred on the current global best particle of the swarm as follows:

$$\mathbf{x}_i(t+1) = \begin{cases} \mathbf{x}_i(t) + \mathbf{v}_i(t+1) & \text{if } Q_i = 0 \\ B_n(r_{cloud}) & \text{if } Q_i \neq 0 \end{cases} \tag{4.4}$$

Create and initialise an n_x-dimensional swarm, S;
A percentage of the particles have their charge set to a non-zero value (charged particles), and the rest of the particles have their charge set to zero (neutral particles);
repeat:
  for *each particle* $i = 1, \ldots, S.n_s$ **do**
    // set the *pbest* position based on the current fitness at the *pbest*'s position
    if $f(S.\mathbf{x}_i) > f(S.\mathbf{y}_i)$ **then**
      $S.\mathbf{y}_i = S.\mathbf{x}_i$;
    end
    // set the *nbest* position
    if $f(S.\mathbf{y}_i) > f(S.\hat{\mathbf{y}}_i)$ **then**
      $S.\hat{\mathbf{y}}_i = S.\mathbf{y}_i$;
    end
  end
  for *each particle* $i = 1, \ldots, S.n_s$ **do**
    update the velocity using equation (4.1);
    update the position using equation (3.2);
  end
until stopping condition is true;

Algorithm 4.2: APSO

QSO is summarised in algorithm 4.3.

4.8.4 Multi-swarm

The idea behind the multi-swarm algorithm introduced by Blackwell and Branke [8, 11, 12] is to divide the swarm into a number of sub-swarms each tracking a different peak. Although the dynamics governing the position and velocity updates of a particle in a particular sub-swarm are specified only by the parameters of that sub-swarm, the sub-swarms interact locally via an *exclusion* operator and globally via an *anti-convergence*

```
Create and initialise an n_x-dimensional swarm, S;
A percentage of the particles have their charge set to a non-zero value
to identify them as quantum particles, and the rest of the particles
have their charge set to zero (neutral particles);
repeat:
    for each particle i = 1, ..., S.n_s do
        // set the pbest position based on the current fitness at the pbest's position
        if f(S.x_i) > f(S.y_i) then
            S.y_i = S.x_i;
        end
        // set the nbest position
        if f(S.y_i) > f(S.ŷ_i) then
            S.ŷ_i = S.y_i;
        end
    end
    for each particle i = 1, ..., S.n_s do
        update the velocity using equation (3.3);
        update the position using equation (4.4);
    end
until stopping condition is true;
```

Algorithm 4.3: QSO

operator.

Exclusion prevents multiple sub-swarms from clustering around a single peak, the sub-swarms have an *exclusion radius*, r_{excl}, centred on their global best particle. If a sub-swarm enters the exclusion radius of another sub-swarm, the weaker sub-swarm, i.e. the sub-swarm whose best particle has the lower fitness, is reinitialised. The best particle of a sub-swarm is referred to as *swarm attractor* and denoted $\mathbf{p}$.

Multi-swarm convergence occurs when all sub-swarms have converged on different peaks. In such a state, multi-swarm algorithm cannot detect the appearance of new

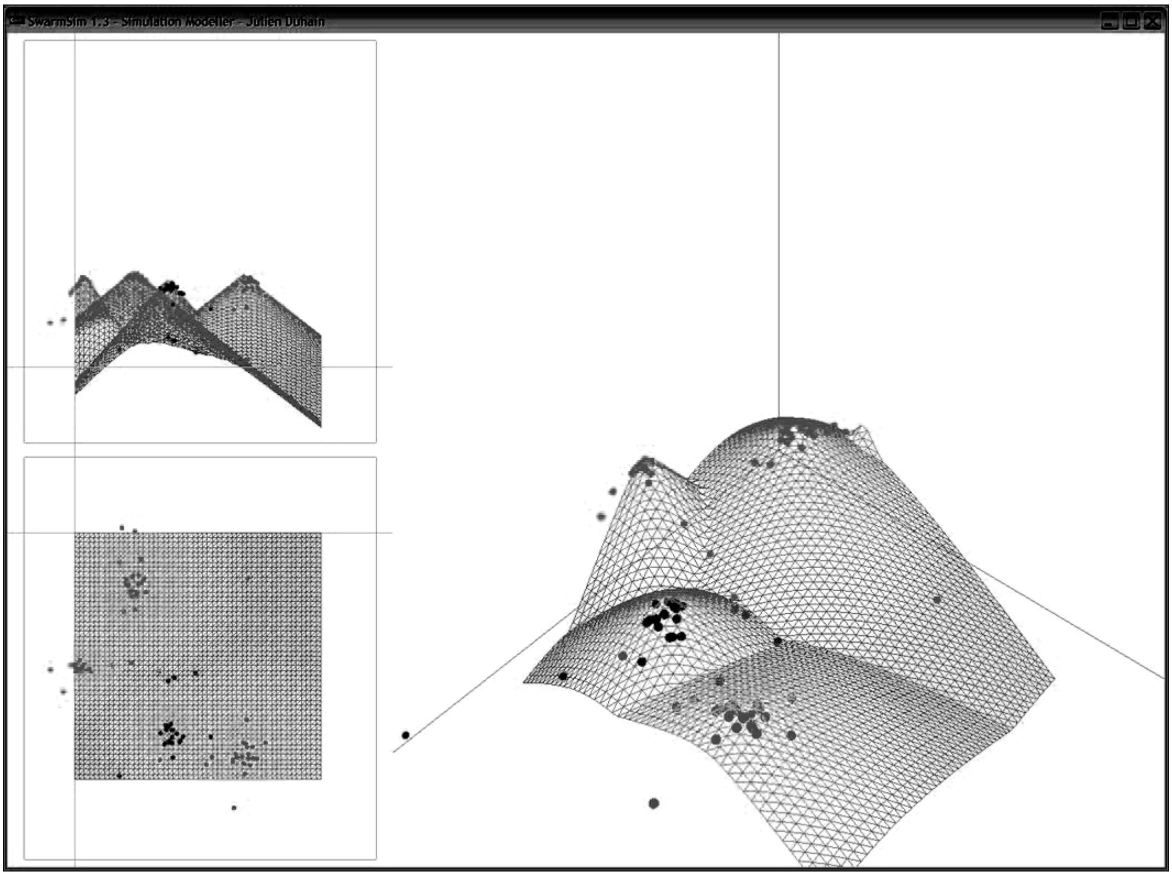


Figure 4.6: Multi-swarm with four QSO sub-swarms in a progressively changing environment

peaks. **Anti-convergence** guarantees that the multi-swarm algorithm keeps its peak-detection capabilities by reinitialising the weakest swarm when all sub-swarms have converged. The *diameter* of a swarm is the maximum distance between two particles [12]. In this context, the *radius* of the swarm is half that distance. A sub-swarm is considered converged if its radius is smaller than the convergence radius, r_{conv}. If the sub-swarms make use of CPSO/APSO or QSO, the radius of a sub-swarm is calculated using only the neutral particles of that sub-swarm as the charged or quantum particle cannot, by design, converge.

The multi-swarm algorithm is summarised in algorithm 4.4. To allow a sub-swarm to track a moving peak, diversity must be kept within the sub-swarm. CPSO/APSO or

QSO are therefore used as sub-swarms because of their capacity to maintain diversity. Blackwell and Branke [11] have demonstrated the superiority of multi-swarm that use QSO as sub-swarms over multi-swarm using CPSO's or APSO's. Choosing the optimal number of sub-swarms requires knowledge of the number of peaks that are present in the environment. If the number of peaks varies with time, selecting the optimal number of sub-swarms can be difficult.

```
Create and initialise C n_x-dimensional swarms, S;

repeat:
    if all swarms have converged then
        Reinitialise the worst swarm
    end
    for each sub-swarm, S_k do
        Perform one iteration of sub-swarm S_k using the corresponding algorithm
    end
    for each sub-swarm S_k do
        for each sub-swarm S_l ≠ S_k do
            if swarm attractor p_k is within r_excl of p_l then
                if f(p_k) ≤ f(p_l)
                    Reinitialise S_k
                else
                    Reinitialise S_l
                end
            end
        end
    end
until stopping condition is true;
```

Algorithm 4.4: Multi-swarm

4.8.5 Self-adapting Multi-swarm

To avoid selecting a maximum number of sub-swarms for the multi-swarm algorithm, Blackwell and Branke [8] developed SAMS which regulates the number of sub-swarms by itself. The SAMS algorithm is similar to the multi-swarm algorithm except for the following:

- A new sub-swarm is created and initialised in the search space when all sub-swarms have converged. There is therefore no need for the anti-convergence operator since the new swarm can detect appearing peaks.
- Instead of a fixed convergence radius, the convergence radius is calculated as follows:
$$r(t) = \frac{X}{2C^{1/n_x}} \tag{4.5}$$
where X is the length of the domain range, C is the current number of sub-swarms and n_x is the number of dimensions of the search space. The *dynamic convergence radius* $r(t)$ replaces both r_{excl} and r_{conv} in the multi-swarm algorithm, and therefore determines both convergence and exclusion.
- If the diameter of a sub-swarm is smaller than $2r(t)$, the sub-swarm has converged. The SAMS algorithm introduces an extra parameter n_{excess}, which is the maximum number of *free* (i.e. not converged) sub-swarms allowed. If the number of free sub-swarms, C_{free}, exceeds n_{excess}, the worst free sub-swarm is removed. SAMS therefore eliminates two control parameters from the multi-swarm algorithm (r_{excl} and r_{conv}) but introduces a new parameter (n_{excess}).

The number of sub-swarms at any iteration t is given by

$$\begin{aligned} C(0) &= 1 \\ C(t) &= \begin{cases} C(t-1)+1 & \text{if } C_{free} = 0 \\ C(t-1)-1 & \text{if } C_{free} > n_{excess} \end{cases} \end{aligned} \tag{4.6}$$

Blackwell [8] argues that n_{excess} should not be set to 1 for it can cause repetitive addition and removal of a sub-swarm and waste function evaluations. A very large value for n_{excess} provides good results when the number of peaks is large, but may cause the generation of too many un-removable free sub-swarms as the convergence criterion can on occasion count a sub-swarm as converged even though the sub-swarm is not associated with a peak [8].

Since the number of particles in a sub-swarm is fixed, the number of particles in SAMS grows as more sub-swarms are created. The algorithm can therefore become computationally expensive if a large number of sub-swarms are present simultaneously. SAMS algorithm is summarised in algorithm 4.5.

4.9 Summary

This chapter presented the limitations of the original PSO algorithm when applied to DEs, namely outdated memory and diversity loss. A number of approaches to reduce or eliminate the effect of these limitations have been discussed. First, the optimisation of certain parameters of the standard PSO (section 4.3), then approaches that can be taken to modify the PSO algorithm (section 4.4 to 4.7), and eventually variations of the PSO algorithm (section 4.8) that make use these approaches have been described. These algorithms are empirically evaluated in the experimental part of this work, and the next chapter inspects ways to measure their performance in DEs.

Create and initialise one n_x-dimensional swarm, S;

repeat:
    if all swarms have converged **then**
        Create and initialise a new n_x-dimensional sub-swarm
    else if $C_{free} > n_{excess}$ **then**
        remove the worst free sub-swarm
    end
    for *each sub-swarm* S_k **do**
        Perform one iteration of sub-swarm S_k using the corresponding algorithm
    end
    for *each sub-swarm* S_k **do**
        for *each sub-swarm* $S_l \neq S_k$ **do**
            if swarm attractor $\mathbf{p_k}$ is within r of $\mathbf{p_l}$ **then**
                if $f(\mathbf{p_k}) \leq f(\mathbf{p_l})$
                    Reinitialise S_k
                else
                    Reinitialise S_l
                end
            end
        end
    end
until stopping condition is true;

Algorithm 4.5: SAMS

Chapter 5

Performance Measures for Dynamic Environments

> *"Trying to improve something when you don't have a means of measurement and performance standards is like setting out on a cross-country trip in a car without a fuel gauge. You can make calculated guesses and assumptions based on experience and observations, but without hard data, conclusions are based on insufficient evidence."*
>
> – Mikel Harry

A number of swarm algorithms have been discussed in the previous chapters. To allow empirical evaluation of these algorithms, the way in which the algorithms' performance is measured must be defined. This chapter presents performance measures tailor made for population-based algorithms applied to dynamically changing environments.

5.1 Introduction

Although performance measures are well defined for population-based algorithms solving static problems [29], it is not the case for algorithms working in DEs. Several measures are in use but there is no universal agreement on how the efficiency of dynamic algorithms has to be evaluated. This chapter outlines some of the limitations of existing measures and contains a proposal for a new approach to quantify the performance of algorithms in DEs. It is worth noting that some of these measures have been used in GAs, but are equally applicable to PSO.

Section 5.2 discusses the limitations of the performance measures used to evaluate swarm algorithms applied to static problems, section 5.3 describes the qualities required from algorithms applied to dynamic problems and critically discusses the measures that have been used to evaluate such algorithms, section 5.4 describes guidelines for the selection of performance measures that evaluate PSO algorithms applied to DEs and section 5.5 discusses the selection of performance measures that can be used to assess the qualities required from swarm algorithms applied to DEs.

5.2 Limitation of Classic Performance Measures

In static environments, the qualitiy of a standard PSO can be measured in terms of the following [29]:

- **Accuracy**, i.e. the quality of the best solution obtained at the end of a simulation;
- **Reliability**, i.e. the percentage of simulations that reach a specified accuracy;
- **Robustness**, also referred to as stability, i.e. the variance in performance over a number of simulations which indicates how reliable the algorithm is at providing a given performance level; and

- **Efficiency**, i.e. the number of fitness evaluations needed to reach a specified accuracy.

Unfortunately, these measures only take into account the values obtained at the end of the simulations. In DEs, performance has to be assessed over a time period, and therefore these measures are insufficient to quantify the performance level of algorithms applied to DEs. Algorithms working in DEs must not only be able to *find* the optimum, but also to *keep track* of the known optimum and to *detect* new optima as they appear. The performance of the algorithm must therefore be measured during the entire simulation to represent the performance through all the environmental states.

5.3 Existing Measures for Dynamic Environments

To design an adequate approach to performance measurement, it is important to know what qualities of the algorithm are to be assessed. Weicker [102] suggested evaluating algorithms applied to dynamic problems based on the following:

- *Accuracy*, i.e. how close to the optimum the best particle is on average;
- *Stability*, i.e. how much the current best solution's accuracy decreases on average after a change; and
- *Recovery time* or *reactivity*, i.e. how long does the algorithm take on average to recover to an acceptable accuracy after a change occurred.

These three characteristics give a reasonable indication of the algorithm's capacity to detect and keep track of peaks since the undetected appearance of a new optimum and the loss of the current optimum are reflected by a drop in accuracy.

In the context of DEs (and for the remainder of this work), stability indicates the *robustness* of the solutions found, that is, the capacity of a solution to remain good after an environmental change. For instance, solutions located on a high plateau have a better chance to keep a good fitness after a change (hence a higher stability) than solutions on a thin peak. Figure 5.1 illustrates the concept of stability. In this figure, solution B is more stable than solution A because, after the change, the drop in B's fitness is smaller than the drop in A's fitness. After optimisation has stopped, solutions such as B can potentially keep a higher fitness for longer periods of time. In real-world situations where there are costs associated with adaptation, the algorithm aims to find these solutions that are likely to remain good after an environmental change [15]. In addition to the accuracy, stability and reactivity, it can be of interest to look at the *exploitation capacity*, which is the maximum accuracy the algorithm can reach, on average, before the optimum is modified. Indeed, in an environment where the algorithm is given a certain amount of time to produce a solution, algorithms producing the best solution at the end of the period may be preferable to algorithms providing a more accurate solution on average.

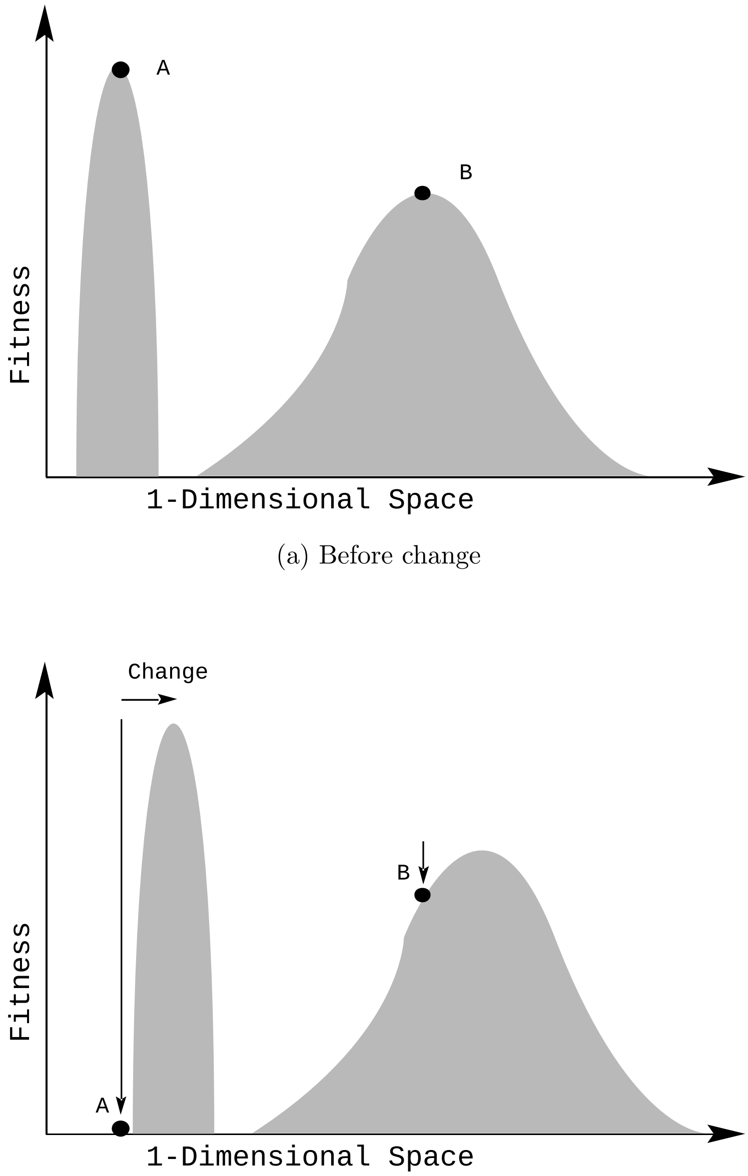
In order to evaluate the accuracy, stability, reactivity and exploitation capacity of an algorithm, several performance measures have been used. These measures are critically discussed in the following list:

- The *online performance* is the average of all fitness evaluations of all particles over the entire simulation [15]. Formally, online performance is defined as

$$online\ performance = \frac{1}{n_e}\sum_{v=1}^{n_e} e_v \tag{5.1}$$

  where e_v is the v^{th} fitness evaluation and n_e is the number of evaluations considered. This performance measure provides little information about the best values found, which are the only important values for a practical implementation of an algorithm [62]. Furthermore, in DEs diversity is essential but a diverse swarm is likely to have a number of particles with poor fitness and therefore have a low online performance.

- The average over all iterations of the Euclidian distance between the best solution and the optimum [103], called the *mean tracking error* [63] when taken over a large



(a) Before change

(b) After change

Figure 5.1: Illustration of stability in DEs

number of simulations, is formally defined as

$$mean\ tracking\ error = \frac{\sum_{m=1}^{n_m}\left(\frac{\sum_{t=1}^{n_t}\sqrt{\sum_{j=1}^{n_x}(gbest_j(t,m)-max_j(t,m))^2}}{n_t}\right)}{n_m} \tag{5.2}$$

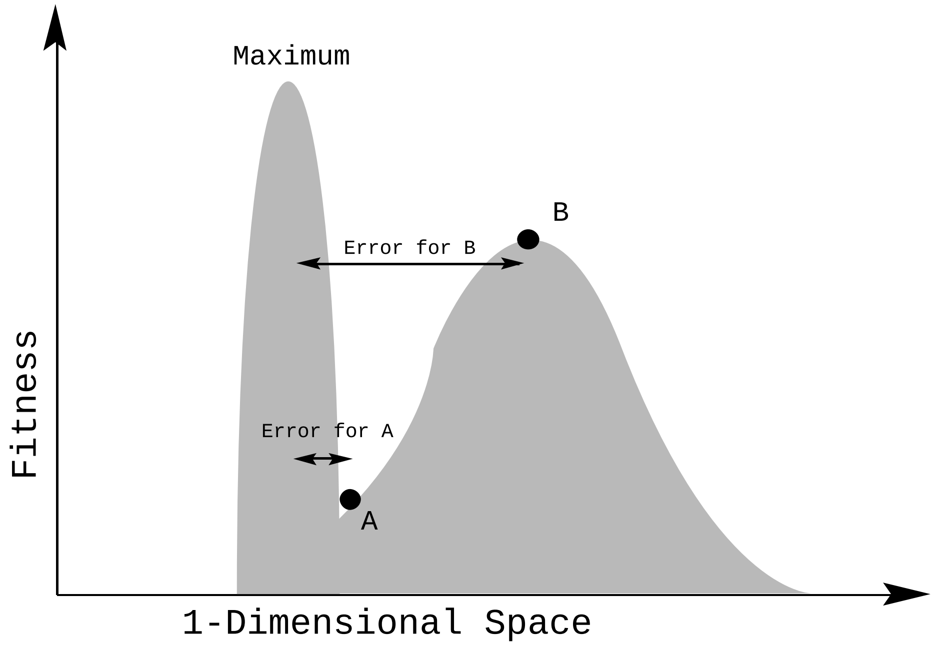


Figure 5.2: Limitations of mean tracking error

where $gbest_j(t,m)$ is the position in dimension j of the best particle at iteration t in simulation m, $max_j(t,m)$ is the position of the global optimum in dimension j at iteration t in simulation m, n_t is the maximum number of iterations, n_m is the maximum number of simulations and n_x is the number of dimensions.

The mean tracking error requires knowledge of the exact position of the optimum at all times – information that may not be available. Also, this approach can make a bad solution "geographically" close the optimum look good as only the distance from the optimum is measured while the fitness of the solution is ignored. Figure 5.2 illustrates the limitations of the mean tracking error: solution A has poor quality but is close to the optimum, while solution B, which has better quality, shows a larger error.

- Weicker measured accuracy for a small window of iterations by taking the difference between the best and the worst fitness of an iteration over the difference of the best and the worst fitness in the window [102] or, formally,

$$window\ accuracy(t) = \max\left\{\frac{f(gbest(t)) - window\_worst}{window\_best - window\_worst}\right\} \tag{5.3}$$

with

$$window\_best = \max\{f(gbest(t'))|t - W \leq t' \leq t\} \tag{5.4}$$

$$window\_worst = \min\{f(gworst(t'))|t - W \leq t' \leq t\} \tag{5.5}$$

where $gbest(t)$ is the best particle of the swarm at iteration t, $gworst(t)$ is the worst particle of the swarm at iteration t and W is the window size in number of iterations.

This technique offers an alternative to the average error when the optimum's value is unknown but is based on the assumption that the best fitness value does not change much within a small number of iterations. This assumption is correct only in progressively changing environments. In an abruptly or chaotically changing environment the optimum's value can be heavily modified from one iteration to the next.

Figure 5.3 illustrates what happens when Weicker's measure is used on an algorithm applied to an abruptly changing environment. As the best fitness increases, the window accuracy is equal to one because the current best is equal to the window's best. When a change occurs, the window-accuracy drops to zero since the window now contains both the high fitness from before the change and the low fitness from after the change. After a number of iterations, equal to the window size, the high fitness from before the change is excluded from the window and the window-accuracy becomes once again equal to one. In this context, the window-accuracy fails to provide information on the accuracy of the solutions found by the algorithm.

- The *modified offline performance* takes the average of the best values found so far at each evaluation, with a reset of the best-so-far after a change occurs [15]. This technique provides information about how quickly the algorithm adapts to a change but requires knowledge of when a change occurs. It also requires consideration of the progression of the best-so-far value through the simulation instead of providing

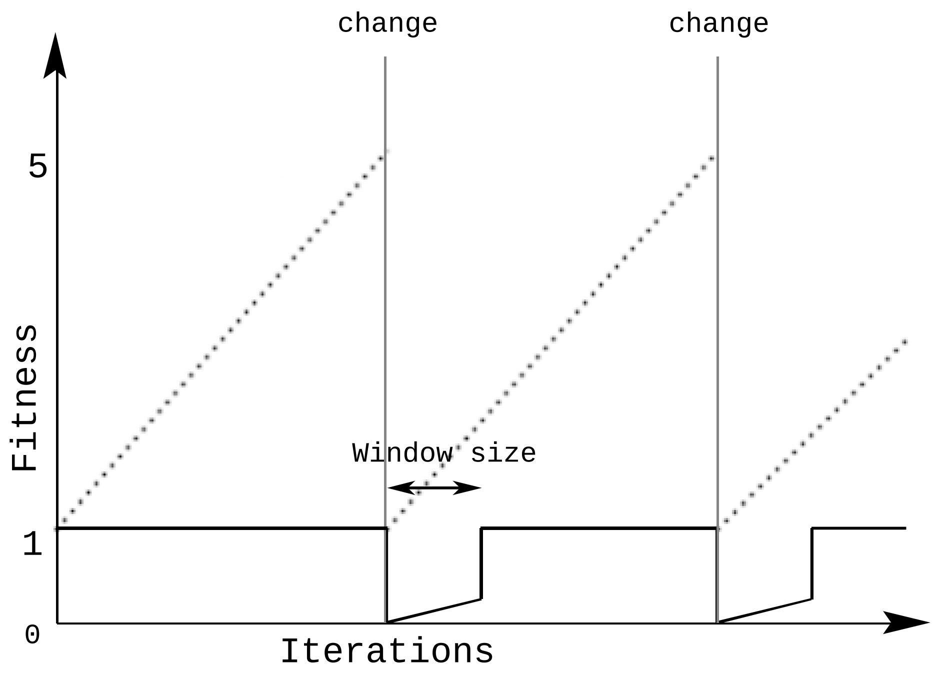


Figure 5.3: Limitation of the window-accuracy for an abruptly changing environment. The black line sketches the window-accuracy over a number of iterations, while the dotted line sketches the progression of the best fitness over the same period of time.

a single average value. Alternatively, the values obtained before every change can be averaged to obtain a single value that represents the performance of the algorithm, or formally

$$offline\ performance = \frac{1}{n_e}\sum_{v=1}^{n_e} e_v^* \tag{5.6}$$

with

$$e_v^* = max\{e_c, e_{c+1}, e_{c+2}, \ldots, e_v\} \tag{5.7}$$

where c is the iteration during which the last change occurred.

- Bird and Li [5] proposed the *best known peak error* (BKPE) to measure the convergence speed of an algorithm after a peak has been found. BKPE is computed in a similar fashion to the modified offline performance except that the offline error is computed per iteration and for each peak individually. The offline error of a peak is calculated using only the particles located on this peak and the peak's optimal value. Just before a change occurs, the peaks that have not been covered by at least one particle every iteration since the last change are discarded. The offline error of the peak with the best particle among the peaks left is then added to the total. At the end of the simulation, the total is averaged per iteration. A full description of the BKPE can be found in [5]. Although this measure gives insight into the exploitation capacity of the algorithm, the measurements can be inaccurate if few or no peaks are constantly covered by particles.

- Trojanowski and Michalewicz [91] measured the difference between the optimum value and the best solution's fitness just before a change. Using this measure entails knowing when the changes occur, as well as the value of the optimum, but offers some information on the exploitation capacity of the algorithm no other measure can provide. The *average best error before change* (ABEBC) is formally described as

$$ABEBC = \frac{1}{n_k}\sum_{c=0}^{n_k}(err_{c,r-1}) \tag{5.8}$$

where r is the number of iterations between two environmental changes, n_k is the number of environmental changes, $err_{c,t}$ is the difference between the best fitness and the optimal fitness at iteration t after the last change c. Change zero marks the beginning of the simulation. ABEBC is described further in section 5.5.

- One of the most commonly used measures is the best-of-generation average [2, 34, 36] which is the average over all iterations of the best solution of each iteration. Morrison [62] criticized this approach for its inability to measure the statistical significance of the result and to compare performance across the full range of landscape dynamics. He proposed the *collective mean fitness* (CMF) which is a best-of-average exposed to a representative sample of all possible landscape dynamics, formally defined as

$$CMF = \frac{\sum_{m=1}^{n_m}\left(\frac{\sum_{t=1}^{n_t} f(gbest(t,m))}{n_t}\right)}{n_m} \tag{5.9}$$

where $f(gbest(t,m))$ is the fitness of the best particle at iteration t of simulation m.

The main disadvantage of the CMF is the large number of simulations required.

5.4 Guidelines for Performance Measures

When designing performance measures for algorithms applied to changing environments, certain guidelines should be considered. These guidelines are listed below:

1. First of all, the measures should always be an **average** of a collection of values sampled during the whole simulation and not a single value. There may be problems where fitness of a particle can be positive or negative. For such cases, one must take the absolute value of the fitnesses to make sure the elements of the average do not cancel each other out.

2. If the value of the optimum is known at all times, this information can be used to calculate the **error**, that is the difference between the fitness of the best particle in the swarm and the value of the optimum. Performance measures using the error are indeed always preferable to those using the fitness since, for every problem, the optimal error is always zero while the optimal fitness value varies with the value of the optimum. Using the error therefore allows for better comparison between algorithms working on different problems.

3. If the optimum's value is unknown, the error cannot be calculated and the **fitness** of the best particle must therefore be used instead of the error. If the number of iterations and simulations are high enough and if the algorithm is exposed to a sufficient number of changes, the average fitness should permit a suitable comparison between different algorithms as long as they work on identical problem instances.

5.5 Selection of Performance Measures

Based on the different approaches described in section 5.3, it becomes apparent that the performance measure to use depends on *the intent of the practitioner*, *the amount of information available about the optimum* and, to a lesser extent, on *the class of DEs tested*. Indeed, the more information available about the landscape dynamics, the more accurate the assessment can be. Depending on what the objective of the algorithm is, a single measure can sometimes be enough. However, for a complete evaluation of an algorithm's capacity to work in a changing environment, several performance measures may be needed in order to assess the accuracy, stability, reactivity and exploitation capacity of the algorithm.

Different performance measures designed to evaluate specific qualities of an algorithm are listed below. Some of the methods from section 5.3 that provide useful information are reintroduced here:

- The **collective mean fitness** defined in equation (5.9) [62] (or **collective mean error** (CME)) is typically a useful performance measure. It provides measurements similar to the averaged offline error and gives a good general idea of the stability, reactivity, detection capacity, tracking capacity and accuracy of an algorithm since a drop in any one of these characteristics translates into a drop in CMF (or a rise in CME).
- In a progressively or chaotically changing environment, the high frequency of change causes the best fitness before a change to be very close or equal to the best fitness after a change, which means that the exploitation capacity of an algorithm is hardly distinguishable from its stability. In such environments, the CME/CMF is therefore the only measure needed. However, in an abruptly changing environment, the *exploitation capacity* can be measured using the **average best error before change** (ABEBC) defined in equation (5.8) or **average best fitness before change** (ABFBC)) (if the optimum value is unknown). Figure 5.4 represents the change in error value for algorithms ALG1 and ALG2 over a sample

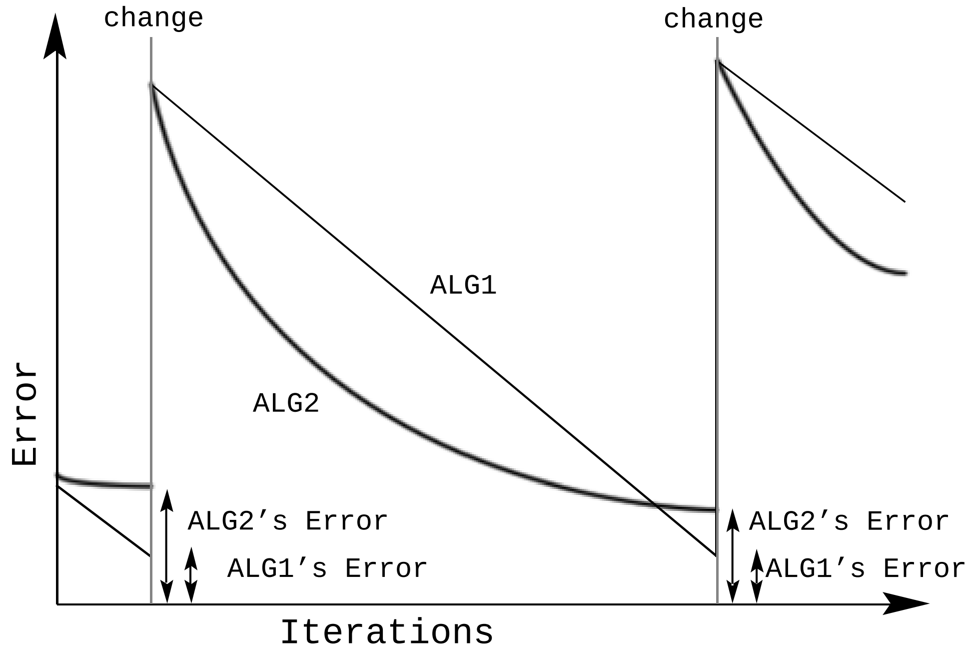


Figure 5.4: ABEBC.

of iterations (the sample does not start at the beginning of the simulation). Algorithm ALG1 has a better average error than algorithm ALG2, but ALG1 continues to approach the optimum until a change occurs, while ALG2 stagnates at a certain accuracy level. If the objective of the problem is to find the best possible solution within a given time (the time between two environmental changes), then algorithm ALG1 performs better.

- To measure the *stability* of an algorithm in an abruptly changing environment more accurately, the **average best error after change** (ABEAC) or the **average best fitness after change** (ABFAC) can be used. These measures average the best values (fitness or error) after a change and therefore promotes algorithms that find more stable solutions. The ABEAC is formally defined as

$$ABEAC = \frac{1}{K}\sum_{c=1}^{K}(err_{c,0}) \qquad (5.10)$$

Measuring the difference between the error/fitness before and after a change would make solutions located on low plateaus look good and is therefore an unreliable measure. As illustrated in figure 5.5, the difference in fitness before and after the change is smaller for solution B than for solution A. Although, solution A is preferable to solution B even after the environment has changed.

- The *recovery time* is only applicable in abruptly changing environments as only a low change frequency can allow the algorithm to "recover". In such environments the **average number of iterations needed to reach an acceptable error after a change**, or **iterations to error limit** (ITEL), can be used to measure the average recovery time of an algorithm as illustrated in figure 5.6. ITEL is formally defined as

$$ITEL = \frac{1}{n_k} \sum_{c=1}^{n_k} \iota \tag{5.11}$$

where

$$\iota = \begin{cases} \min_{\forall t, err_{c,t} < \tau}\{t\} & \text{if } \exists \; err_{c,t} \; < \tau \\ r & \text{otherwise} \end{cases} \tag{5.12}$$

where r is the number of iterations between two changes, τ is the target error the algorithm must reach, and $err_{c,t}$ is the difference between the best fitness and the optimal fitness at iteration t after the last change c. The value of the optimum must be known at all times to calculate the error by taking the difference between the optimum and the best particle's fitness. Once an acceptable error level has been defined for the problem, it is possible to count how many iterations – or function evaluations – are needed to reach an acceptable error after an environmental change. Knowledge about the time of change is therefore also required. The reactivity cannot be measured without information about the optimum's value since it is not possible to define an acceptable fitness level without knowing if such fitness is actually reachable within the search space. Figure 5.7 shows what can happen when a fitness threshold is used: after *change* 2, the recovery time is equal to zero because the fitness of the best individual (black line) does not drop below the threshold (dotted line), although the algorithm never becomes close to the optimum (thick line). After *change* 3, the algorithm performs well, but because the optimum is lower than the threshold, the best fitness never reaches an acceptable level.

- Although an algorithm's capacity to detect a new optimum is shown to some extent by the CME, the **average diversity** of the population throughout the simulation can give an extra indication of the *detection capacity* of the algorithm: the larger the diversity of the population, the higher the probability of detecting the appearance of a new peak. Diversity can be measured by taking the largest distance between any pair of particles as follows:

$$average\ diversity = \frac{1}{n_t}\sum_{t=1}^{n_t} \max_{(i\neq o)\in[1,n_s]} \left\{ \sqrt{\sum_{j=1}^{n_x}(x_{ij}-x_{oj})^2} \right\} \tag{5.13}$$

where x_{ij} is the position of particle i in dimension j, n_x is the dimensionality of the problem and n_s is the size of the swarm. Alternative ways to calculate the diversity can be found in [67]. Also, the diversity can be measured using **peak cover** [5, 15]. Peak cover is the ratio of the number of peaks covered with at least one particle over the total number of peaks in the environment. Using only average diversity and/or peak cover is not enough to determine the performance of an algorithm, but can provide useful complementary information.

- In the case of an abruptly changing environment with known change time, the **average diversity level after change** (ADAC) can be calculated instead of a general average of the diversity. ADAC is formally defined as

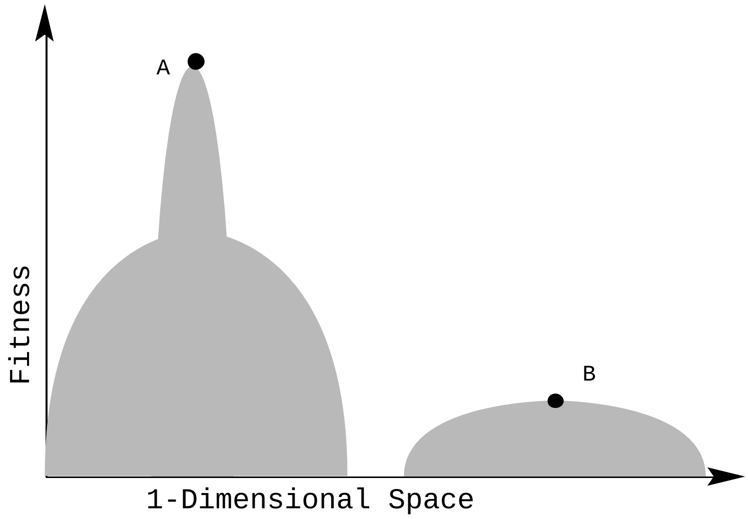
$$ADAC = \frac{1}{n_k}\sum_{t=1}^{n_k} \max_{(i\neq o)\in[1,n_x]} \left\{ sqrt\sum_{j=1}^{n_x}(x_{ij}-x_{oj})^2 \right\} \tag{5.14}$$

where n_k is the number of environmental changes. Indeed, the only time a peak can appear is right after a change and the diversity of the swarm is therefore relevant only at that specific time. However, this measure is only applicable to algorithms that artificially reintroduce diversity when a change is detected. For cases where the diversity level of the swarm increases progressively after a change, an average of the maximum diversity levels achieved between changes can be used.

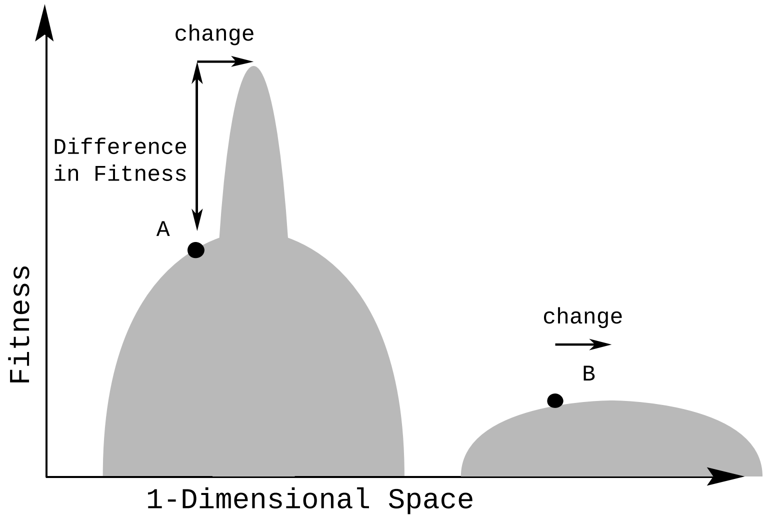
This section's list can be thought of as a toolbox containing the various measurement tools that can be used depending on the purpose of the algorithm and the kind of dynamic problem the algorithm is applied to. The CME (or CMF) can always be used to evaluate how the algorithm performs in the DE. For progressive environments of type II and III the average diversity can also be used to measure the algorithm's detection capacity. All the other measures in the above list are applicable solely to environments with a relatively low change frequency, but allow an evaluation of the stability, reactivity, and exploitation capacity of the algorithm. The evaluation of these qualities is not possible using the measures discussed in section 5.3.

5.6 Summary

After studying the existing performance measures for evolutionary and swarm based algorithms applied to dynamically changing environments and identifying the qualities such algorithm should possess, evaluation guidelines and specific performance measures have been discussed. The background and theory part of this work is now complete. The experimental setup and result analysis part of the thesis begins in the next chapter with a description of the experimental procedure.



(a) Before change



(b) After change

Figure 5.5: Illustration of how the fitness difference before and after a change as measure of stability can be misleading.

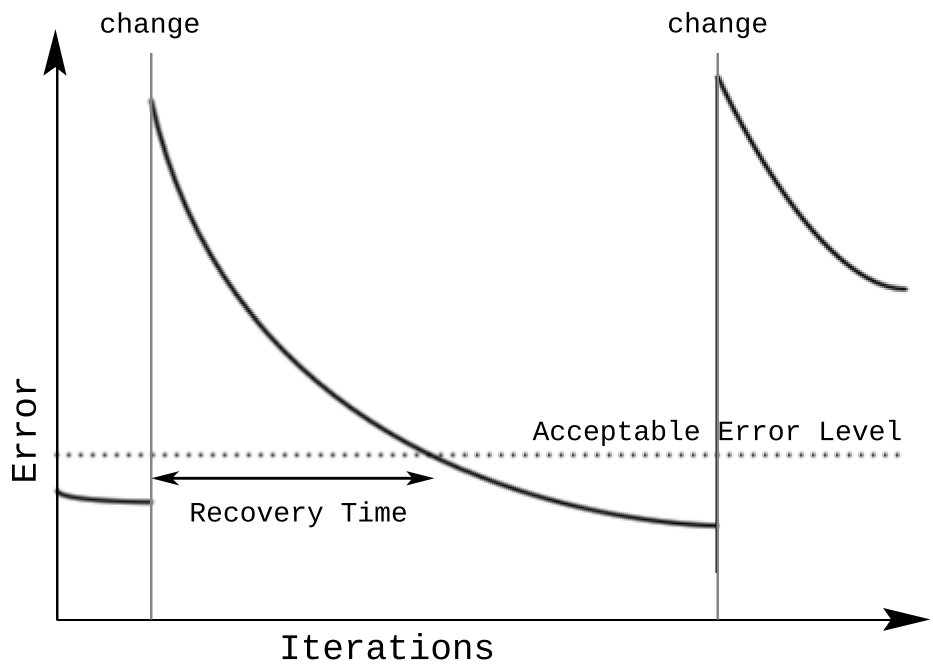


Figure 5.6: Average number of iterations needed to reach an acceptable error after a change.

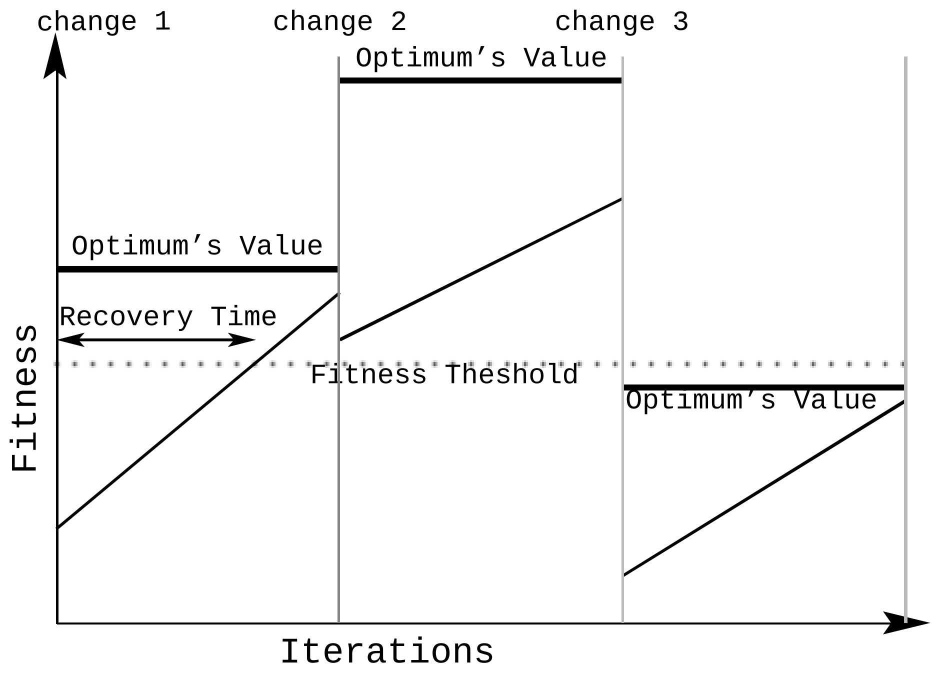


Figure 5.7: Limitation when using a fitness threshold to measure the reactivity.

Part II

Experimental Setup and Results Analysis

"Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning."

– Sir Winston Churchill

Chapter 6

Experimental Procedure

> *"Not everything that can be counted counts, and not everything that counts can be counted."*
>
> – Albert Einstein

The experimental part of this work begins with the definition of the experimental procedure used to evaluate the algorithms. To obtain meaningful results, it is necessary to narrow down a potentially limitless number of experiments to a representative panel of test cases. This chapter selects the problems, performance measures and settings common to all algorithms to allow the isolation of the main features of the algorithms and environments.

6.1 Introduction

The aim of this thesis is to provide a better understanding of the strengths and weaknesses of the various swarm algorithms and to investigate how the different approaches taken by the PSO algorithms from section 4.8 influence the performance of PSO. To reach this goal, standard values are selected for the parameters of the original PSO model which are shared by all algorithms. Having a common base for all modified PSO algorithms permits isolation of the effect that a particular approach has on performance and makes the algorithms easier to compare to each other. Similarly, the environments used to test the algorithms share a number of characteristics, but each environment presents a unique feature. The influence of that feature on the performance of the various algorithms can then be studied.

Section 6.2 describes the various problems the algorithms are applied to. Section 6.3 defines the experimental method and specifies which performance measures are used to evaluate the algorithm, and section 6.4 selects standard values for the parameters that are common to all algorithms.

6.2 Description of the Test Environments

Section 2.3 listed numerous types of DEs and the many ways in which dynamic problems can differ from one another. From the multitude of potential test cases, five environments with specific characteristics are selected to assess certain qualities of the algorithms. All environment test cases are created using the MPB described is section 2.5. The domain of the MPB is set to [0, 100]. The peaks never go outside the domain of the search space but particles are free to leave the domain if their velocity leads them to do so. Boundary constraints are deliberately not enforced because the repositioning of the particles inside the domain can improve the peak detection capacity of the swarm. Since the aim of this work is also to investigate the weaknesses of the algorithms, external factors that influence the performances – such as repositioning of particles that leave the domain – should be eliminated.

To study the effect that modality and dimensionality have on the performance of the algorithms, instances of the five environment test cases are created with 1, 5 and 15 peaks, each of these in 2, 5, 10, 30 and 50 dimensions. This means that for each set of parameters, an algorithm is tested against 75 test environments.

Ideally, environment test cases should be created for all combinations of Eberhart *et al.*'s types, Angeline's types, a range of values for temporal severity, and a range of values for spatial severity. However, evaluating the algorithms on such a large number of test environments would require an astronomical number of computer simulations and is not possible in practice since every configuration of each algorithm would have to be evaluated for each of these test cases. Therefore, a subset of test cases has to be selected. The test environments are selected as follows:

- E-STATIC – A static environment:

  To measure how much the dynamic nature of an environment influences performance, the algorithms are first tested on a static environment. Fifteen instances of this environment varying in modality and dimensionality are created. The peaks are initialised randomly at the beginning of the simulation (once per environment instance), but are never changed in position or value. Evaluation of the algorithms for this environment indicates how close to the optimum an algorithm can get if given 1000 iterations to optimise the function. A static environment is obtained using a $height\_severity$ of 0 and $width\_severity$ of 0 in equation (2.16), and $s = 0$ in equation (2.18).

- E-PROGRESS – A progressively changing environment:

  This environment test case is classified as a progressive, random, type III environment according to the classification system of section 2.4.2. The 15 instances of E-PROGRESS environments are used to evaluate the tracking capacity of the algorithms. However, because this environment is of type III, an algorithm with bad detection capacity that tracks only one peak should show a lower performance level for the multimodal environments when a different peak becomes the global best. For such an environment, the changes occur at every iteration with a low spatial severity obtained using a $height\_severity$ of 1 and a $width\_severity$ of 0.05 in equation (2.16), and $s = 1$ in equation (2.18).

- E-ABRUPT – An abruptly changing environment:

  To assess the peak detection capacity of the algorithms, all algorithms are evaluated on the 15 instances of an abrupt, random, type III environment. In this environment, peaks are relocated to a distant location after every change making peak tracking difficult and giving an advantage to algorithms that are good at peak detection. Furthermore, when peaks are close to each other, a lower peak can be hidden beneath a higher peak, which simulates the appearance and disappearance of optima. As mentioned in section 5.5, the abruptly changing nature of the environment also permits analysis of the exploitation ability, and reactivity of the algorithm. In this environment, changes occur every 200 iterations with a high spatial severity obtained using a $height\_severity$ of 10 and a $width\_severity$ of 0.05 in equation (2.16), and $s = 50$ in equation (2.18).

- E-CHAOS – A chaotically changing environment:

  The 15 instances of this environment test the algorithms in more extreme conditions than E-PROGRESS and E-ABRUPT with changes occurring every five iterations with a high spatial severity obtained using a $height\_severity$ of 10 and a $width\_severity$ of 0.05 in equation (2.16), and $s = 50$ in equation (2.18). Applying the algorithms to the E-CHAOS environments allows to observe how the performance of an algorithm that obtained good results for E-ABRUPT is affected by an increase in temporal severity in the environment. Observations can also be made on how the performance of an algorithm that obtained good results for E-PROGRESS is affected by an increase in spatial severity in the environment.

- E-PATTERN – An environment with a circular pattern in the change:

  To assess if the algorithms can make use of a pattern in the change to improve performance, the algorithms are tested on the 15 instances of a circular type III environment. In this environment, the entire search space is rotated around the centre of the domain (the pseudocode for the rotation can be found in appendix A). The speed at which a peak moves across the search space depends on the distance between this peak and the rotation centre. The function is rotated each iteration by an angle of 3.6° so that the function has completed one rotation on itself every 100 iterations. The environment is defined by equation (2.16) with a *height_severity* of 1 and a *width_severity* of 0.05; $s = 0$ in equation (2.18) since the relocation of the peaks is ensured by the rotation. The type III nature of the environment allows a fairer comparison between E-PATTERN and the other environments. However, since each rotation occurs in multiple dimensions, the spatial severity increases with increase in the dimensionality. E-PATTERN environments therefore range from progressively changing (two dimensions) to chaotically changing (50 dimensions). This increase in severity also permits to observe the effect that a gradual increase in spatial severity has on the performance.

According to the classification described in section 2.2.2, all test environments are multivariate, continuous, nonlinear, unconstrained, uni-objective problems. Test environments with one peak are unimodal problems and environments with five and 15 peaks are multimodal problems.

For all DE test cases, type III of Eberhart *et al*'s classification is selected. In type III environments, both the location and the value of the optima are subject to change which allows assessment of both the tracking capacity and the detection capacity of the algorithms. As mentioned in section 2.3.2, algorithms that can track the optimum effectively have an advantage in type I environments, and algorithms that can detect new optima effectively have an advantage in type II environments. As described above, the tracking capacity of the algorithm is evaluated in E-PROGRESS environments and the detection capacity of the algorithms is evaluated in E-ABRUPT environments. Therefore, even though none of the test environments is of type I or type II, knowledge about detection and tracking capacity of an algorithm can still give an indication about the ability of an algorithm to perform well in a type I or a type II environment.

Unless specified otherwise in its description, each environment is random according to Angeline's classification ($\lambda = 0.0$ in equation (2.16)). The effect of an environment change pattern on the performance is tested by the E-PATTERN test cases. Having a common random type III nature for all problems also allows a fairer comparison between the performances of an algorithm for the different test environments.

6.3 Evaluation Method

The evaluation of the algorithms is performed through computer simulations. A simulation is the application of a specific algorithm to a specific problem instance. All algorithms, problems and measurements are implemented using the Computational Intelligence Library (CIlib) [74]. CIlib is a collective project aimed at building a generic framework designed to accommodate the development of computational intelligence (CI) algorithms. CIlib is written in the Java 1.6 programming language. CIlib version 0.7.5 is available at `git://github.com/gpampara/cilib.git`. Appendix B provides an XML configuration for CIlib which illustrates how simulations are generated.

The algorithms are tested individually with a range of values (provided in chapter 8) for the parameters specific to each algorithm. Each algorithm is evaluated on all test environments. Each simulation lasts for 1000 iterations and is repeated 30 times on the same problem instance before the results are averaged.

Random numbers are generated using the Mersenne Twister [61], an equidistributed uniform pseudorandom number generator that can be seeded to produce the same series of random numbers repetitively. All test environments use random numbers, including the static environments since the peaks are randomly positioned at the beginning of the simulation. For fairness, the environments are seeded with the same seed (seed value: 1). This ensures that each of the 75 test environments are unique with each specific test environment undergoing the same modifications during each one of the 30 simulations. The seeding also ensures that the different algorithms are evaluated on test environments that go through the exact same states.

To measure the accuracy, tracking capacity, and detection capacity of the algorithms, the CME is used as performance measure in all test environments. For the E-ABRUPT and E-CHAOS environments, the algorithm's exploitation capacity and reactivity are also measured using respectively the ABEBC and ITEL. The ABEBC measures the exploitation capacity by calculating the average error of the best particle before a change. The ABEBC therefore measures how close to the optimal value the algorithm can get while the environment stays static. The ITEL measures the reactivity by calculating the average number of iterations needed by the algorithm to reach an error of 10 after a change. In static environments, the exploitation capacity is also measured by the average error of the best particle at the end of the simulation. The algorithms tested are exclusively seeking solutions with high fitness; the stability of the solutions found is therefore not measured.

Diversity is calculated as in [51, 100], by using

$$diversity(S(t)) = \frac{1}{n_s}\sum_{i=1}^{n_s}\sqrt{\sum_{j=i}^{n_x}(x_{ij}(t) - \bar{x}_j(t))^2} \tag{6.1}$$

where $S(t)$ is the swarm at iteration t, n_s is the swarm size, x_{ij} is the position of particle i in dimension j, and $\bar{x}_j(t)$ is the average position of the particles in the j^{th} dimension, i.e.

$$\bar{x}_j(t) = \frac{\sum_{i=1}^{n_s} x_{ij}(t)}{n_s} \tag{6.2}$$

The Mann-Whitney U test [60] is used to compare the results of two sets of simulations. The difference between two sets of results is considered statistically significant only if the p-value obtained is below 0.05.

6.4 Parameter Selection

To establish a common ground among the algorithms tested, the parameters shared by all algorithms, i.e. the parameters of the standard PSO algorithm, are set to the same values for all algorithms. The parameter values are selected so that all algorithms can function without requiring the modification of the configuration. Where possible, values are chosen so that an unfair advantage is not given to any particular algorithm. The following parameter values have been used (where no further detail is given, values as proposed in literature and summarised in sections 3.3 and 4.3 were used):

- **Velocity model**: Full model

  Although, as mentioned in section 4.3, the social-only model can outperform the full model on some dynamic problems when no re-evaluation of the *pbest* takes place [20], all the dynamic algorithms tested implement *pbest* re-evaluation and are therefore unlikely to gain any benefit from using the social-only model. Additionally, algorithms such as QSO use the *pbest* of the quantum particles to store the location of attractive solutions while the particles move to a different location. The cognitive component is therefore needed for such algorithms to function more effectively.

- **Inertia weight** : 0.729844, as proposed in literature and summarised in sections 3.3 and 4.3.

- **Velocity clamping**: none

  In dynamically changing environments, clamping the velocity slows down the swarm and limits exploration by facilitating cluster formation of particles. Figure 6.1(a) illustrates a re-evaluating PSO for a two-dimensional E-CHAOS environment with $v_{max} = 5$ at iteration 204. Because the particles cannot leap far beyond their neighbourhood best (*nbest*), the swarm has lost most of its diversity. Also, although the last change happened at iteration 200 and a new change is occurring at the next iteration, the swarm cannot move fast enough to reach the closest peak before the environment changes again. On the other hand, figure 6.1(b) illustrates a re-evaluating PSO without v_{max} for the same environment at iteration 279. A change occurs at iteration 280, but this time the swarm is covering a large area in the search space and is likely to detect the new optima.

- **Social network structure**: Von Neumann

  In addition to the advantages mentioned in section 4.3, the connectivity of the Von Neumann topology is intermediate between that of the star topology and ring topology. Although the importance of exploration in DEs has been highlighted, the quick exploitation of an optimum before a change occurs is also important. The Von Neumann social structure offers an adequate trade-off between exploration and exploitation.

- **Acceleration coefficients**: $c_1 = c_2 = 1.496180$, as proposed in literature and summarised in sections 3.3 and 4.3. $\mathbf{r_1}$ and $\mathbf{r_2}$ contain random numbers between 0 and 1 generated by a Mersenne Twister [61] (uniform distribution).

- **Swarm size**: 100 particles

  Because certain modified PSO algorithms use various types of particles or subdivide the swarm into sub-swarms, a relatively large number of particles should be used to allow subsets of particles to be made within the swarm.

- **Termination conditions**: Maximum number of iterations of 1000

  This number allows that the behaviour of the algorithm can be observed for numerous environment changes.

- **Constriction coefficient**: None

- **Neighbourhood best update strategy**: Memory based

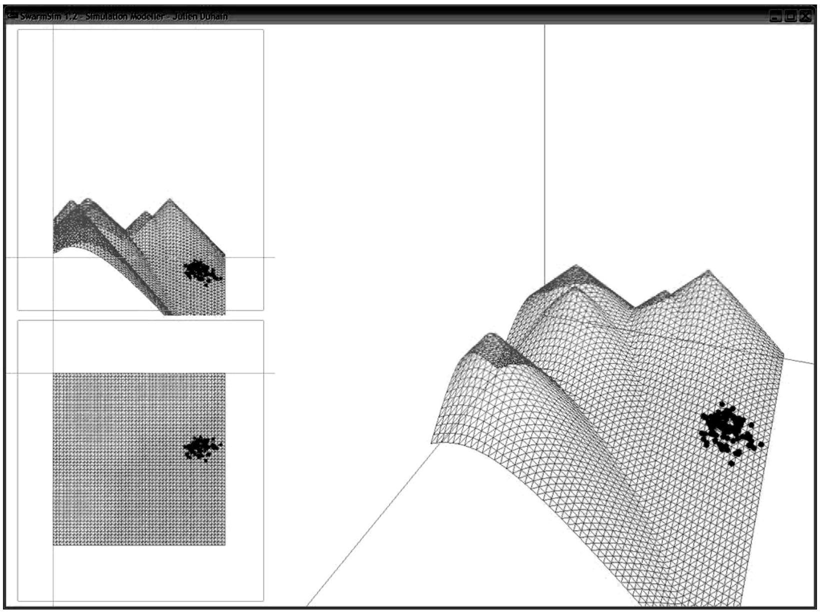
  The benefits of an iteration based update strategy has been mentioned in section 4.3. However, a memory based strategy can be used in the experiments as the *pbest* of the particles is re-evaluated by all modified PSO algorithms. Furthermore, with QSO, for instance, the *pbest* of a quantum particle stores the location of the best position visited by the particle. Therefore, the sharing of this information is only possible using the memory based neighbourhood best update strategy.

- **Update strategy**: Synchronous update

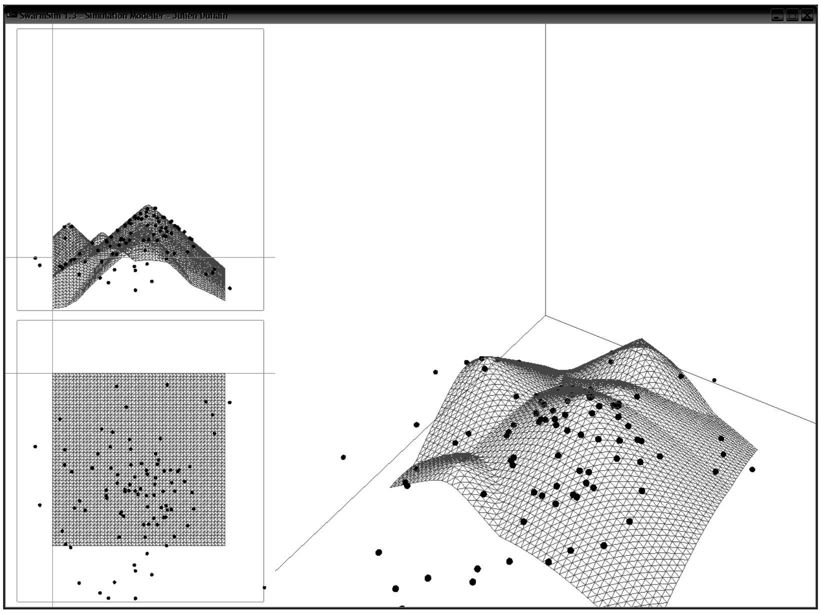
  Mussi *et al.* [64] have shown that the update strategy only influences performance when a fully connected topology is used. However, since the memory based neighbourhood best update strategy is used in the experiments in conjunction with *pbest* re-evaluation, using asynchronous update can be problematic. If asynchronous velocity update is used, the selection of the *nbest* of a particle would require the re-evaluation of the *pbest*'s of the entire neighbourhood of the particle to avoid selecting the *nbest* based on outdated information. This re-evaluation of the *pbest* neighborhood after neighborhood is inefficient and a better solution would be to re-evaluate all *pbest*'s synchronously before updating the velocity of the particles asynchronously. However, this would mean to have a synchronous update for *pbest* re-evaluation and an asynchronous update for the velocity. Therefore, since the selection of the update strategy has little effect on the performance when a non-fully connected neighbourhood topology is used, the synchronous update is selected for the sake of consistency.

6.5 Summary

This chapter has defined the experimental procedure. Five test environments have been described and values have been selected for the parameters common to all algorithms. The next chapter presents the parameters specific to each of the algorithms tested, the results of the experiments, and an analyses of these results.



(a) Swarm with $v_{max} = 5$, iteration 204



(b) Swarm with no v_{max}, iteration 279

Figure 6.1: Effect of v_{max} on re-evaluating PSO for E-CHAOS with five peaks

Chapter 7

Establishing Benchmarks

"The nice thing about standards is that there are so many of them to choose from."

– Andrew S. Tanenbaum

The purpose of this chapter is to evaluate the ability of the standard PSO and re-evaluating PSO to work in DEs. This chapter presents and analyses the results of applying the standard PSO and re-evaluating PSO to the test-environments from section 6.2.

7.1 Introduction

Before evaluating the algorithms designed for DEs, it is important to set benchmarks against which the performance level of these algorithms can be compared. This chapter evaluates the standard PSO and a PSO using re-evaluation of the *pbest* using the procedure defined in chapter 6. A number of observations are made regarding each algorithm's behaviour and performance level for the various test environments. Tables are provided to summarise the empirical results. In these tables, the first column contains the name of the environment with the measure used in brackets, and the number of peaks in the environment is indicated in the second column. *Scientific e notation* is used for values smaller than 0.1. Decimal notation (with two digits after the decimal point) is used for values greater than 0.1 to save space and improve readability. Because Java doubles are used in the code, values between -2^{-1022} and 2^{-1074} are rounded to 0.00. The confidence interval (95%) is stated along with each measurement and calculated as follows:

$$confidence\ interval = \frac{2.048 \times standard\ deviation}{\sqrt{30}} \tag{7.1}$$

where *standard deviation* is the standard deviation of a sample calculated as follows:

$$standard\ deviation = \sqrt{\sum_{m=1}^{30} \frac{(sample_m - mean)^2}{29}} \tag{7.2}$$

where $sample_m$ is the value of the measurement for simulation m and $mean$ is the average value of the 30 samples.

The Mann-Whitney U test [60] is used to compare the results of two sets of simulations. The difference between two sets of results is considered statistically significant only if the p-value obtained is below 0.05. Tables containing the p-values are provided in appendix C.

Section 7.2 presents and analyses the results obtained from applying the standard PSO to the test environments, section 7.3 presents and analyses the results obtained from applying the re-evaluating PSO to the test environments, and section 7.4 compares the performance of the standard PSO to that of the re-evaluating PSO for the various test environments.

7.2 Standard PSO

The first set of experiments evaluates the performance of the original PSO applied to the environment test cases listed in section 6.2. The aim of this section is to provide a point of reference in terms of performance level and to observe how a standard swarm behaves in the environment test cases. The standard PSO has shown to be effective at solving static problems and should therefore perform well for the static environment test cases (E-STATIC). However, because of outdated memory and diversity loss, the algorithm is likely to perform poorly for the DE test cases (E-PROGRESS, E-ABRUPT, E-CHAOS and E-PATTERN).

7.2.1 Results

The results from table 7.1 were obtained by an implementation of the PSO algorithm described in chapter 3 using the configuration from section 6.4. The code for the standard PSO can be found in CIlib [74].

7.2.2 Analysis of Results

As expected, the standard PSO performed well for the static environments, but struggled for the dynamic test cases.

For E-STATIC, the error at the end of the simulation, referred to as *Error* in table 7.1, was close to zero for all environments with 10 dimensions or less, indicating that the PSO found the global optimum. The CME increased with the number of dimensions indicating that the problems with higher dimensionality were harder to solve. For environments with 10 dimensions or less, the final error was always 0.00. The results also show that the algorithm performed consistently better for unimodal environments than for multimodal environments. In unimodal environments, the optimum can be found faster as the swarm

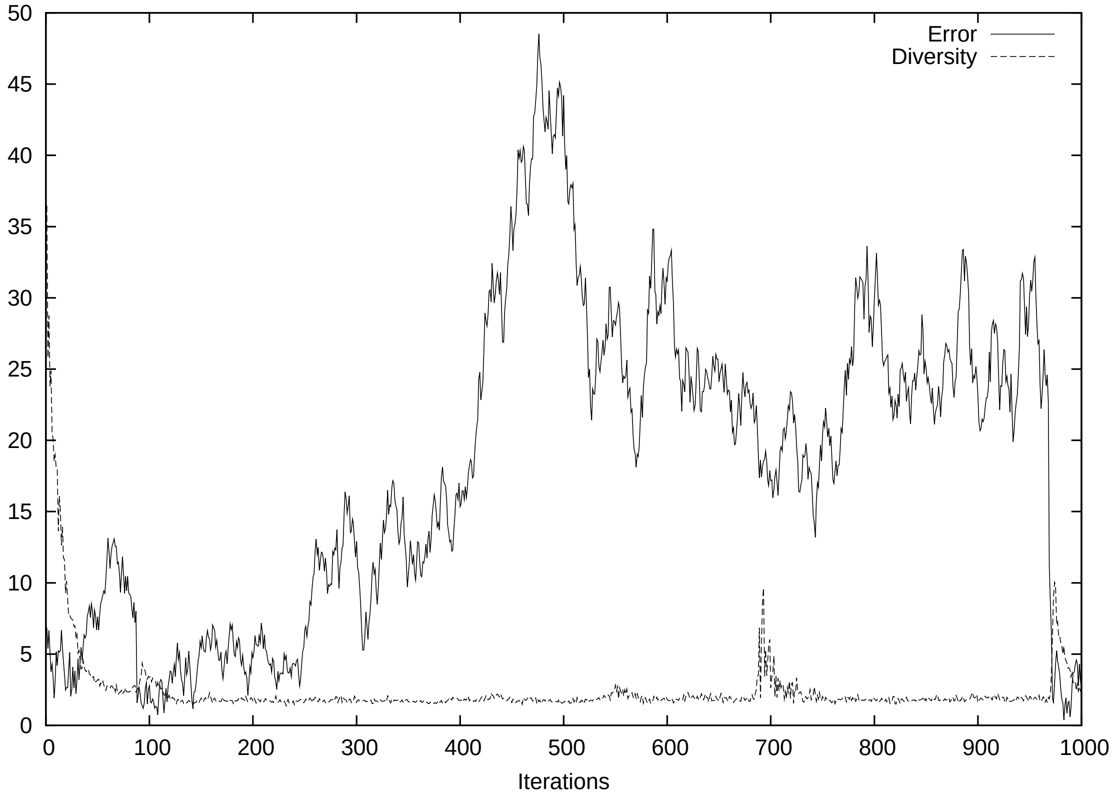
Table 7.1: Experimental results of standard PSO

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$3.12e{-}2 \pm 5.04e{-}3$	$0.48 \pm 2.75e{-}2$	$2.27 \pm 8.59e{-}2$	17.90 ± 0.58	42.64 ± 0.96
	5 p	$3.46e{-}2 \pm 4.31e{-}3$	1.33 ± 0.62	2.29 ± 0.11	19.24 ± 0.47	52.62 ± 1.48
	15 p	$5.61e{-}2 \pm 7.98e{-}3$	$0.52 \pm 4.62e{-}2$	3.98 ± 1.33	24.78 ± 1.08	52.57 ± 1.38
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$4.22e{-}7 \pm 2.93e{-}8$	$6.91e{-}4 \pm 4.80e{-}5$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$7.57e{-}2 \pm 5.26e{-}3$	$3.66e{-}3 \pm 2.54e{-}4$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-}3 \pm 2.58e{-}4$	$0.28 \pm 1.93e{-}2$	$6.45e{-}2 \pm 4.48e{-}3$
E-PROGRESS (CME)	1 p	32.31 ± 1.63	29.85 ± 0.93	24.42 ± 0.53	50.67 ± 1.17	125.54 ± 4.18
	5 p	28.71 ± 1.19	37.05 ± 2.15	41.87 ± 5.06	47.90 ± 1.00	143.73 ± 5.92
	15 p	29.78 ± 1.29	35.44 ± 2.24	38.92 ± 1.42	60.86 ± 2.28	121.19 ± 6.14
E-ABRUPT (CME)	1 p	$67.82 \pm 3.76e{-}3$	$113.39 \pm 3.59e{-}2$	$78.47 \pm 9.85e{-}2$	118.00 ± 0.67	146.00 ± 5.15
	5 p	59.43 ± 0.88	79.90 ± 0.27	91.50 ± 3.55	144.94 ± 2.15	166.48 ± 3.78
	15 p	$30.40 \pm 3.89e{-}3$	77.45 ± 0.69	94.74 ± 1.30	140.30 ± 3.77	155.83 ± 5.83
E-ABRUPT (ABEBC)	1 p	$67.68 \pm 1.31e{-}9$	$112.65 \pm 4.32e{-}6$	$75.84 \pm 2.21e{-}3$	104.11 ± 0.58	119.11 ± 6.77
	5 p	58.99 ± 1.31	79.31 ± 0.27	88.83 ± 3.59	129.96 ± 1.96	142.06 ± 4.53
	15 p	$30.32 \pm 6.05e{-}10$	76.86 ± 0.69	92.38 ± 1.29	126.46 ± 3.96	130.79 ± 7.28
E-ABRUPT (ITEL)	1 p	$160.31 \pm 5.08e{-}2$	163.21 ± 0.28	170.97 ± 0.37	199.57 ± 0.35	200.00 ± 0.00
	5 p	158.63 ± 2.58	163.47 ± 0.34	175.01 ± 3.35	200.00 ± 0.00	200.00 ± 0.00
	15 p	$160.24 \pm 3.62e{-}2$	163.65 ± 0.35	171.72 ± 0.66	200.00 ± 0.00	200.00 ± 0.00
E-CHAOS (CME)	1 p	87.85 ± 2.09	187.68 ± 4.23	272.85 ± 5.51	422.52 ± 8.60	500.86 ± 7.45
	5 p	49.91 ± 1.05	102.14 ± 2.37	170.40 ± 4.37	274.02 ± 5.56	368.76 ± 5.64
	15 p	34.48 ± 0.46	81.01 ± 1.35	124.71 ± 1.90	233.32 ± 3.95	298.03 ± 3.61
E-CHAOS (ABEBC)	1 p	87.15 ± 2.05	186.51 ± 4.24	272.28 ± 5.51	421.96 ± 8.59	499.98 ± 7.44
	5 p	49.52 ± 1.05	101.63 ± 2.36	169.37 ± 4.36	273.11 ± 5.59	367.94 ± 5.66
	15 p	34.26 ± 0.45	80.32 ± 1.39	123.88 ± 1.93	232.43 ± 3.99	297.19 ± 3.60
E-CHAOS (ITEL)	1 p	$4.86 \pm 1.06e{-}2$	$5.00 \pm 3.41e{-}4$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$4.88 \pm 1.04e{-}2$	$5.00 \pm 2.09e{-}3$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$4.89 \pm 1.95e{-}2$	$4.99 \pm 2.57e{-}3$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	63.43 ± 0.40	89.86 ± 0.66	171.46 ± 0.92	255.05 ± 1.90	356.34 ± 4.48
	5 p	49.31 ± 0.48	90.78 ± 1.23	134.41 ± 1.44	235.63 ± 3.28	329.57 ± 4.29
	15 p	39.65 ± 1.45	56.35 ± 1.45	90.86 ± 1.59	225.05 ± 4.02	294.16 ± 3.99

does not spend time exploring local optima. However, for multimodal environments the algorithm did not perform systematically better (or worse) for environments with 15 peaks than for those with 5 peaks.

As explained in chapter 4, the capacity of the standard PSO algorithm to optimise dynamic problems is limited. Therefore, few conclusions can be drawn from the (poor) performance of the algorithm on the dynamic test cases. Yet, a number of observations can be made.

The performance level of the algorithm for E-PROGRESS was much lower than for E-STATIC (higher CME), which illustrates the limitations of the standard PSO applied to DEs. The E-PROGRESS environments are modified frequently and as mentioned in chapter 4, the standard PSO is limited by outdated memory and diversity loss. As for E-STATIC, the algorithm generally performed better on problems with lower dimensionality but, for E-PROGRESS, unimodal problems have not systematically shown to be easier to solve. Graph 7.1 shows the progression of the error and diversity for a unimodal E-PROGRESS with two dimensions. The graph shows that the diversity dropped during the first iterations because the swarm converged while the peak remained within the area covered by the swarm. However, once the peak had moved away, the particles remained attracted to formerly good positions due to outdated memory and the swarm did not converge any further. Figure 7.1 shows that after 500 iterations, the peak had moved away from the swarm while the particles remained located in the area where the peak formerly was. As explained in section 4.2, the swarm can only converge or move towards the peak if more attractive solutions than the *pbest* of the particles are discovered. The discovery of a better *pbest* can be seen on graph 7.1 around iterations 100, 700 and 970 where the diversity increased and the error decreased suddenly. For the multimodal E-PROGRESS environments, the performance can be influenced negatively if the swarm converges on a local optimum, or if a new global optimum emerges outside the area covered by the swarm. However, having more peaks travelling in the search space increases the chances for a particle to encounter a solution with higher fitness than its outdated *pbest*.



Graph 7.1: Standard PSO for E-PROGRESS with two dimensions and one peak

For E-ABRUPT, the CME was higher than for E-PROGRESS. For the environments with two or five dimensions, the CME was higher for the unimodal problems than for the multimodal problems. In these environments, the swarm is likely to converge before the first change. The results obtained for E-STATIC have indeed showed that the swarm can quickly find the optimum on problems with low dimensionality and converge towards the optimum. If an abruptly changing environment stays static for long enough before the first change occurs, the swarm can converge to a point where all the particles are located at the same position. In such scenarios, the particles remain on their positions until the end of the simulation no matter how much the environment changes, and the optimisation process stops. For unimodal E-ABRUPT environments with a higher number of dimensions, the swarm does not have time to fully converge before the first change. For multimodal E-ABRUPT environments, a loosely connected swarm can be

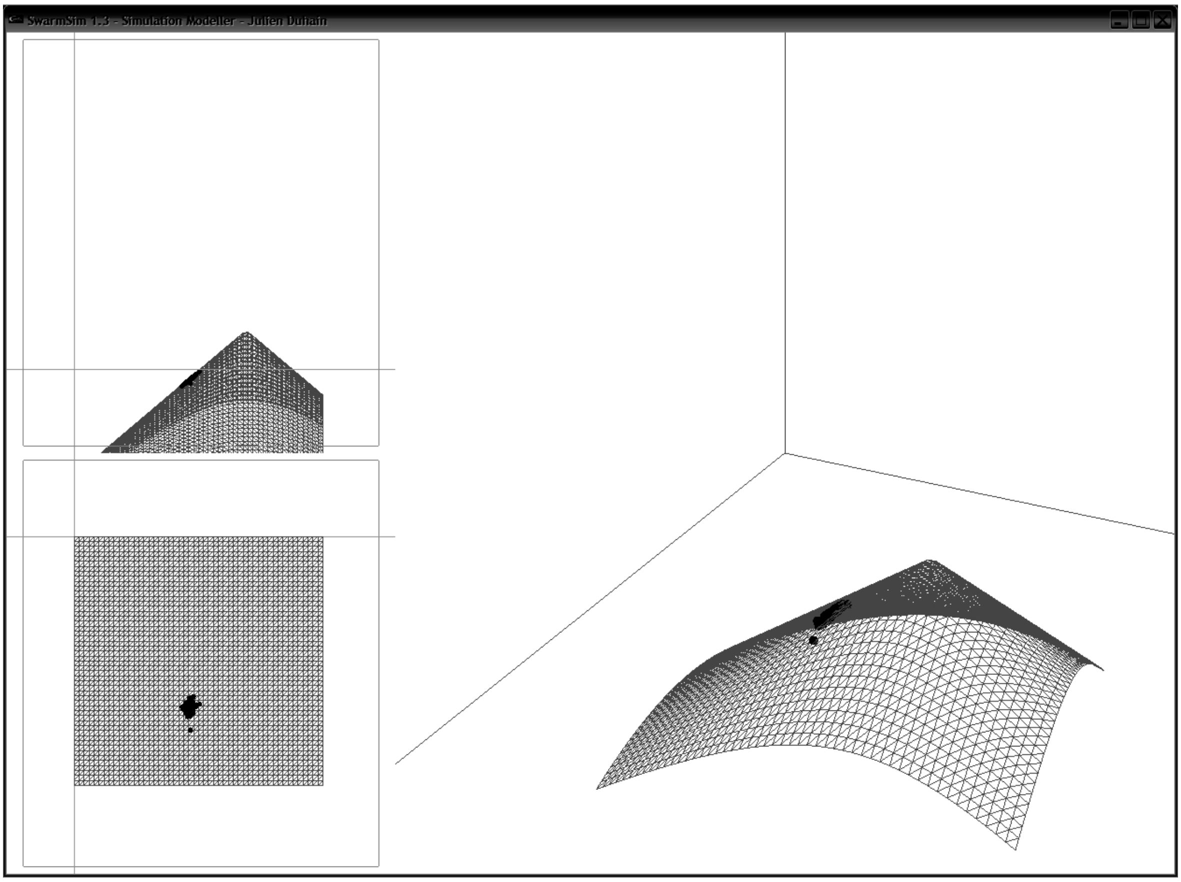
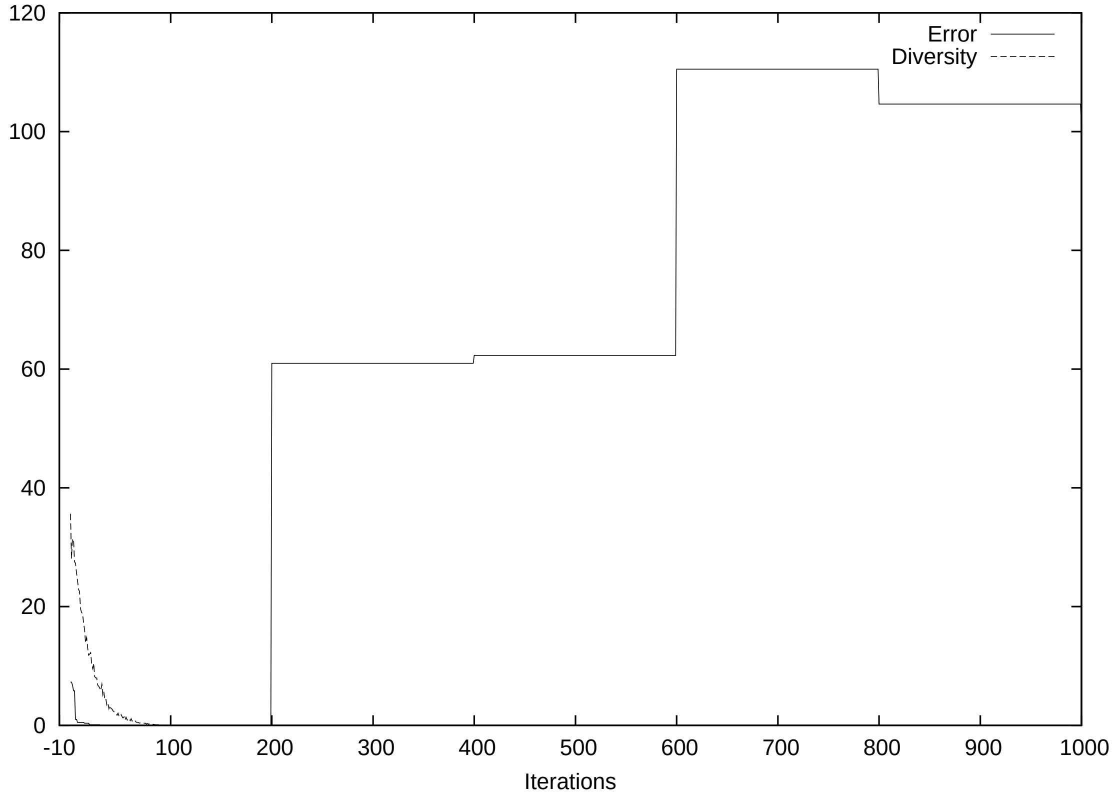


Figure 7.1: Standard PSO for E-PROGRESS (iteration 500)

prevented from converging if the swarm explores multiple peaks simultaneously as was shown in chapter 4 and illustrated in figure 4.4. After the first change, outdated memory can also maintain the swarm in place after changes have occurred as shown in figure 4.1.

Graph 7.2 illustrates the progression of the error and diversity of the swarm in the two-dimensional E-ABRUPT with one peak for one of the simulations. The PSO algorithm performed well before the first change, but the diversity (higher line on the graph) dropped and optimisation stopped before iteration 200. For the rest of the simulation, the diversity was close to zero: the best particle kept its position but the fitness at the position of the best particle varied with each environmental change as can be seen on the graph. The ABEBC and ITEL measurements also indicate a poor performance



Graph 7.2: Standard PSO for E-ABRUPT with two dimensions and one peak

level. The ABEBC measurements are only slightly lower than the corresponding CME measurements. Since the ABEBC only takes error measurements before changes, having a similar CME and ABEBC shows that the error is roughly the same right after a change and 200 iterations later (right before the next change). This means that there is little or no improvement in the fitness of the best particle while the environment stays static.

For E-CHAOS, the CME was significantly higher than for E-ABRUPT for all environments except the two-dimensional, five peaks environment. The changes for E-CHAOS are as severe as for E-ABRUPT but much more frequent. However, the environment does not remain static for long enough for the swarm to converge to a point where all particles are located at the same position. This could explain why the performance level was higher for E-CHAOS with two dimensions and five peaks than for the E-ABRUPT counterpart. ITEL reached 5.00 for all environments with a dimensionality of 10 or higher indicating that an error level of 10 or below was never reached. The ABEBC measurements were not significantly lower than the CME measurements, showing that there was little or no improvement in the fitness of the best particle while the environment stayed static. For E-CHAOS, the performance level increased with the number of peaks. If the swarm cannot optimise the function because of outdated memory, the particles have a higher probability of being (by chance) located on a peak if there are more peaks in the landscape.

For E-PATTERN, the CME was higher than for E-PROGRESS, but lower than for E-CHAOS. As mentioned in section 6.2, while the temporal severity is identical for E-PATTERN and E-PROGRESS, E-PATTERN is modified differently than the other DEs where the spatial severity of E-PATTERN increases with increase in the dimensionality. Figure 7.2 illustrates the swarm taking a crescent shape as the particles gather along a part of the trajectory of the peak. When the peak was located on the part of the trajectory covered by the swarm, the error was relatively small but the rest of the time the algorithm performed poorly as shown in graph 7.3. Outdated memory prevented the swarm from tracking the optimum but the particles gathered closer to the trajectory of the peak each time that the peak went through the swarm. The behaviour of the standard PSO was therefore influenced by the pattern in the change.

7.3 Re-evaluating PSO

As mentioned in section 4.4, particles should re-evaluate their *pbest* to overcome the outdated memory limitation of the PSO algorithm. The re-evaluating PSO is therefore expected to outperform the standard PSO for the dynamic test cases. Since all the algorithms tested in chapter 8 are re-evaluating the *pbest* of the particles, the performance of these algorithms should be compared to that of the re-evaluating PSO, so that if a PSO algorithm designed for DEs improves on the re-evaluating PSO, this improvement can be attributed to the approach taken by that particular algorithm and not on the

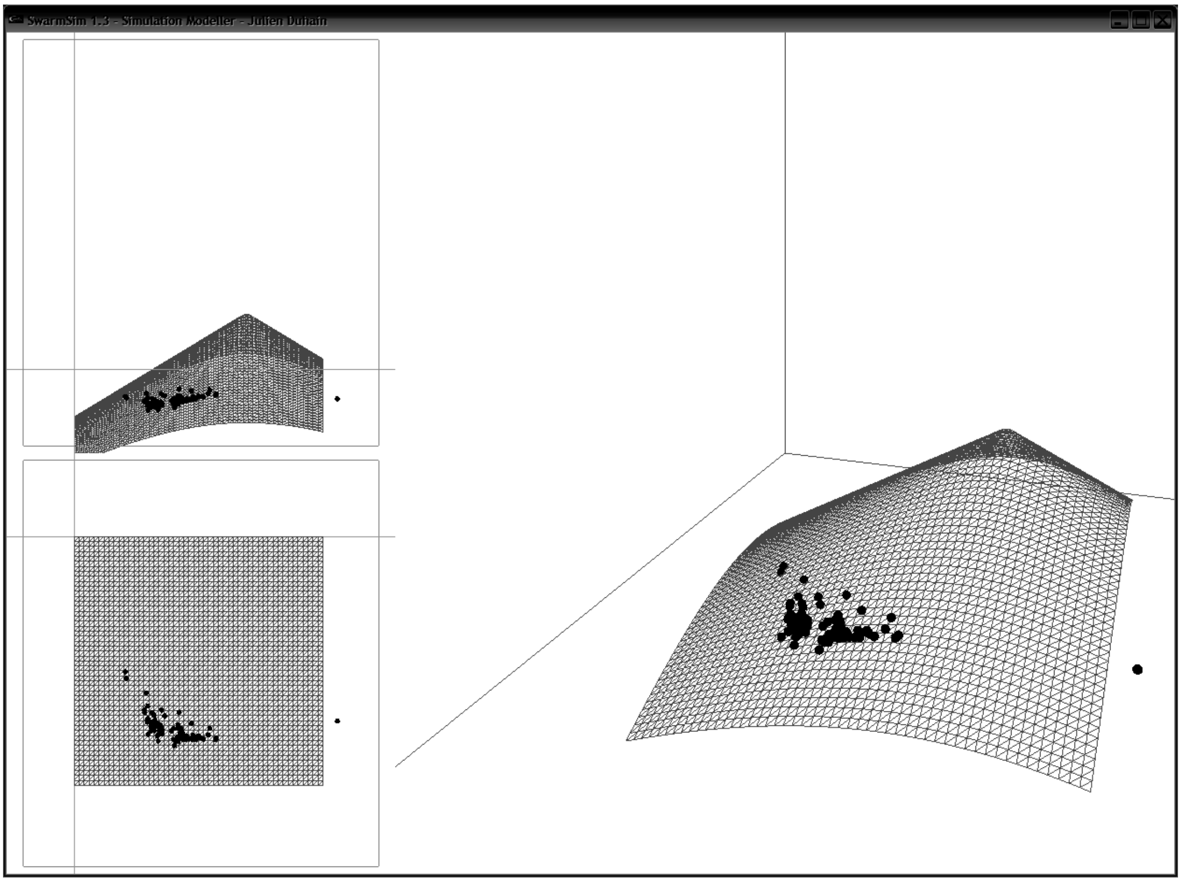


Figure 7.2: Standard PSO for E-PATTERN (iteration 941)

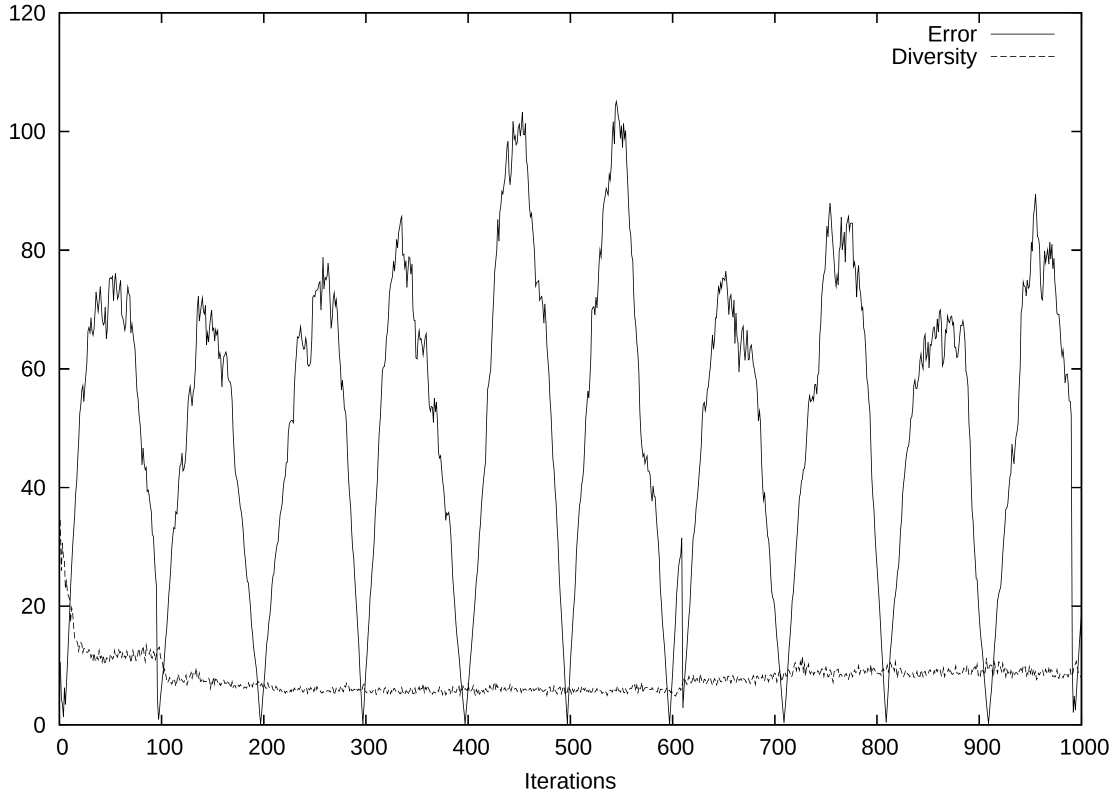
effect of *pbest* re-evaluation. Section 7.3.1 presents the results obtained from applying the re-evaluating PSO to the test environments. Section 7.3.2 analyses these results and examines the behaviour of the re-evaluating swarm in the various environments.

7.3.1 Results

The results in table 7.2 were obtained by an implementation of the PSO algorithm described in chapter 3 using the configuration from section 6.4 and re-evaluating the *pbest* values according to section 4.4 every iteration. The code for the re-evaluating PSO can be found in CIlib [74].

Table 7.2: Experimental results of re-evaluating PSO

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$2.37e{-2}\pm3.51e{-3}$	$0.42\pm2.49e{-2}$	$2.30\pm9.72e{-2}$	17.40 ± 0.41	42.59 ± 0.86
	5 p	$2.75e{-2}\pm3.20e{-3}$	1.59 ± 0.74	2.96 ± 1.24	18.55 ± 0.46	52.04 ± 1.80
	15 p	$5.64e{-2}\pm1.14e{-2}$	$0.46\pm3.19e{-2}$	3.29 ± 1.02	23.72 ± 1.14	51.44 ± 1.76
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$1.63e{-6}\pm1.13e{-7}$	$1.08e{-3}\pm7.51e{-5}$
	5 p	0.00 ± 0.00	$0.16\pm1.14e{-2}$	0.00 ± 0.00	$7.57e{-2}\pm5.26e{-3}$	$1.74e{-3}\pm1.21e{-4}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-3}\pm2.58e{-4}$	$0.28\pm1.97e{-2}$	$0.14\pm9.97e{-3}$
E-PROGRESS (CME)	1 p	$1.68\pm3.13e{-3}$	$2.31\pm2.68e{-2}$	$5.27\pm9.79e{-2}$	25.05 ± 0.39	52.98 ± 0.84
	5 p	4.37 ± 0.35	6.99 ± 0.47	10.89 ± 1.36	27.80 ± 0.62	57.60 ± 1.13
	15 p	6.45 ± 0.56	11.14 ± 0.77	12.23 ± 1.83	34.64 ± 1.23	53.69 ± 1.03
E-ABRUPT (CME)	1 p	6.79 ± 0.23	9.00 ± 0.18	11.08 ± 0.15	32.52 ± 0.41	63.22 ± 0.73
	5 p	7.17 ± 1.48	8.24 ± 0.53	14.42 ± 0.59	37.56 ± 0.72	80.72 ± 1.17
	15 p	6.69 ± 1.91	5.48 ± 1.01	14.94 ± 0.37	45.48 ± 2.16	72.06 ± 1.09
E-ABRUPT (ABEBC)	1 p	$2.53e{-8}\pm8.87e{-9}$	$5.62e{-5}\pm4.99e{-6}$	$1.50e{-2}\pm9.52e{-4}$	4.49 ± 0.17	21.47 ± 0.49
	5 p	3.05 ± 1.03	0.71 ± 0.30	3.64 ± 0.58	7.75 ± 0.37	36.70 ± 0.87
	15 p	4.49 ± 1.68	1.54 ± 0.74	5.49 ± 0.22	14.82 ± 2.14	30.58 ± 1.05
E-ABRUPT (ITEL)	1 p	22.81 ± 0.73	30.77 ± 0.60	44.43 ± 0.67	132.69 ± 0.99	198.96 ± 0.45
	5 p	53.22 ± 10.75	27.54 ± 1.26	81.71 ± 4.47	155.43 ± 1.53	200.00 ± 0.00
	15 p	43.82 ± 13.91	26.36 ± 6.02	105.96 ± 2.40	183.20 ± 4.95	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.66\pm5.13e{-2}$	42.26 ± 0.12	74.05 ± 0.29	173.72 ± 0.90	257.67 ± 1.40
	5 p	$14.29\pm6.90e{-2}$	38.79 ± 0.13	67.80 ± 0.39	157.03 ± 2.13	226.89 ± 1.79
	15 p	$10.69\pm5.77e{-2}$	33.07 ± 0.13	59.34 ± 0.27	132.55 ± 1.25	195.47 ± 3.18
E-CHAOS (ABEBC)	1 p	$2.64\pm3.59e{-2}$	21.99 ± 0.15	58.73 ± 0.32	165.70 ± 0.88	251.75 ± 1.40
	5 p	$3.48\pm3.85e{-2}$	23.75 ± 0.15	56.62 ± 0.39	150.58 ± 2.05	222.11 ± 1.79
	15 p	$2.86\pm4.09e{-2}$	20.98 ± 0.14	49.75 ± 0.23	127.52 ± 1.21	191.60 ± 3.11
E-CHAOS (ITEL)	1 p	$1.42\pm1.45e{-2}$	$4.95\pm5.97e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.42\pm1.39e{-2}$	$4.95\pm1.12e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.11\pm1.04e{-2}$	$4.93\pm9.92e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.17\pm2.88e{-3}$	20.92 ± 0.12	86.66 ± 0.39	217.04 ± 0.65	292.54 ± 0.70
	5 p	$4.15\pm4.18e{-3}$	23.88 ± 0.17	73.47 ± 0.33	187.22 ± 0.58	271.79 ± 1.13
	15 p	5.38 ± 0.10	20.38 ± 0.12	59.96 ± 0.17	178.18 ± 0.65	239.10 ± 0.75



Graph 7.3: Standard PSO for E-PATTERN with two dimensions and one peak

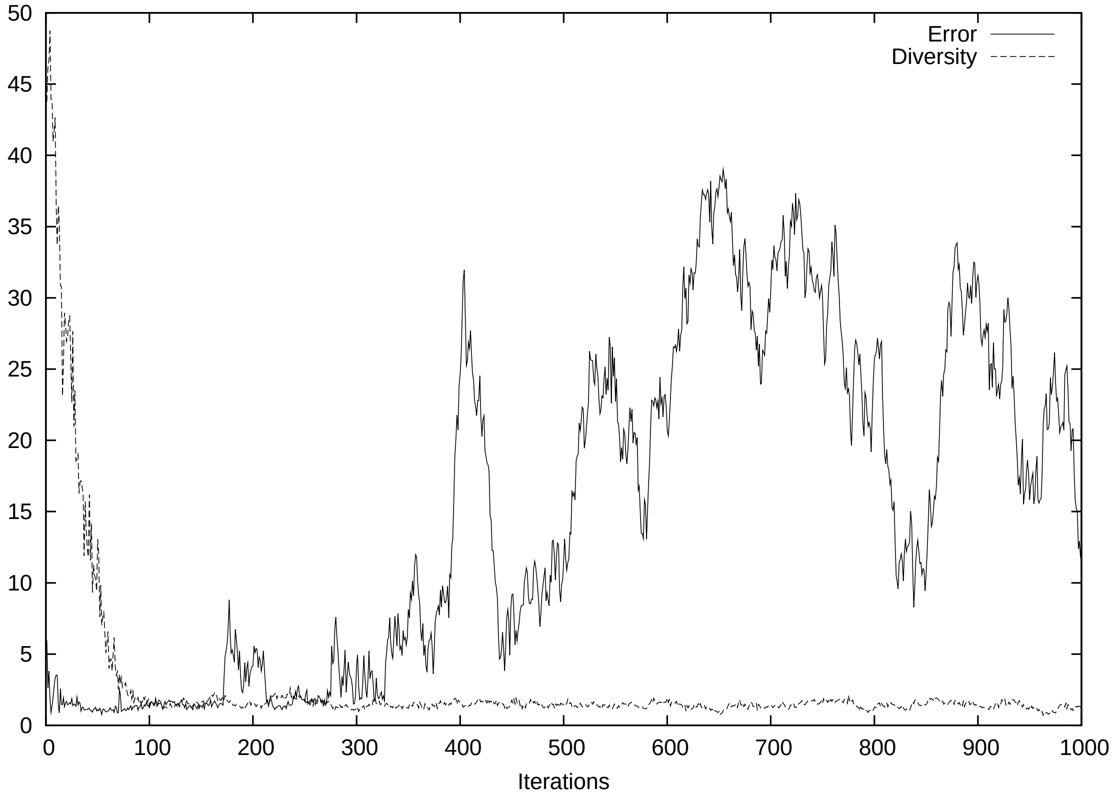
7.3.2 Analysis of Results

The p-values determining the significance of the difference between the results of the standard PSO and those of the re-evaluating PSO are listed in table C.1.

For E-STATIC, the re-evaluating PSO behaved in the exact same way as the standard PSO since the re-evaluations of the *pbests* were always equal to the original evaluations. The experiments confirm that the performance level of the re-evaluating PSO was not significantly different from that of the standard PSO for E-STATIC.

For E-PROGRESS, the CME was higher than for E-STATIC. As the changes occur every iteration in E-PROGRESS, the algorithm did not have the opportunity to have all its particles converge on one location before a change occurs. The algorithm performed considerably better for unimodal E-PROGRESS environments than for their multimodal counterparts. This indicates that the algorithm struggled to detect the appearance of new optima. Graph 7.4 shows an example of the progression of the error and diversity over 1000 iterations for an E-PROGRESS simulation with two dimensions and five peaks. The diversity dropped quickly during the first iterations as the swarm converged around one peak. The diversity was then maintained at a relatively constant level. With each change, the peak was relocated to a close-by location within the area covered by the swarm which allowed the swarm to track the peak. A different set of particles therefore became attractive after every change which prevented the swarm from converging below a certain threshold. The swarm kept track of the same peak unless a higher peak moved within the area covered by the swarm. The behaviour of the re-evaluating PSO for the two-dimensional E-PROGRESS with five peaks is illustrated in figure 7.3. Graph 7.4 shows that the tracked peak was originally the global optimum, but around iteration 170 and again around iteration 280, the algorithm failed to detect that the peak being tracked was no longer the global optimum. The diversity level remained more or less constant but low through the simulation which illustrates that, for E-PROGRESS, *pbest* re-evaluation does not prevent diversity loss. For multimodal E-PROGRESS environments, the CME increased with the number of peaks for all but the 50 dimensional problems. A higher number of peaks increases the swarm's chances to track a local optimum (instead of the global optimum). However, a higher number of peaks also increases the chances of the swarm to encounter a higher peak.

For E-ABRUPT, the CME was higher than for E-PROGRESS (for all but the five-dimensional environment with 15 peaks) showing that on average, the error was lower for the E-PROGRESS environments. However, the ABEBC for E-ABRUPT was significantly lower than the CME for E-PROGRESS showing that the algorithm exploited better for E-ABRUPT than for E-PROGRESS. The ITEL measurements show that a higher dimensionality generally increased the number of iterations needed for the algorithm to reach an error of 10 or less. However, the ITEL was lower for the multimodal environments with five dimensions than for those with two dimensions.



Graph 7.4: Re-evaluating PSO for E-PROGRESS with two dimensions and five peaks

Graph 7.5 illustrates the progression of the error and diversity over 1000 iterations for the two-dimensional E-ABRUPT with five peaks. Initially, both error and diversity dropped as the swarm converged on the optimum. After the first change, the diversity increased quickly then dropped close to zero. After each of the last three changes, the diversity increased slowly while the error decreased slowly. Then, about 20 iterations after the change, the error dropped as the diversity increased more and more. About 40 iterations after the change, a decrease in diversity can be seen as the swarm started to exploit a new peak. After 400 iterations, the swarm did not regain as much diversity and as a consequence, became trapped in a local optimum until iteration 600. A similar scenario happened between iteration 600 and 800. Every time a change occurred, diversity was re-introduced in the swarm. As long as the environment was static, the attractors (*pbest*'s and *nbest*'s) became closer and closer to an optimum and the velocity of the

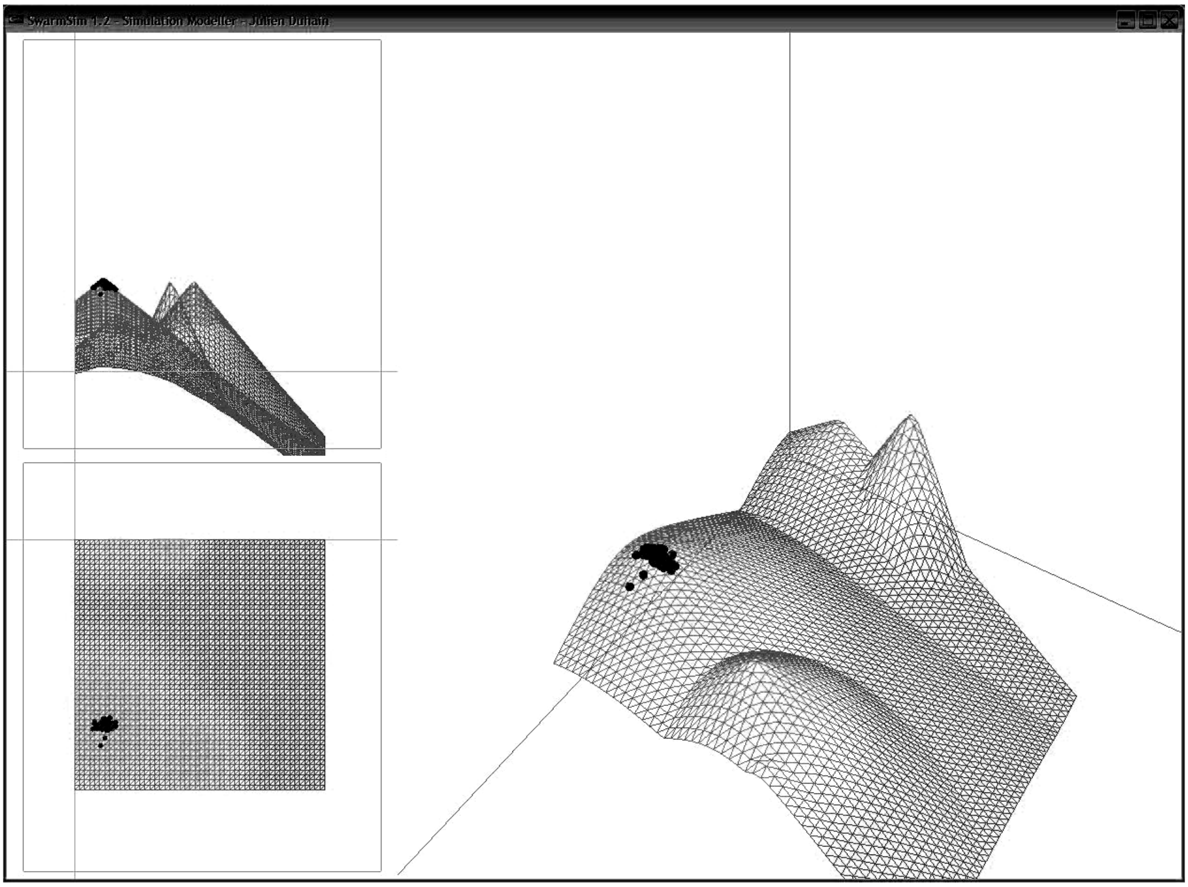
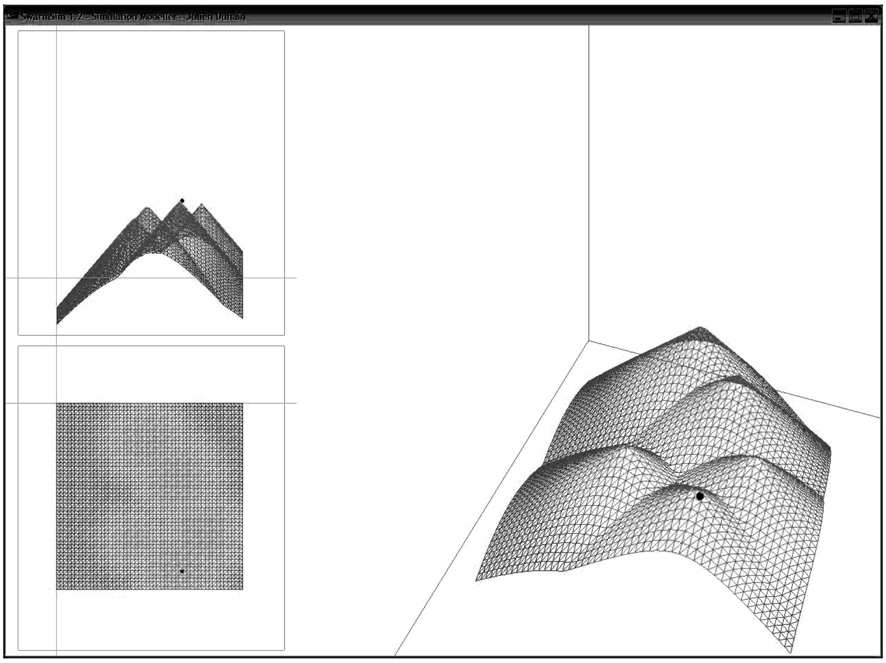
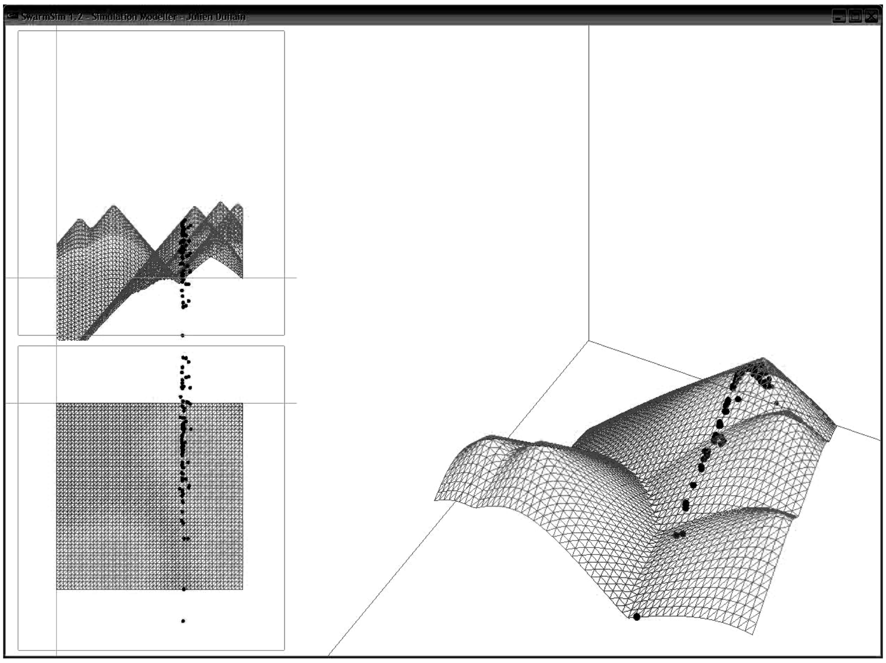


Figure 7.3: Re-evaluating PSO for E-PROGRESS (iteration 500)

particles decreased progressively as the swarm converged. When the optimum moved to a new location, the particles closest to the new optimum became the attractors. Consequently, the velocity of most particles were likely to grow since the particles were now located further from their neighborhood best. Particles could jump beyond their *nbest*, possibly discovering a better position which caused the other particles to accelerate even more.

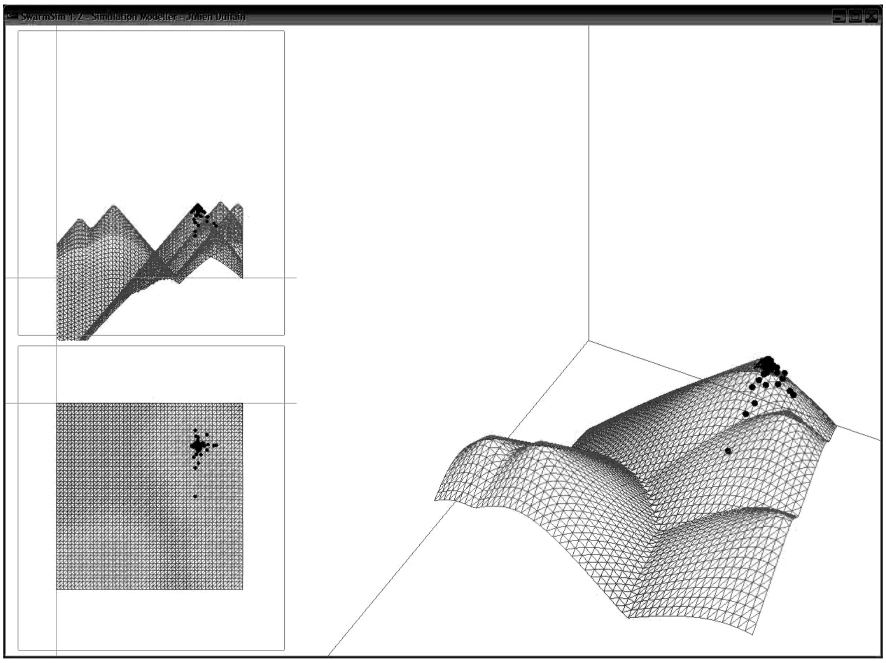


(a) Iteration 399



(b) Iteration 428

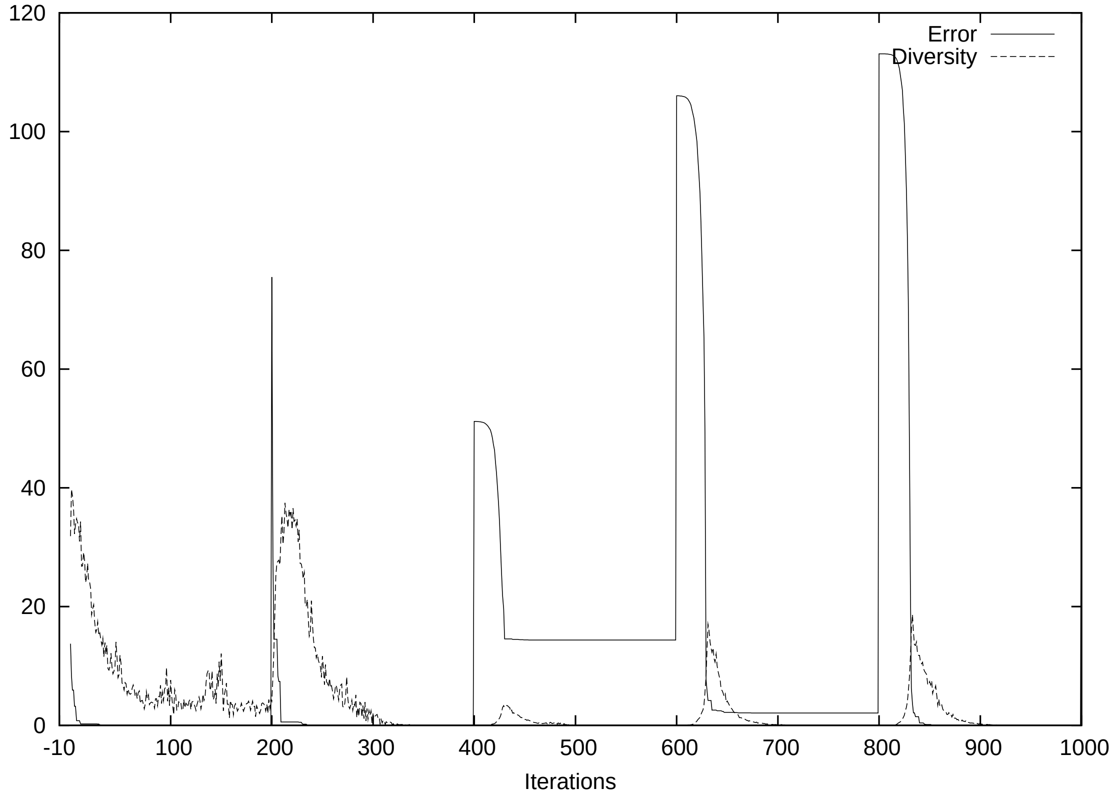
Figure 7.4: Behaviour of the re-evaluating PSO for E-ABRUPT (change at iteration 400)



(c) Iteration 463

Figure 7.4: (continued)

For E-ABRUPT, the swarm converged to an extent while the environment was static but after a change the particles accelerated while "climbing" towards the closest peak as illustrated in figure 7.4. Once an optimum was found, the swarm converged again. The level of diversity regained by the swarm after a change depends on the velocity the particles can reach before the swarm starts converging again. If an optimum was found quickly near the location of the swarm, the particles gained little velocity before converging on the new peak and only a fraction of the search space was explored. It can also be observed in figure 7.4(b) that the particles almost formed a line parallel to one dimension. The velocity regained is therefore not equal in all dimensions. The attractors' positions influence the step size, and therefore the area of the search space that is explored after a change. As long as some particles have a non-zero velocity, the diversity of the swarm can increase after a change. However, if at any point in time the swarm converges and the velocity of all particles tends to zero in all directions, the particles



Graph 7.5: Re-evaluating PSO for E-ABRUPT with two dimensions and five peaks

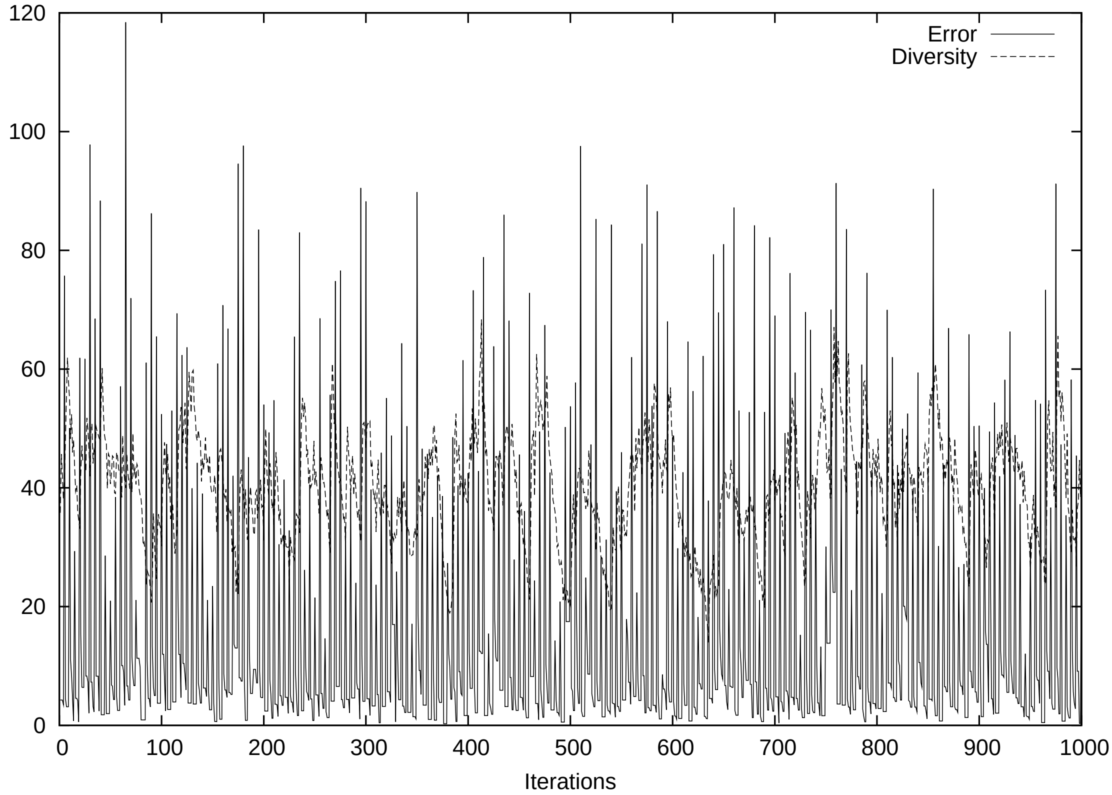
remain in place from that point onward, no matter how much the environment changes. As observed previously, the swarm took longer to converge for environments of higher dimensionality. This means that for higher dimensional E-ABRUPT environments, particles kept a higher velocity while the environment was static and consequently regained diversity faster after a change. This explains why the ITEL was lower for the multimodal environments with five dimensions than for those with two dimensions.

For E-CHAOS, the performance worsened with increase in the dimensionality. The CME was lower than for E-ABRUPT. Also, the CME increased proportionally to the number of peaks present in the environment. This seems to indicate that the algorithm struggled to exploit and that a higher number of peaks increased the chance of a particle to randomly land on a good position. However, the ABEBC measurements were significantly lower than the CME measurements indicating that a considerable improvement in error was made while the environment stayed static. For the E-CHAOS with two dimensions and 15 peaks, the ABEBC was lower than for the corresponding E-ABRUPT environment, possibly because the swarm kept a higher diversity for E-CHAOS and was therefore less susceptible to become trapped in a local optimum. Graph 7.6 illustrates the progression of the error and diversity of the re-evaluating PSO for an E-CHAOS environment with two dimensions and five peaks. The diversity was fluctuating widely but remained at a high while the error increased with every change before reducing rapidly. A high level of diversity was maintained by the frequent and severe changes. A large area was therefore covered by the swarm without using any mechanism to re-introduce diversity. Figure 6.1(b) from chapter 6 illustrates the re-evaluating PSO for the two-dimensional E-CHAOS with five peaks just before a change. This figure shows that the swarm exploited the optimum while exploring a large section of the search space.

For E-PATTERN, the CME increased dramatically with increase in dimensionality. For the two-dimensional E-PATTERN, the CME increased with the number of peaks, as was the case for E-PROGRESS. Figure 7.5 illustrates the re-evaluating PSO for the two-dimensional E-PATTERN with five peaks. The behaviour of the swarm was similar to the behaviour observed for E-PROGRESS and illustrated in figure 7.3. However, for the E-PATTERN environments with 10 or more dimensions, where the spatial severity is higher, the CME decreased as the number of peaks increased, as was the case for E-CHAOS. Graph 7.7 shows that the error of the re-evaluating PSO increased and decreased periodically with the environment cycle. This shows that the swarm could not track the optimum effectively for high dimensionality E-PATTERN as the peaks are moving too quickly through the landscape.

7.4 Comparison of Standard and Re-evaluating PSO

For E-STATIC, the standard and the re-evaluating PSO behaved in the exact same way and their performances were similar. However, the re-evaluating PSO wastes computa-



Graph 7.6: Re-evaluating PSO for E-CHAOS with two dimensions and five peaks

tion time re-evaluating the *pbest* of the particles and is therefore more computationally expensive than the standard PSO.

As expected, for all the dynamic test cases the results from table 7.2 show a significant improvement in performance compared to the results from table 7.1. The re-evaluating PSO is not limited by outdated memory, and with an up-to-date memory the particles of the re-evaluating swarm are only attracted towards *currently* good solutions.

For E-PATTERN, the standard PSO was also outperformed by the re-evaluating PSO. However, by comparing the performance of the standard PSO for the high-dimensional E-PATTERN and E-CHAOS environments, it can be observed that this algorithm takes advantage of the pattern in the change. The CME of the re-evaluating PSO was considerably higher for the E-PATTERN environments with 10 or more dimensions than its

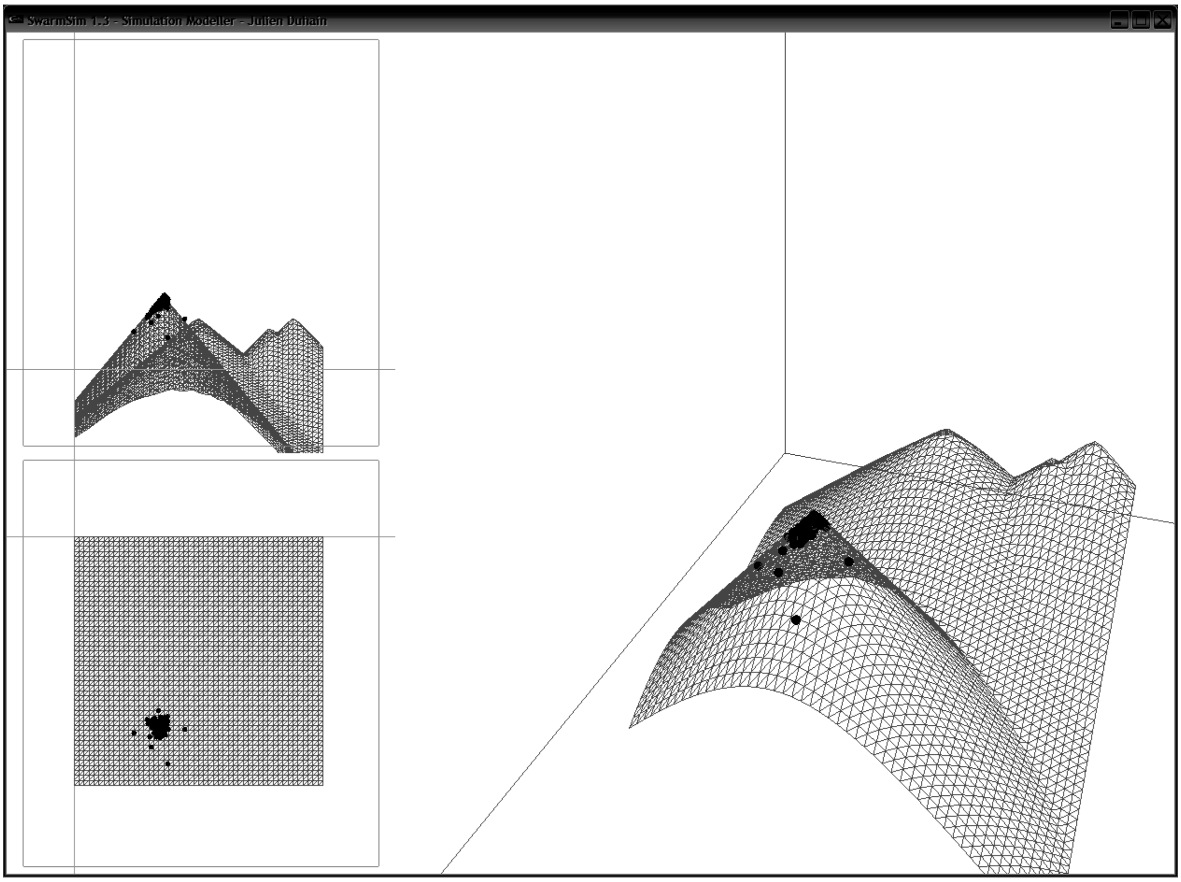
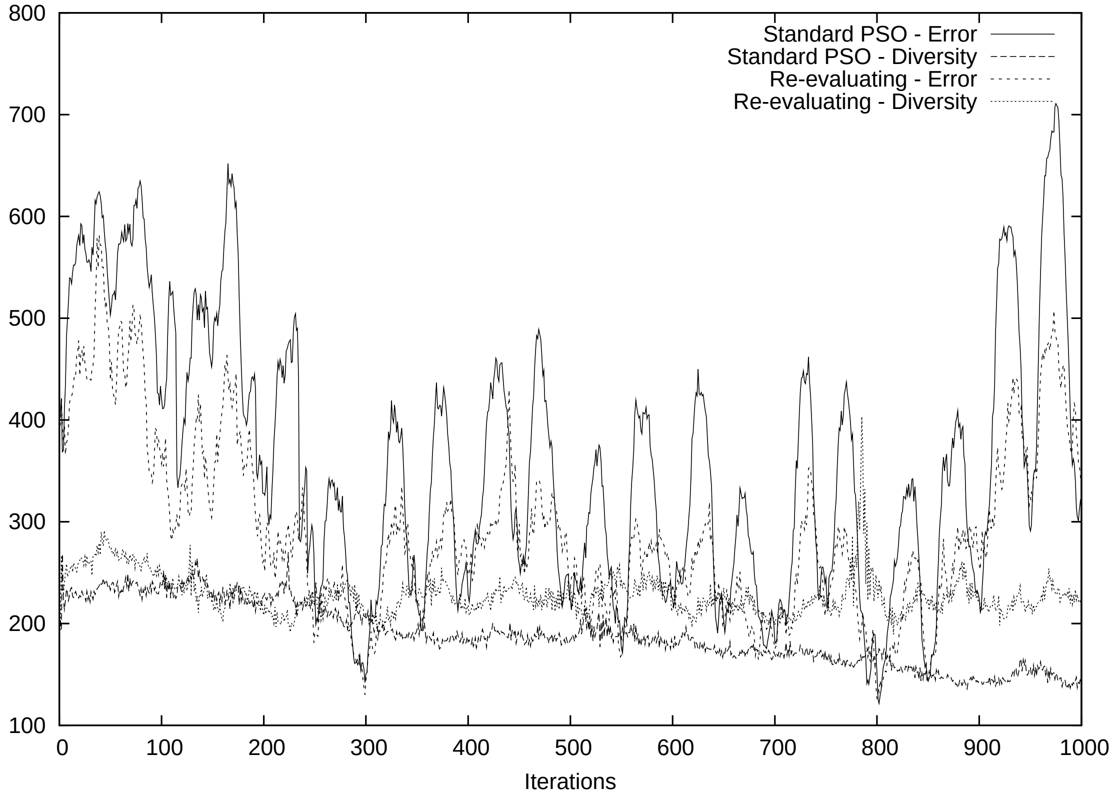


Figure 7.5: Re-evaluating PSO for E-PATTERN (iteration 100)

CME for the corresponding E-CHAOS environments. In contrast, the standard PSO performed significantly better for these E-PATTERN environments than for the corresponding E-CHAOS. This indicates that the presence of a pattern in the change influences the performance of the standard PSO positively. Graph 7.7 compares the progression of the error and diversity of the standard PSO and re-evaluating PSO applied to the 50-dimensional E-PATTERN with one peak. It can be seen that the diversity decreased with time for the standard PSO unlike the diversity of the re-evaluating PSO which kept fluctuating but did not clearly decrease over time. The diversity of the swarm of the standard PSO decreased as the particles gathered along the path of the optimum as illustrated in figure 7.2. Even though the re-evaluating PSO kept a lower error than the standard PSO, the capacity of the standard PSO to converge along the path of the



Graph 7.7: Standard and re-evaluating PSO for E-PATTERN with 50 dimensions and one peak

optimum could be further researched and exploited.

7.5 Summary

The standard and re-evaluating PSO have been applied to the environment test cases. The results of these experiments have been examined in this chapter and a number of observations have been made. The standard PSO performed well for E-STATIC but poorly for the DEs. However, the standard PSO showed a capacity to take advantage of the presence of a pattern in the change. The re-evaluating PSO behaved similarly to the standard PSO for E-STATIC. For E-PROGRESS, the re-evaluating PSO showed a capacity to track a peak but struggled to detect new optima. For E-ABRUPT, the re-evaluating PSO showed an increase in diversity after a change which allows the swarm to detect new optima. For E-CHAOS, the re-evaluating PSO maintained a high level of diversity and was able to optimise the function between the changes. The performance measurements of the re-evaluating PSO for E-PATTERN indicate similarities between the behaviour of the algorithm for low dimensionality E-PATTERN and E-PROGRESS environments. Similarities were also observed for high dimensionality E-PATTERN and E-CHAOS environments. The performance level of the algorithms evaluated in this chapter can now be used as benchmarks when evaluating algorithms designed for DEs.

Chapter 8

Evaluation of Swarm Algorithms Designed for Dynamic Environments

"It is not the strongest species that survive, nor the most intelligent, but the ones most responsive to change."

– Charles Darwin

With the experimental procedure and benchmarks being established, the main experiments can be carried out. This chapter presents and analyses the results of applying the selected swarm algorithms designed to optimise dynamic problems to various DEs. The analysis of these results aims at providing information about the performance and behaviours of the various swarm algorithms for the different types of DEs. This chapter also suggests the most effective approaches to overcome diversity loss for each type of DEs based on a comparison of the results.

8.1 Introduction

The dynamic algorithms from section 4.8 have been applied to the environment test cases described in section 6.2 according to the experimental procedure set up in chapter 6. Tables with the experimental results are provided in this chapter. These tables are formatted as in chapter 7. The first column contains the name of the environment with the measure used in brackets, and the number of peaks in the environment is indicated in the second column. *Scientific e notation* is used for values smaller than 0.1 and decimal notation (with two digits after the decimal point) is used for values greater than 0.1. Values between -2^{-1022} and 2^{-1074} are rounded to 0.00. The confidence interval (95%) is stated along with each measurement and calculated according to equation (7.1). The performance level of each algorithm is analysed for the various environments and compared to the performance level of the benchmarks. The Mann-Whitney U test [60] is used to compare the results of two sets of simulations. The difference between two sets of results is considered statistically significant only if the p-value obtained is below 0.05. Tables containing the p-values are provided in appendix C. As part of the analysis, a number of observations are made regarding the behaviour of the algorithms, and strengths and weaknesses are identified for each algorithm. The algorithms are then compared to each other based on the experimental results and suggestions are made regarding which approach is most suitable for which types of environments.

Reinitialising PSO, QSO, APSO, multi-swarm and SAMS are evaluated in sections 8.2 to 8.6. Section 8.7 contains a general comparison of the various algorithms based on the experiments and suggests the most efficient methods to adopt for the different types of environments.

8.2 Reinitialising PSO

The first dynamic algorithm to be assessed is the reinitialising PSO described in section 4.8.1. This algorithm uses a detection and response mechanism to overcome diversity loss and sets a percentage of the particles in the swarm to random positions in the search space when a change is detected.

Reinitialising the entire swarm has proven to be inefficient [27], and Eberhart and Shi have observed that reinitialising a lower percentage of the population is preferred for environments that only undergo smaller changes. Five versions of the algorithm are evaluated, reinitialising respectively 10%, 30%, 50%, 70% and 90% of the particles.

As mentioned in section 4.5.1, several detection mechanisms can be used by the reinitialising PSO. Evaluating the detection mechanisms is outside the scope of this work but it is important to note that the accuracy of the detection mechanism used can affect the performance level of the reinitialising PSO and that the computational cost associated with detection should also be taken into consideration when applying the algorithm to real-world problems. In the experiments, the algorithm is notified of the changes to simulate a failsafe detection mechanism and to produce results that are not affected by undetected changes.

The reinitialising PSO should perform best for environments where changes are not too frequent, so that the swarm has time to exploit before the reinitialisation occurs.

8.2.1 Results

The results in tables 8.1 to 8.5 were obtained by an implementation of algorithm 4.1 using the parameters from section 6.4 and reinitialising the percentage of the swarm mentioned in the legend of the table when a change occurs. The code for the reinitialising PSO can be found in CIlib [74].

Table 8.1: Experimental results of reinitialising PSO - 10% reinitialisation

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$3.24e{-2}\pm3.82e{-3}$	$0.48\pm2.49e{-2}$	2.35 ± 0.10	17.76 ± 0.47	43.70 ± 0.91
	5 p	$3.47e{-2}\pm4.72e{-3}$	1.94 ± 0.90	2.52 ± 0.10	18.89 ± 0.48	50.78 ± 1.55
	15 p	$6.19e{-2}\pm9.16e{-3}$	$0.56\pm5.51e{-2}$	3.11 ± 0.57	24.58 ± 1.50	53.60 ± 1.56
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$2.39e{-6}\pm1.66e{-7}$	$1.50e{-3}\pm1.04e{-4}$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$7.57e{-2}\pm5.26e{-3}$	$1.05e{-3}\pm7.32e{-5}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$1.77e{-5}\pm1.23e{-6}$	$0.39\pm2.69e{-2}$
E-PROGRESS (CME)	1 p	$1.72\pm4.37e{-3}$	$4.40\pm4.89e{-2}$	12.95 ± 0.30	58.69 ± 2.15	119.45 ± 4.26
	5 p	$2.43\pm6.37e{-2}$	$6.68\pm7.42e{-2}$	16.40 ± 0.37	63.43 ± 2.56	150.79 ± 7.38
	15 p	$2.42\pm2.14e{-2}$	8.76 ± 0.39	23.59 ± 0.82	70.82 ± 2.31	126.36 ± 6.84
E-ABRUPT (CME)	1 p	$0.48\pm2.04e{-2}$	$2.77\pm9.25e{-2}$	8.73 ± 0.17	32.82 ± 0.53	63.28 ± 0.82
	5 p	$0.56\pm1.81e{-2}$	2.97 ± 0.26	13.57 ± 0.44	37.23 ± 0.55	81.37 ± 1.17
	15 p	$0.51\pm3.32e{-2}$	3.81 ± 0.24	13.30 ± 0.38	46.95 ± 2.29	72.51 ± 0.90
E-ABRUPT (ABEBC)	1 p	$2.00e{-9}\pm4.68e{-10}$	$2.41e{-5}\pm2.42e{-6}$	$1.36e{-2}\pm9.17e{-4}$	4.56 ± 0.18	21.55 ± 0.52
	5 p	$2.60e{-6}\pm5.31e{-6}$	0.16 ± 0.24	3.98 ± 0.53	7.38 ± 0.30	36.81 ± 0.76
	15 p	$3.53e{-2}\pm3.00e{-2}$	0.93 ± 0.25	5.27 ± 0.41	16.11 ± 2.57	30.95 ± 0.64
E-ABRUPT (ITEL)	1 p	2.00 ± 0.15	13.58 ± 0.66	40.91 ± 0.89	135.13 ± 1.15	199.65 ± 0.27
	5 p	2.20 ± 0.20	15.69 ± 1.02	83.49 ± 3.62	156.20 ± 1.56	200.00 ± 0.00
	15 p	1.57 ± 0.12	18.48 ± 1.30	101.84 ± 4.82	185.10 ± 4.48	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.51\pm5.25e{-2}$	42.82 ± 0.13	76.56 ± 0.31	182.80 ± 0.77	273.86 ± 1.50
	5 p	$14.11\pm5.79e{-2}$	38.81 ± 0.12	68.71 ± 0.26	153.43 ± 1.03	236.03 ± 1.49
	15 p	$10.61\pm4.40e{-2}$	32.95 ± 0.14	59.24 ± 0.25	136.39 ± 0.86	200.00 ± 1.65
E-CHAOS (ABEBC)	1 p	$2.68\pm2.58e{-2}$	22.93 ± 0.16	61.64 ± 0.34	175.45 ± 0.75	268.42 ± 1.53
	5 p	$3.33\pm5.22e{-2}$	23.88 ± 0.15	57.68 ± 0.27	147.95 ± 1.02	231.77 ± 1.50
	15 p	$2.75\pm4.03e{-2}$	20.94 ± 0.15	50.12 ± 0.25	131.73 ± 0.85	196.53 ± 1.61
E-CHAOS (ITEL)	1 p	$1.37\pm1.62e{-2}$	$4.96\pm7.04e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.34\pm2.02e{-2}$	$4.95\pm1.09e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.08\pm7.43e{-3}$	$4.94\pm9.38e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.30\pm8.58e{-3}$	23.66 ± 0.14	82.87 ± 0.34	224.94 ± 0.46	323.69 ± 0.66
	5 p	$4.24\pm1.52e{-2}$	25.25 ± 0.13	71.06 ± 0.26	192.25 ± 0.48	285.50 ± 0.46
	15 p	$4.60\pm2.17e{-2}$	21.71 ± 0.12	60.46 ± 0.17	176.79 ± 0.38	243.78 ± 0.47

Table 8.2: Experimental results of reinitialising PSO - 30% reinitialisation

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$3.45e{-2}\pm4.23e{-3}$	$0.48\pm2.20e{-2}$	$2.31\pm9.16e{-2}$	17.41 ± 0.53	42.94 ± 0.88
	5 p	$3.67e{-2}\pm4.38e{-3}$	2.09 ± 0.93	2.39 ± 0.12	19.48 ± 0.49	51.56 ± 2.02
	15 p	$6.09e{-2}\pm8.41e{-3}$	$0.50\pm4.70e{-2}$	3.17 ± 0.62	23.80 ± 1.28	52.28 ± 1.65
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$1.19e{-6}\pm8.27e{-8}$	$1.22e{-3}\pm8.44e{-5}$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$7.57e{-2}\pm5.26e{-3}$	$0.34\pm2.36e{-2}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$5.21e{-15}\pm3.62e{-16}$	$1.98e{-5}\pm1.38e{-6}$	$0.39\pm2.69e{-2}$
E-PROGRESS (CME)	1 p	$1.94\pm1.46e{-2}$	10.22 ± 0.22	29.26 ± 1.32	121.49 ± 6.47	235.38 ± 10.54
	5 p	$2.56\pm3.50e{-2}$	12.96 ± 0.31	34.15 ± 1.70	141.84 ± 5.97	241.12 ± 6.08
	15 p	$2.70\pm1.95e{-2}$	14.45 ± 0.32	36.98 ± 0.96	135.32 ± 4.25	217.21 ± 2.33
E-ABRUPT (CME)	1 p	$0.38\pm1.09e{-2}$	$2.41\pm5.26e{-2}$	8.25 ± 0.15	33.53 ± 0.54	64.61 ± 0.87
	5 p	$0.48\pm1.43e{-2}$	$2.68\pm7.94e{-2}$	12.80 ± 0.44	38.67 ± 0.79	81.37 ± 1.33
	15 p	$0.47\pm2.63e{-2}$	3.40 ± 0.17	12.74 ± 0.39	46.09 ± 2.06	73.21 ± 1.22
E-ABRUPT (ABEBC)	1 p	$1.88e{-9}\pm3.90e{-10}$	$2.65e{-5}\pm2.30e{-6}$	$1.26e{-2}\pm6.62e{-4}$	4.74 ± 0.20	21.97 ± 0.51
	5 p	$8.91e{-9}\pm5.20e{-9}$	$4.91e{-5}\pm7.35e{-6}$	3.76 ± 0.58	7.75 ± 0.33	36.21 ± 1.00
	15 p	$2.63e{-2}\pm2.82e{-2}$	0.64 ± 0.18	4.79 ± 0.63	14.74 ± 2.24	30.77 ± 0.83
E-ABRUPT (ITEL)	1 p	1.38 ± 0.10	12.65 ± 0.42	40.76 ± 0.76	138.27 ± 1.55	199.69 ± 0.26
	5 p	1.54 ± 0.19	15.03 ± 0.70	81.22 ± 5.20	159.78 ± 1.72	200.00 ± 0.00
	15 p	$1.19\pm7.73e{-2}$	17.29 ± 1.20	97.92 ± 6.01	181.93 ± 4.70	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.30\pm3.99e{-2}$	43.79 ± 0.18	80.69 ± 0.24	202.54 ± 0.94	310.50 ± 2.27
	5 p	$13.83\pm5.78e{-2}$	38.93 ± 0.15	70.37 ± 0.27	162.56 ± 0.86	246.47 ± 1.26
	15 p	$10.52\pm4.74e{-2}$	32.87 ± 0.12	60.75 ± 0.22	144.25 ± 0.85	213.85 ± 1.09
E-CHAOS (ABEBC)	1 p	$2.62\pm3.08e{-2}$	24.32 ± 0.19	66.58 ± 0.28	196.03 ± 0.91	305.96 ± 2.27
	5 p	$3.08\pm4.21e{-2}$	24.08 ± 0.16	59.76 ± 0.27	157.68 ± 0.84	243.13 ± 1.24
	15 p	$2.61\pm3.32e{-2}$	20.84 ± 0.12	51.58 ± 0.23	140.03 ± 0.77	210.94 ± 1.08
E-CHAOS (ITEL)	1 p	$1.30\pm1.43e{-2}$	$4.97\pm8.34e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.26\pm1.65e{-2}$	$4.96\pm7.29e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.06\pm7.15e{-3}$	$4.94\pm1.06e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.58\pm1.59e{-2}$	26.25 ± 0.12	84.06 ± 0.31	225.37 ± 0.42	327.66 ± 0.54
	5 p	$4.47\pm2.11e{-2}$	27.74 ± 0.13	72.30 ± 0.19	193.52 ± 0.27	285.80 ± 0.36
	15 p	$4.89\pm2.69e{-2}$	23.68 ± 0.14	62.20 ± 0.15	175.81 ± 0.33	243.77 ± 0.33

Table 8.3: Experimental results of reinitialising PSO - 50% reinitialisation

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	3.80e−2±4.80e−3	0.49±2.61e−2	2.40±0.10	17.60±0.48	41.96±0.88
	5 p	3.89e−2±3.91e−3	1.55±0.93	2.44±0.10	18.77±0.53	49.06±1.69
	15 p	5.81e−2±1.04e−2	0.54±5.47e−2	3.69±1.07	24.45±1.58	52.01±1.44
E-STATIC (Error)	1 p	0.00±0.00	0.00±0.00	0.00±0.00	1.56e−6±1.09e−7	1.41e−3±9.79e−5
	5 p	0.00±0.00	0.16±1.14e−2	0.00±0.00	7.57e−2±5.26e−3	1.39e−3±9.65e−5
	15 p	0.00±0.00	0.00±0.00	3.72e−3±2.58e−4	1.63e−5±1.13e−6	6.34e−2±4.40e−3
E-PROGRESS (CME)	1 p	2.28±1.75e−2	15.49±0.34	43.42±3.45	190.22±7.92	306.02±6.42
	5 p	3.00±3.47e−2	17.67±0.57	48.91±1.67	178.27±2.64	272.50±1.39
	15 p	3.08±2.91e−2	18.05±0.42	48.56±1.21	161.84±1.95	233.60±0.92
E-ABRUPT (CME)	1 p	0.37±8.26e−3	2.32±5.49e−2	8.07±0.18	35.34±0.49	65.36±0.77
	5 p	0.46±8.91e−3	2.56±7.36e−2	13.06±0.85	40.45±0.69	83.48±1.02
	15 p	0.41±1.26e−2	3.27±0.16	12.56±0.54	47.99±2.44	74.69±0.97
E-ABRUPT (ABEBC)	1 p	1.77e−9±2.89e−10	2.58e−5±2.97e−6	1.34e−2±6.03e−4	5.08±0.17	22.36±0.49
	5 p	5.18e−9±1.36e−9	5.18e−5±9.12e−6	3.58±0.89	8.10±0.27	37.51±0.74
	15 p	5.87e−8±9.54e−8	0.69±0.16	4.47±0.71	15.09±2.50	31.41±0.80
E-ABRUPT (ITEL)	1 p	1.27±6.31e−2	12.51±0.47	41.13±0.85	143.66±1.31	199.82±0.22
	5 p	1.32±9.54e−2	14.25±0.84	81.05±7.54	163.12±1.51	200.00±0.00
	15 p	1.08±4.21e−2	16.37±0.91	95.47±7.44	184.58±4.34	200.00±0.00
E-CHAOS (CME)	1 p	21.23±3.24e−2	44.88±0.11	85.29±0.36	222.13±1.18	342.63±2.14
	5 p	13.73±6.08e−2	39.19±0.16	72.56±0.25	173.22±0.90	261.67±1.42
	15 p	10.51±4.87e−2	32.91±0.11	62.42±0.26	152.43±0.64	226.22±0.72
E-CHAOS (ABEBC)	1 p	2.61±3.97e−2	25.61±0.15	71.74±0.41	216.21±1.12	338.70±2.12
	5 p	2.85±3.27e−2	24.37±0.17	62.02±0.30	168.78±0.86	258.81±1.38
	15 p	2.52±3.08e−2	20.88±0.14	53.48±0.26	148.38±0.64	223.34±0.69
E-CHAOS (ITEL)	1 p	1.27±1.16e−2	4.98±5.43e−3	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	1.20±1.31e−2	4.97±7.79e−3	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	1.05±6.60e−3	4.94±8.94e−3	5.00±0.00	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	3.91±1.86e−2	28.18±0.13	86.87±0.24	227.81±0.29	330.84±0.41
	5 p	4.79±2.43e−2	29.92±0.12	74.29±0.33	195.42±0.29	287.39±0.37
	15 p	5.23±2.95e−2	25.59±0.10	64.26±0.20	177.17±0.25	245.20±0.21

Table 8.4: Experimental results of reinitialising PSO - 70% reinitialisation

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$3.70e{-2}\pm3.64e{-3}$	$0.46\pm2.66e{-2}$	2.40 ± 0.10	17.58 ± 0.55	43.50 ± 0.91
	5 p	$3.62e{-2}\pm5.30e{-3}$	2.04 ± 1.02	3.13 ± 1.25	18.67 ± 0.51	52.70 ± 1.81
	15 p	$5.88e{-2}\pm9.80e{-3}$	$0.48\pm3.78e{-2}$	4.23 ± 1.55	24.43 ± 0.99	53.29 ± 1.78
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$8.21e{-7}\pm5.70e{-8}$	$7.84e{-4}\pm5.45e{-5}$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$7.57e{-2}\pm5.26e{-3}$	$0.34\pm2.35e{-2}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-3}\pm2.58e{-4}$	$0.28\pm1.97e{-2}$	$0.16\pm1.13e{-2}$
E-PROGRESS (CME)	1 p	$2.68\pm2.04e{-2}$	19.80 ± 0.58	63.77 ± 3.94	216.27 ± 1.70	324.51 ± 0.74
	5 p	$3.39\pm3.28e{-2}$	20.60 ± 0.60	60.13 ± 1.91	190.32 ± 0.89	281.61 ± 0.59
	15 p	$3.41\pm3.76e{-2}$	20.16 ± 0.39	54.35 ± 1.58	169.78 ± 1.11	240.16 ± 0.38
E-ABRUPT (CME)	1 p	$0.36\pm8.05e{-3}$	$2.34\pm4.68e{-2}$	8.51 ± 0.15	37.37 ± 0.50	68.64 ± 0.88
	5 p	$0.44\pm8.81e{-3}$	$2.69\pm6.65e{-2}$	12.55 ± 0.50	42.36 ± 0.52	86.67 ± 1.06
	15 p	$0.41\pm1.23e{-2}$	3.11 ± 0.15	11.33 ± 0.53	50.80 ± 2.45	77.78 ± 1.27
E-ABRUPT (ABEBC)	1 p	$1.91e{-9}\pm3.11e{-10}$	$2.94e{-5}\pm2.32e{-6}$	$1.43e{-2}\pm6.52e{-4}$	5.31 ± 0.18	23.56 ± 0.59
	5 p	$5.42e{-9}\pm1.38e{-9}$	$1.30e{-2}\pm2.65e{-2}$	2.76 ± 0.74	8.36 ± 0.24	38.53 ± 0.77
	15 p	$4.07e{-3}\pm8.34e{-3}$	0.32 ± 0.17	2.12 ± 0.82	17.05 ± 2.64	33.65 ± 0.93
E-ABRUPT (ITEL)	1 p	$1.20\pm7.61e{-2}$	12.99 ± 0.49	42.90 ± 0.77	148.27 ± 1.30	200.00 ± 0.00
	5 p	$1.24\pm6.92e{-2}$	15.66 ± 0.69	74.83 ± 5.99	167.33 ± 1.45	200.00 ± 0.00
	15 p	$1.08\pm6.08e{-2}$	17.37 ± 0.91	72.19 ± 8.37	188.40 ± 4.03	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.13\pm4.83e{-2}$	46.52 ± 0.18	90.87 ± 0.40	242.09 ± 1.35	368.13 ± 2.20
	5 p	$13.60\pm5.25e{-2}$	39.84 ± 0.14	76.07 ± 0.29	185.07 ± 0.75	277.59 ± 1.58
	15 p	$10.47\pm4.96e{-2}$	33.32 ± 0.15	64.33 ± 0.25	161.26 ± 0.74	236.63 ± 0.95
E-CHAOS (ABEBC)	1 p	$2.52\pm3.70e{-2}$	27.23 ± 0.17	77.46 ± 0.39	236.81 ± 1.26	364.57 ± 2.21
	5 p	$2.65\pm4.49e{-2}$	24.97 ± 0.15	65.46 ± 0.26	180.86 ± 0.72	274.66 ± 1.49
	15 p	$2.47\pm2.72e{-2}$	21.18 ± 0.17	55.25 ± 0.25	157.20 ± 0.74	233.78 ± 0.91
E-CHAOS (ITEL)	1 p	$1.25\pm1.64e{-2}$	$4.99\pm4.66e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.15\pm1.13e{-2}$	$4.97\pm5.50e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.04\pm6.09e{-3}$	$4.95\pm9.86e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$4.35\pm2.56e{-2}$	$30.16\pm9.98e{-2}$	90.89 ± 0.31	231.92 ± 0.40	335.21 ± 0.29
	5 p	$5.24\pm3.07e{-2}$	31.92 ± 0.14	76.83 ± 0.21	198.82 ± 0.33	290.47 ± 0.29
	15 p	$5.59\pm3.87e{-2}$	27.13 ± 0.12	66.42 ± 0.17	179.19 ± 0.26	247.29 ± 0.26

Table 8.5: Experimental results of reinitialising PSO - 90% reinitialisation

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$3.28e{-2}\pm3.99e{-3}$	$0.47\pm2.93e{-2}$	2.34 ± 0.12	17.97 ± 0.43	42.84 ± 0.94
	5 p	$3.98e{-2}\pm5.78e{-3}$	1.79 ± 0.77	2.43 ± 0.14	18.79 ± 0.60	52.47 ± 1.60
	15 p	$6.48e{-2}\pm1.10e{-2}$	$0.50\pm6.56e{-2}$	3.07 ± 0.93	23.30 ± 1.46	52.17 ± 1.56
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$1.10e{-6}\pm7.67e{-8}$	$7.67e{-4}\pm5.33e{-5}$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$2.27e{-5}\pm1.58e{-6}$	$0.34\pm2.36e{-2}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$0.21\pm1.45e{-2}$	$0.14\pm1.00e{-2}$
E-PROGRESS (CME)	1 p	$3.05\pm3.16e{-2}$	22.29 ± 0.49	77.27 ± 2.15	225.15 ± 0.71	332.64 ± 0.51
	5 p	$3.66\pm4.09e{-2}$	22.64 ± 0.53	67.80 ± 1.14	196.34 ± 0.87	287.96 ± 0.38
	15 p	$3.61\pm3.49e{-2}$	21.23 ± 0.43	58.74 ± 1.10	174.79 ± 0.50	244.35 ± 0.30
E-ABRUPT (CME)	1 p	$0.36\pm6.81e{-3}$	$2.52\pm4.20e{-2}$	9.43 ± 0.16	41.03 ± 0.60	72.58 ± 0.89
	5 p	$0.45\pm8.78e{-3}$	$2.84\pm7.98e{-2}$	11.41 ± 0.50	45.65 ± 0.58	90.54 ± 1.10
	15 p	$0.40\pm1.31e{-2}$	3.03 ± 0.11	10.29 ± 0.28	51.39 ± 1.61	80.30 ± 1.22
E-ABRUPT (ABEBC)	1 p	$2.55e{-9}\pm3.90e{-10}$	$3.05e{-5}\pm2.32e{-6}$	$1.65e{-2}\pm1.17e{-3}$	6.01 ± 0.13	25.11 ± 0.65
	5 p	$4.95e{-9}\pm9.73e{-10}$	$1.31e{-2}\pm2.65e{-2}$	0.52 ± 0.56	9.17 ± 0.30	40.53 ± 0.86
	15 p	$4.07e{-3}\pm8.34e{-3}$	0.21 ± 0.12	0.34 ± 0.20	14.71 ± 1.63	34.24 ± 0.89
E-ABRUPT (ITEL)	1 p	$1.12\pm5.41e{-2}$	14.19 ± 0.59	46.60 ± 0.81	159.31 ± 1.54	200.00 ± 0.00
	5 p	$1.25\pm7.78e{-2}$	16.57 ± 0.65	57.64 ± 4.28	175.47 ± 1.23	200.00 ± 0.00
	15 p	$1.02\pm2.28e{-2}$	17.45 ± 0.59	54.10 ± 2.23	187.92 ± 2.53	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.15\pm5.03e{-2}$	49.45 ± 0.17	100.55 ± 0.39	262.18 ± 1.48	388.88 ± 3.05
	5 p	$13.53\pm3.95e{-2}$	40.95 ± 0.12	80.95 ± 0.36	196.89 ± 0.91	292.45 ± 1.71
	15 p	$10.45\pm5.12e{-2}$	34.14 ± 0.14	67.77 ± 0.26	169.79 ± 0.60	245.22 ± 1.53
E-CHAOS (ABEBC)	1 p	$2.52\pm2.48e{-2}$	29.58 ± 0.20	86.89 ± 0.40	257.08 ± 1.44	385.46 ± 2.96
	5 p	$2.58\pm3.80e{-2}$	25.80 ± 0.15	70.11 ± 0.31	192.75 ± 0.85	289.51 ± 1.68
	15 p	$2.42\pm3.94e{-2}$	21.78 ± 0.15	58.40 ± 0.23	165.73 ± 0.62	242.43 ± 1.42
E-CHAOS (ITEL)	1 p	$1.24\pm1.34e{-2}$	$4.99\pm3.09e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.14\pm1.07e{-2}$	$4.98\pm5.21e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.05\pm7.81e{-3}$	$4.96\pm9.49e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$4.82\pm2.86e{-2}$	32.51 ± 0.13	94.91 ± 0.32	235.94 ± 0.42	340.22 ± 0.36
	5 p	$5.65\pm3.33e{-2}$	33.83 ± 0.15	79.30 ± 0.19	201.90 ± 0.26	293.93 ± 0.28
	15 p	$5.90\pm2.69e{-2}$	28.34 ± 0.15	68.37 ± 0.20	181.19 ± 0.28	249.98 ± 0.22

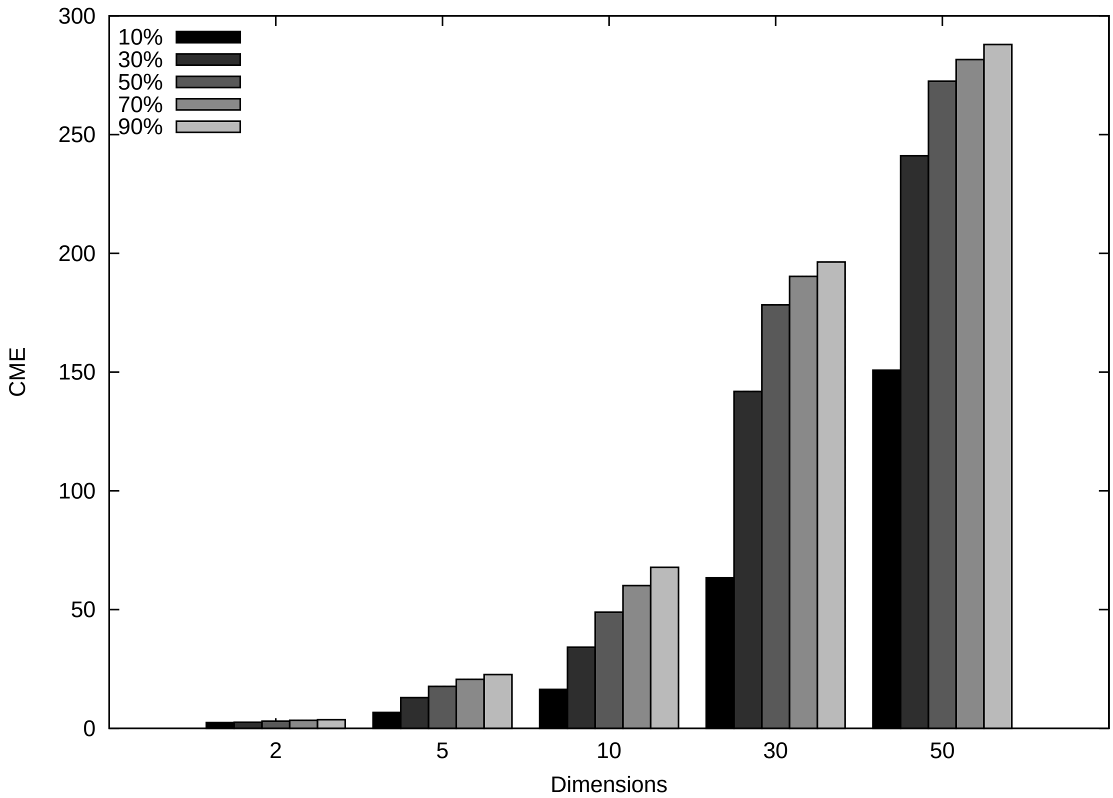
8.2.2 Results Analysis

The p-values determining the significance of the difference between the results of the re-evaluating PSO and those of the reinitialising PSO are listed in tables C.2 to C.6.

For all test environments, the performance level decreased as dimensionality increased.

For E-STATIC, the reinitialising PSO performed similarly to the standard PSO since no reinitialisation was triggered. The percentage of the swarm being reinitialised does not matter for a static environment, since the reinitialising PSO behaved exactly like a standard PSO.

For E-PROGRESS the average error was higher than for E-STATIC. The algorithm reinitialising 10% of the swarm gave the best performance and the CME increased dramatically with the reinitialisation ratio and with the dimensionality. Graph 8.1 highlights the influence that the percentage of reinitialisation had on the performance for the E-PROGRESS environments with five peaks. The CME was lower for unimodal than for multimodal environments. The reinitialising PSO performed significantly worse than the re-evaluating PSO on all but the multimodal problems with two and five dimensions where the reinitialising PSO (10%) performed better. These results seem to confirm that for E-PROGRESS environments with low dimensionality, the re-evaluating PSO can miss the appearance of new peaks which reduces performance. In contrast with the re-evaluating PSO, the reinitialising PSO relocates particles in all parts of the search space immediately after a change which allows for the detection of new peaks for the multimodal E-PROGRESS with two and five dimensions. However, the frequent changes in E-PROGRESS environments cause frequent reinitialisations of a portion of the swarm. As the size of the search space increases with the dimensionality, the probability for a particle to be relocated in the proximity of an optimum decreases as the dimensionality increases. Reinitialising a larger portion of the swarm further reduces the exploitation capacity of the swarm as more particles are relocated to a random location. Since the reinitialised particles are selected randomly, using a large reinitialisation ratio

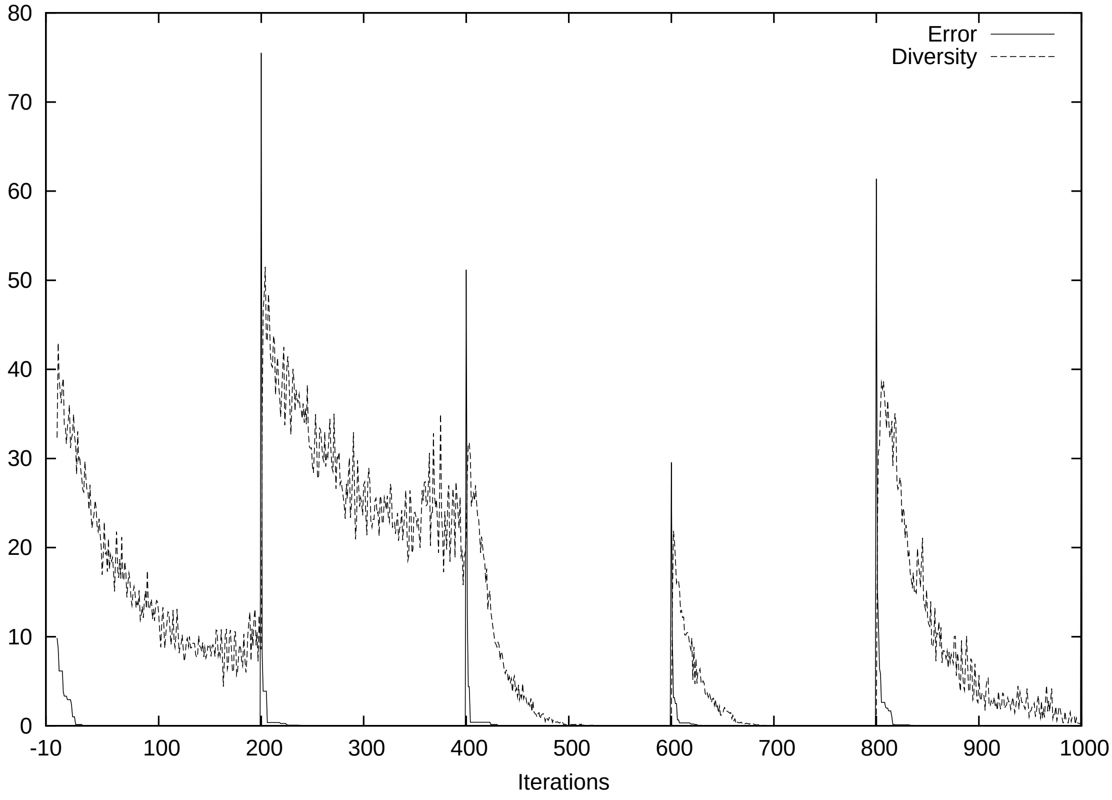


Graph 8.1: Reinitialising PSO with the various reinitialisation percentage for E-PROGRESS with five peaks

for an environment with a high change frequency is likely to cause all particles to be reinitialised within a few iterations. This limits the capacity of the algorithm to exploit the optima. Because the changes are not severe in the E-PROGRESS environments, an optimum remains close to its former position after a change and, therefore, randomly repositioning the particles anywhere in the search space is not an effective approach. In fact, the reinitialising PSO performed worst than the standard PSO on problems with 30 and 50 dimensions, and algorithms reinitialising 50% or more particles performed worst than the standard PSO on problems with 10 or more dimensions. A possible explanation is that, as the environment only undergoes small modifications, the particles with outdated memory of the standard PSO remained closer to the peak(s) than the randomly reinitialised particles of the reinitialising PSO.

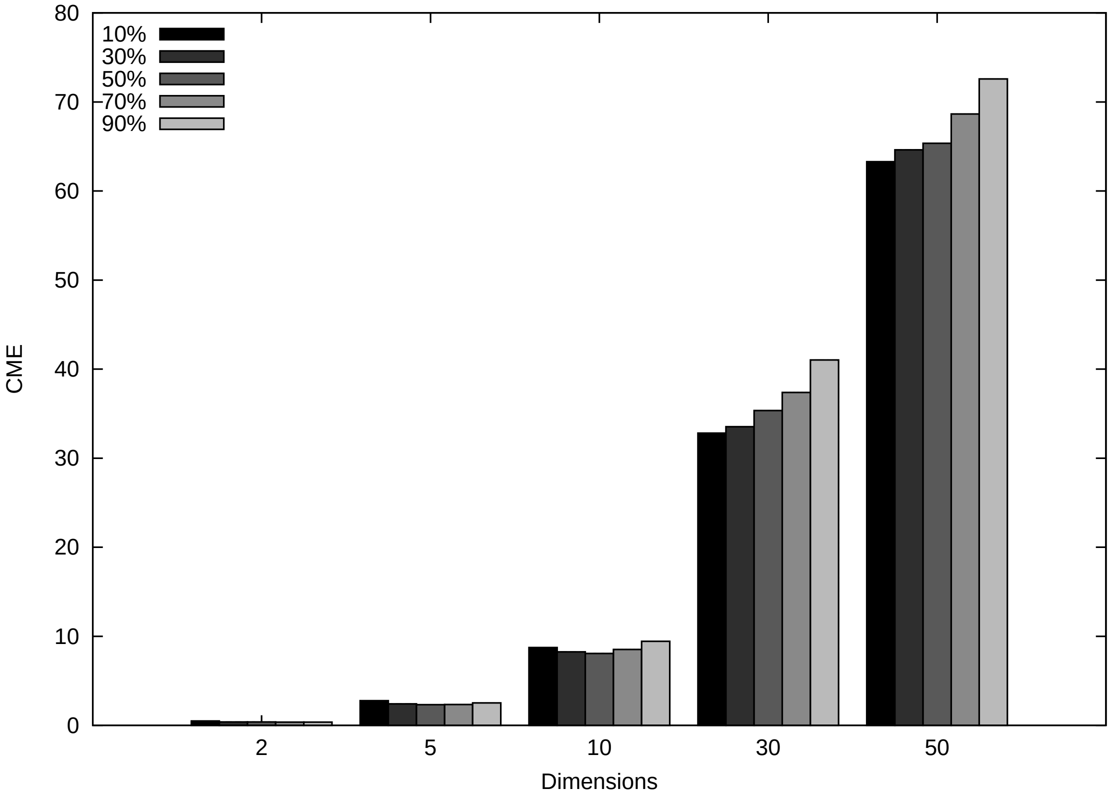
For E-ABRUPT, the average error (CME) was higher than for E-STATIC but, as expected, the algorithm performed significantly better than for E-PROGRESS. The reinitialisation ensures that the swarm has a high diversity after a change which promotes exploration when it is needed. Between two changes, the diversity decreased because the swarm behaved like a standard PSO and converged to exploit the optima. Graph 8.2 illustrates with an example the behaviour of a reinitialising PSO (50%) for an E-ABRUPT environment with five peaks in two dimensions. According to the results, reinitialising a higher percentage of the swarm tended to improve the performance level on problems with low dimensionality (10 dimensions or less), but reduced the performance on problems with more dimensions. Graph 8.3 highlights how the reinitialising ratio influenced the CME for the unimodal E-ABRUPT environments. The ABEBC measurements show that the error level before a change was significantly lower than the average error level (CME). The ITEL measurements are evidence that the reinitialising PSO was much quicker to find a good solution than the re-evaluating PSO for environments with low dimensionality. For these environments the diversity of the re-evaluating PSO increased progressively while the reinitialising PSO's diversity was high right after a change. Graph 8.4 highlights the difference in behaviour between re-evaluating and reinitialising PSO by comparing graphs 7.5 and 8.2.

As for E-ABRUPT, for E-CHAOS the algorithms reinitialising more particles performed better (considering CME, ABEBC and ITEL) on problems with less dimensions and worse on problems with higher dimensionality. However, for E-CHAOS, having a higher reinitialisation ratio only improved the performance for the two-dimensional environments. Graph 8.5 highlights how the reinitialising ratio influenced the CME for the E-CHAOS environments with five peaks. The reinitialising PSO performed better than the re-evaluating PSO on two-dimensional E-CHAOS but the re-evaluating PSO performed better on the rest of the problems. As was the case for E-PROGRESS, the frequent reinitialisation in a search space with high dimensionality caused the poor performance for E-CHAOS environments of high dimensionality. As for the re-evaluating PSO, the reinitialising PSO's performance improved proportionally to the number of peaks in the environment. This indicates that performance was improved by particles being randomly located on a peak.



Graph 8.2: Reinitialising PSO with 50% reinitialisation for E-ABRUPT

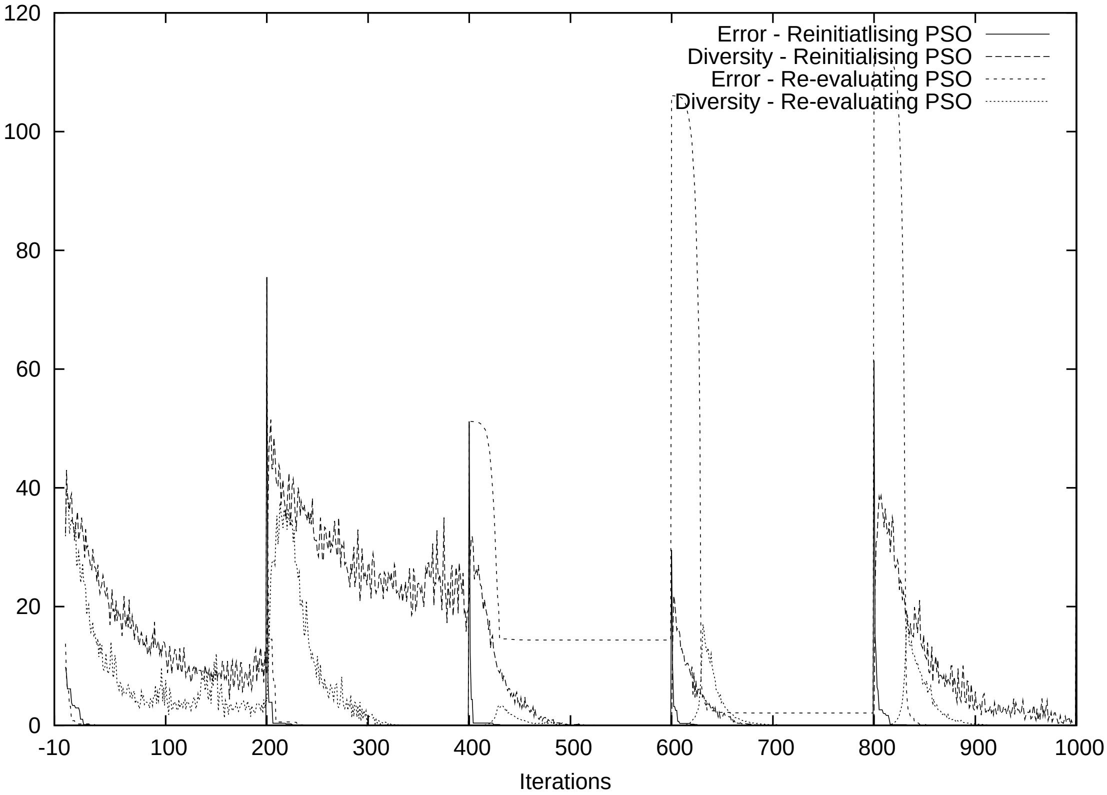
For the E-PATTERN environments, reinitialising 10% of the swarm gave the best performance, even outperforming the re-evaluating PSO for some E-PATTERN environments as can be seen in table C.2. However, the percentage of reinitialisation was generally proportional to the error level. Graph 8.6 highlights how the reinitialising ratio influenced the CME for the E-PATTERN environments with five peaks. As for the re-evaluating PSO, the presence of more peaks in the environment influenced the performance level negatively for two dimensional environments and positively for E-PATTERN environments with 10 dimensions or more.



Graph 8.3: Reinitialising PSO with the various reinitialising ratio for unimodal E-ABRUPT

8.2.3 Summary of Strengths and Weaknesses

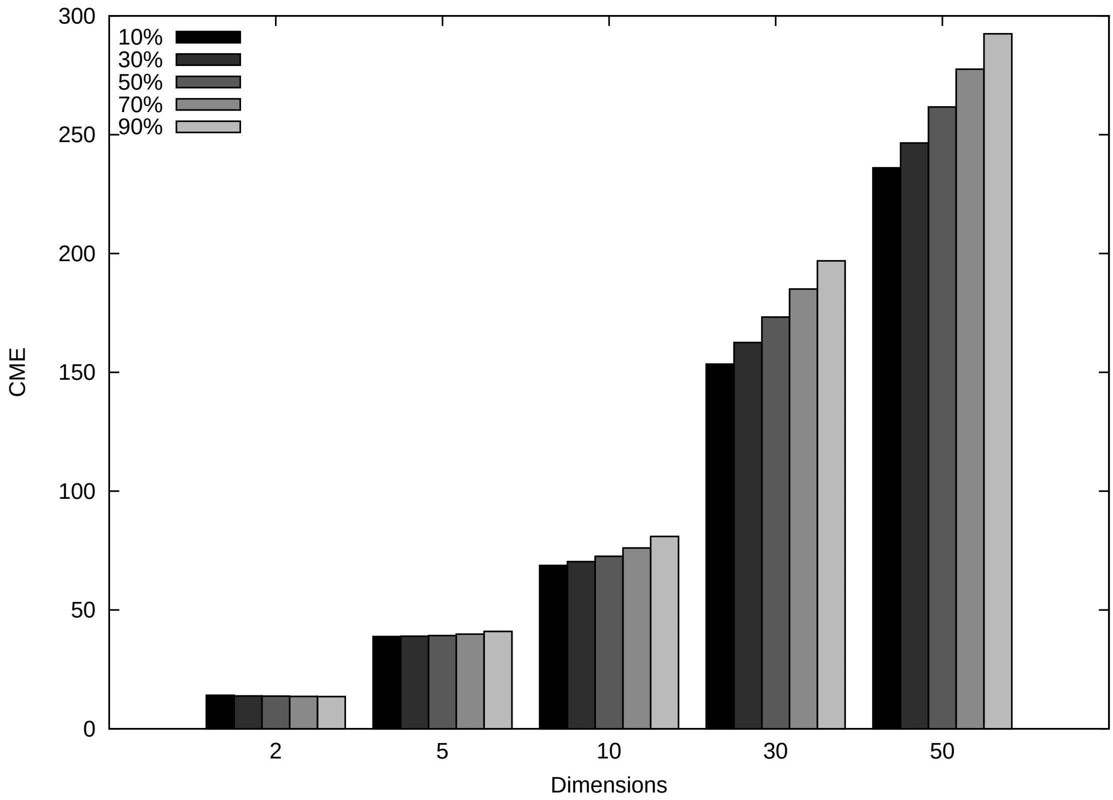
The reinitialising PSO was able to detect new optima but struggled to track peaks. The results confirmed that the reinitialising PSO is not suitable for progressively changing environments but performs well for abruptly changing environments and can be more efficient than the re-evaluating PSO for chaotically changing environments with low dimensionality. In comparison to the re-evaluating PSO, the reinitialising PSO showed to be generally more effective for problems of low dimensionality than on problems of high dimensionality.



Graph 8.4: Comparison between re-evaluating PSO and reinitialising PSO for E-ABRUPT

8.3 Atomic PSO

This section presents and analyses the results obtained by applying the APSO described in section 4.8.2 to the environment test cases. The APSO maintains diversity at all times using inter-particle repulsion. As mentioned in section 4.8.2, the APSO has proven more efficient than the CPSO. The APSO was evaluated with 10%, 30%, 50%, 70% and 90% of charged particles. The electrostatic parameters are set out according to [10] with a Q of 16 and a R_c of 1. A R_p of 30 was selected to only ignore the repulsion between particles that are far apart.



Graph 8.5: Reinitialising PSO with the various reinitialisation percentage for E-CHAOS with five peaks

8.3.1 Results

The results in tables 8.6 to 8.10 were obtained by an implementation of algorithm 4.2 using the parameters from section 6.4. The ratio of charged particles in the swarm is mentioned in the legend of the tables. The code for the APSO can be found in CIlib [74].

Table 8.6: Experimental results of APSO - 10% charged particles

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$2.44e{-2}\pm3.95e{-3}$	$0.42\pm2.43e{-2}$	$2.25\pm8.73e{-2}$	17.76 ± 0.45	42.23 ± 0.92
	5 p	$3.18e{-2}\pm3.25e{-3}$	1.76 ± 0.78	2.20 ± 0.11	19.31 ± 1.13	50.33 ± 1.88
	15 p	$5.61e{-2}\pm1.16e{-2}$	$0.48\pm6.38e{-2}$	2.81 ± 0.46	24.08 ± 1.22	51.37 ± 1.85
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$1.82e{-6}\pm1.26e{-7}$	$1.29e{-3}\pm8.93e{-5}$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$7.57e{-2}\pm5.26e{-3}$	$6.45e{-2}\pm4.48e{-3}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-3}\pm2.58e{-4}$	$5.08e{-5}\pm3.52e{-6}$	$0.39\pm2.70e{-2}$
E-PROGRESS (CME)	1 p	$1.68\pm2.46e{-3}$	$2.29\pm2.74e{-2}$	5.26 ± 0.10	25.12 ± 0.52	52.79 ± 1.16
	5 p	3.26 ± 0.17	7.00 ± 0.36	11.42 ± 1.29	28.24 ± 0.63	58.41 ± 1.26
	15 p	4.26 ± 0.55	10.66 ± 0.78	13.38 ± 2.20	34.51 ± 1.21	52.75 ± 1.19
E-ABRUPT (CME)	1 p	0.88 ± 0.41	3.55 ± 0.56	8.02 ± 0.21	32.77 ± 0.34	63.54 ± 0.91
	5 p	1.44 ± 0.68	3.80 ± 0.36	12.85 ± 0.51	36.78 ± 0.56	81.67 ± 1.37
	15 p	0.91 ± 0.30	4.39 ± 0.77	12.92 ± 0.33	48.01 ± 2.40	71.99 ± 1.11
E-ABRUPT (ABEBC)	1 p	$3.70e{-9}\pm2.76e{-9}$	$2.76e{-5}\pm4.54e{-6}$	$1.23e{-2}\pm9.09e{-4}$	4.56 ± 0.13	21.75 ± 0.57
	5 p	0.20 ± 0.26	0.29 ± 0.22	3.85 ± 0.52	7.58 ± 0.24	37.23 ± 1.07
	15 p	0.20 ± 0.13	1.64 ± 0.73	5.19 ± 0.46	17.91 ± 2.53	31.01 ± 0.83
E-ABRUPT (ITEL)	1 p	3.75 ± 1.32	15.19 ± 1.59	37.33 ± 0.90	134.13 ± 1.09	199.24 ± 0.50
	5 p	10.29 ± 5.22	17.00 ± 0.95	81.14 ± 4.38	157.17 ± 1.46	200.00 ± 0.00
	15 p	3.88 ± 2.53	22.97 ± 7.59	101.73 ± 4.39	188.09 ± 5.29	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.59\pm4.73e{-2}$	42.26 ± 0.12	74.19 ± 0.31	174.17 ± 0.97	257.48 ± 1.34
	5 p	$14.29\pm6.47e{-2}$	38.80 ± 0.18	67.65 ± 0.40	157.40 ± 2.00	227.16 ± 1.74
	15 p	$10.68\pm3.67e{-2}$	33.12 ± 0.18	59.32 ± 0.33	132.69 ± 1.43	196.22 ± 2.63
E-CHAOS (ABEBC)	1 p	$2.63\pm3.41e{-2}$	22.02 ± 0.15	58.82 ± 0.33	166.15 ± 0.95	251.58 ± 1.33
	5 p	$3.51\pm4.99e{-2}$	23.80 ± 0.19	56.34 ± 0.38	150.96 ± 1.88	222.37 ± 1.71
	15 p	$2.84\pm3.50e{-2}$	21.08 ± 0.18	49.84 ± 0.33	127.66 ± 1.38	192.32 ± 2.59
E-CHAOS (ITEL)	1 p	$1.40\pm1.33e{-2}$	$4.95\pm9.81e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.41\pm2.09e{-2}$	$4.94\pm1.05e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.10\pm7.89e{-3}$	$4.93\pm9.43e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.17\pm3.51e{-3}$	21.00 ± 0.10	86.69 ± 0.37	217.02 ± 0.63	292.70 ± 0.88
	5 p	$4.15\pm4.22e{-3}$	23.90 ± 0.15	73.27 ± 0.24	186.16 ± 0.80	272.64 ± 0.76
	15 p	5.19 ± 0.13	20.51 ± 0.14	59.90 ± 0.20	178.34 ± 0.47	239.16 ± 0.67

Table 8.7: Experimental results of APSO - 30% charged particles

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$2.54e{-}2 \pm 3.02e{-}3$	$0.43 \pm 3.12e{-}2$	$2.21 \pm 7.59e{-}2$	17.47 ± 0.48	42.66 ± 0.97
	5 p	$2.62e{-}2 \pm 3.93e{-}3$	1.43 ± 0.79	$2.33 \pm 9.90e{-}2$	19.02 ± 0.69	51.57 ± 2.07
	15 p	$5.58e{-}2 \pm 1.15e{-}2$	0.67 ± 0.37	4.60 ± 1.57	25.06 ± 1.02	52.45 ± 1.67
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$4.53e{-}6 \pm 3.14e{-}7$	$1.20e{-}3 \pm 8.33e{-}5$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$7.57e{-}2 \pm 5.26e{-}3$	$1.39e{-}3 \pm 9.63e{-}5$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-}3 \pm 2.58e{-}4$	$0.28 \pm 1.97e{-}2$	$0.23 \pm 1.63e{-}2$
E-PROGRESS (CME)	1 p	$1.68 \pm 3.24e{-}3$	$2.36 \pm 2.54e{-}2$	5.51 ± 0.11	25.44 ± 0.57	52.97 ± 1.14
	5 p	$2.98 \pm 3.61e{-}2$	7.01 ± 0.30	11.99 ± 1.33	29.00 ± 0.78	58.22 ± 1.22
	15 p	2.94 ± 0.11	9.59 ± 0.80	13.83 ± 2.33	33.72 ± 1.19	53.75 ± 1.00
E-ABRUPT (CME)	1 p	$0.44 \pm 1.44e{-}2$	$2.30 \pm 6.06e{-}2$	6.89 ± 0.16	32.81 ± 0.40	63.55 ± 0.95
	5 p	$0.58 \pm 2.28e{-}2$	2.81 ± 0.14	12.03 ± 0.39	37.29 ± 0.81	81.40 ± 1.39
	15 p	$0.57 \pm 4.95e{-}2$	4.18 ± 0.60	11.96 ± 0.19	48.22 ± 2.77	72.70 ± 1.12
E-ABRUPT (ABEBC)	1 p	$2.97e{-}9 \pm 5.04e{-}10$	$3.31e{-}5 \pm 3.27e{-}6$	$1.43e{-}2 \pm 1.21e{-}3$	5.08 ± 0.13	22.14 ± 0.58
	5 p	$1.08e{-}3 \pm 1.24e{-}3$	0.11 ± 0.15	4.03 ± 0.47	8.12 ± 0.30	37.21 ± 1.03
	15 p	$7.77e{-}2 \pm 4.96e{-}2$	1.51 ± 0.57	5.47 ± 0.25	18.24 ± 3.08	31.04 ± 0.94
E-ABRUPT (ITEL)	1 p	2.00 ± 0.14	10.85 ± 0.55	34.30 ± 0.81	136.93 ± 1.14	199.71 ± 0.32
	5 p	2.32 ± 0.21	13.41 ± 0.56	81.13 ± 3.31	159.59 ± 1.54	200.00 ± 0.00
	15 p	1.60 ± 0.17	20.15 ± 5.48	102.03 ± 2.20	185.95 ± 5.12	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.62 \pm 4.98e{-}2$	42.28 ± 0.14	74.31 ± 0.39	172.97 ± 1.04	256.71 ± 1.75
	5 p	$14.27 \pm 6.88e{-}2$	38.84 ± 0.12	68.03 ± 0.39	156.71 ± 1.80	226.35 ± 1.57
	15 p	$10.71 \pm 4.76e{-}2$	33.14 ± 0.15	59.11 ± 0.22	132.07 ± 1.13	194.50 ± 1.84
E-CHAOS (ABEBC)	1 p	$2.71 \pm 3.96e{-}2$	22.05 ± 0.18	58.88 ± 0.44	165.00 ± 1.03	250.76 ± 1.72
	5 p	$3.56 \pm 4.74e{-}2$	23.74 ± 0.15	56.66 ± 0.41	150.30 ± 1.70	221.70 ± 1.58
	15 p	$2.86 \pm 4.10e{-}2$	21.10 ± 0.17	49.65 ± 0.25	127.02 ± 1.11	190.59 ± 1.83
E-CHAOS (ITEL)	1 p	$1.39 \pm 1.78e{-}2$	$4.95 \pm 9.74e{-}3$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.42 \pm 2.02e{-}2$	$4.95 \pm 8.83e{-}3$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.10 \pm 1.27e{-}2$	$4.94 \pm 8.32e{-}3$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.17 \pm 3.85e{-}3$	20.96 ± 0.14	86.55 ± 0.30	217.93 ± 0.65	293.06 ± 0.90
	5 p	$4.15 \pm 4.39e{-}3$	23.99 ± 0.13	73.24 ± 0.30	186.39 ± 0.66	271.90 ± 0.94
	15 p	$4.94 \pm 1.85e{-}2$	20.51 ± 0.13	60.05 ± 0.15	178.67 ± 0.45	238.74 ± 0.81

Table 8.8: Experimental results of APSO - 50% charged particles

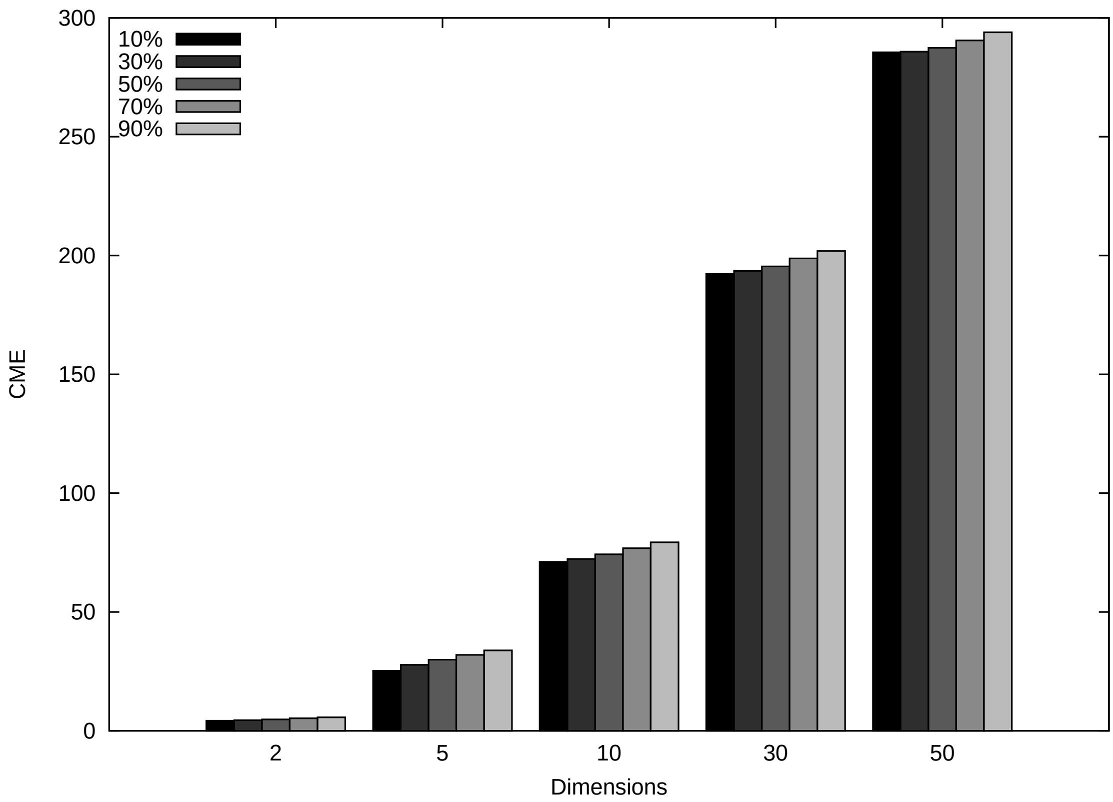
DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$2.99e{-2}\pm3.15e{-3}$	$0.45\pm2.72e{-2}$	$2.32\pm7.82e{-2}$	17.60 ± 0.46	43.21 ± 1.04
	5 p	$3.40e{-2}\pm4.60e{-3}$	2.15 ± 0.85	2.55 ± 0.19	19.16 ± 0.36	51.65 ± 1.82
	15 p	$6.11e{-2}\pm1.21e{-2}$	$0.48\pm3.51e{-2}$	3.53 ± 1.01	24.28 ± 1.22	53.72 ± 1.84
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$1.21e{-5}\pm8.43e{-7}$	$6.78e{-3}\pm4.71e{-4}$
	5 p	0.00 ± 0.00	$0.16\pm1.14e{-2}$	$6.65e{-13}\pm4.61e{-14}$	$7.57e{-2}\pm5.26e{-3}$	$1.22e{-2}\pm8.45e{-4}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-3}\pm2.58e{-4}$	$3.31e{-4}\pm2.30e{-5}$	$9.50e{-3}\pm6.60e{-4}$
E-PROGRESS (CME)	1 p	$1.69\pm4.04e{-3}$	$2.50\pm3.15e{-2}$	5.73 ± 0.11	26.06 ± 0.50	54.44 ± 0.64
	5 p	$2.92\pm2.13e{-2}$	7.06 ± 0.31	11.06 ± 1.23	30.23 ± 0.58	59.70 ± 1.47
	15 p	$2.76\pm8.09e{-2}$	8.85 ± 0.51	13.72 ± 2.22	35.32 ± 1.33	53.34 ± 1.27
E-ABRUPT (CME)	1 p	$0.39\pm6.52e{-3}$	$2.17\pm3.77e{-2}$	6.74 ± 0.12	33.21 ± 0.42	64.16 ± 1.07
	5 p	$0.56\pm1.46e{-2}$	2.77 ± 0.13	11.81 ± 0.33	38.29 ± 0.80	81.07 ± 0.84
	15 p	$0.50\pm2.05e{-2}$	4.40 ± 0.69	11.94 ± 0.17	48.30 ± 1.90	72.54 ± 1.19
E-ABRUPT (ABEBC)	1 p	$5.36e{-9}\pm1.08e{-9}$	$7.90e{-5}\pm9.92e{-6}$	$2.47e{-2}\pm2.58e{-3}$	5.76 ± 0.18	23.24 ± 0.60
	5 p	$1.08e{-3}\pm1.24e{-3}$	0.11 ± 0.15	4.05 ± 0.41	9.17 ± 0.37	37.78 ± 0.69
	15 p	$1.53e{-2}\pm1.81e{-2}$	1.79 ± 0.64	5.47 ± 0.23	18.29 ± 2.23	32.20 ± 0.91
E-ABRUPT (ITEL)	1 p	$1.55\pm9.95e{-2}$	10.02 ± 0.48	34.23 ± 0.74	142.01 ± 1.47	200.00 ± 0.00
	5 p	1.98 ± 0.17	13.77 ± 0.78	82.17 ± 3.46	165.93 ± 1.80	200.00 ± 0.00
	15 p	$1.37\pm8.72e{-2}$	22.51 ± 6.50	101.43 ± 2.82	191.93 ± 3.39	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.64\pm4.30e{-2}$	42.34 ± 0.14	74.22 ± 0.25	174.01 ± 1.02	256.27 ± 1.53
	5 p	$14.27\pm8.01e{-2}$	38.73 ± 0.14	67.56 ± 0.24	156.40 ± 1.65	226.29 ± 1.46
	15 p	$10.71\pm3.97e{-2}$	33.12 ± 0.14	59.14 ± 0.32	132.74 ± 1.34	195.96 ± 2.79
E-CHAOS (ABEBC)	1 p	$2.76\pm4.17e{-2}$	22.10 ± 0.18	58.74 ± 0.31	165.96 ± 1.00	250.37 ± 1.53
	5 p	$3.52\pm6.46e{-2}$	23.76 ± 0.14	56.18 ± 0.26	150.05 ± 1.58	221.55 ± 1.46
	15 p	$2.88\pm3.20e{-2}$	21.09 ± 0.13	49.61 ± 0.31	127.64 ± 1.31	192.03 ± 2.76
E-CHAOS (ITEL)	1 p	$1.40\pm1.45e{-2}$	$4.95\pm1.16e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.40\pm2.00e{-2}$	$4.95\pm1.17e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.10\pm1.12e{-2}$	$4.93\pm1.34e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.19\pm5.16e{-3}$	$20.98\pm9.42e{-2}$	86.67 ± 0.34	217.36 ± 0.73	292.86 ± 0.90
	5 p	$4.16\pm6.50e{-3}$	23.92 ± 0.16	73.60 ± 0.40	186.48 ± 0.61	272.45 ± 0.87
	15 p	$4.82\pm2.47e{-2}$	20.54 ± 0.13	59.99 ± 0.20	178.31 ± 0.43	239.33 ± 0.85

Table 8.9: Experimental results of APSO - 70% charged particles

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$2.95e{-2}\pm3.69e{-3}$	$0.44\pm2.33e{-2}$	2.40 ± 0.13	18.47 ± 0.46	44.20 ± 0.85
	5 p	$3.31e{-2}\pm4.81e{-3}$	1.55 ± 0.61	3.17 ± 1.20	20.10 ± 0.60	52.87 ± 1.83
	15 p	$5.77e{-2}\pm1.30e{-2}$	$0.56\pm6.13e{-2}$	3.07 ± 0.55	25.56 ± 1.21	55.57 ± 1.46
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	$5.09e{-12}\pm3.54e{-13}$	$1.75e{-4}\pm1.21e{-5}$	$6.98e{-2}\pm4.85e{-3}$
	5 p	0.00 ± 0.00	$4.74e{-16}\pm3.29e{-17}$	$9.19e{-7}\pm6.38e{-8}$	$7.73e{-2}\pm5.37e{-3}$	$8.84e{-2}\pm6.14e{-3}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-3}\pm2.58e{-4}$	$0.33\pm2.27e{-2}$	$0.46\pm3.20e{-2}$
E-PROGRESS (CME)	1 p	$1.69\pm4.07e{-3}$	$2.72\pm2.75e{-2}$	$6.20\pm9.56e{-2}$	26.84 ± 0.55	54.66 ± 0.87
	5 p	$2.88\pm3.16e{-2}$	7.64 ± 1.02	12.46 ± 1.22	30.00 ± 0.56	59.26 ± 1.54
	15 p	$2.61\pm3.62e{-2}$	9.20 ± 0.39	14.26 ± 2.07	35.90 ± 0.93	54.47 ± 0.90
E-ABRUPT (CME)	1 p	$0.38\pm6.87e{-3}$	$2.16\pm3.99e{-2}$	6.90 ± 0.11	34.10 ± 0.47	65.01 ± 0.77
	5 p	$0.54\pm1.74e{-2}$	$2.71\pm8.75e{-2}$	11.74 ± 0.36	38.85 ± 0.57	82.43 ± 1.08
	15 p	$0.54\pm3.08e{-2}$	3.98 ± 0.53	12.03 ± 0.31	47.80 ± 2.63	73.59 ± 1.22
E-ABRUPT (ABEBC)	1 p	$4.73e{-8}\pm3.44e{-8}$	$6.36e{-4}\pm3.83e{-4}$	$0.13\pm3.87e{-2}$	7.43 ± 0.21	24.50 ± 0.51
	5 p	$2.25e{-3}\pm1.63e{-3}$	$1.66e{-2}\pm1.23e{-2}$	3.84 ± 0.52	10.63 ± 0.26	39.38 ± 0.74
	15 p	$4.44e{-2}\pm3.14e{-2}$	1.40 ± 0.43	5.20 ± 0.46	19.82 ± 2.59	33.63 ± 0.76
E-ABRUPT (ITEL)	1 p	$1.31\pm6.99e{-2}$	9.89 ± 0.35	34.25 ± 0.89	152.15 ± 2.07	200.00 ± 0.00
	5 p	1.73 ± 0.16	13.17 ± 0.82	82.36 ± 4.53	173.43 ± 1.87	200.00 ± 0.00
	15 p	$1.20\pm8.10e{-2}$	17.40 ± 4.66	97.66 ± 4.93	192.85 ± 3.17	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.64\pm4.86e{-2}$	42.37 ± 0.13	74.05 ± 0.23	174.20 ± 0.88	257.41 ± 1.62
	5 p	$14.30\pm7.46e{-2}$	38.88 ± 0.14	68.00 ± 0.32	157.65 ± 1.85	226.26 ± 1.31
	15 p	$10.74\pm5.21e{-2}$	33.14 ± 0.13	59.21 ± 0.20	132.12 ± 1.24	197.66 ± 2.47
E-CHAOS (ABEBC)	1 p	$2.86\pm5.03e{-2}$	22.24 ± 0.14	58.61 ± 0.25	166.28 ± 0.85	251.40 ± 1.60
	5 p	$3.56\pm5.33e{-2}$	23.80 ± 0.18	56.83 ± 0.32	151.17 ± 1.72	221.48 ± 1.31
	15 p	$2.93\pm4.68e{-2}$	21.06 ± 0.14	49.74 ± 0.19	127.08 ± 1.20	193.74 ± 2.42
E-CHAOS (ITEL)	1 p	$1.38\pm1.42e{-2}$	$4.95\pm6.57e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.39\pm1.60e{-2}$	$4.95\pm9.04e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.10\pm1.01e{-2}$	$4.93\pm1.09e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.23\pm7.92e{-3}$	21.12 ± 0.12	86.82 ± 0.40	216.96 ± 0.60	292.74 ± 1.00
	5 p	$4.21\pm7.67e{-3}$	23.93 ± 0.16	73.40 ± 0.28	186.68 ± 0.56	272.08 ± 0.84
	15 p	$4.72\pm2.49e{-2}$	20.62 ± 0.16	60.13 ± 0.20	178.09 ± 0.51	238.61 ± 0.72

Table 8.10: Experimental results of APSO - 90% charged particles

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	3.81e−2±4.96e−3	0.76±0.11	4.01±0.29	22.56±0.71	49.03±1.10
	5 p	3.79e−2±6.17e−3	2.17±0.83	4.11±0.33	23.73±0.72	57.10±1.93
	15 p	7.78e−2±1.08e−2	0.94±0.13	5.09±1.01	29.52±1.35	59.83±1.66
E-STATIC (Error)	1 p	4.74e−16±3.29e−17	1.03e−2±7.18e−4	1.61e−3±1.12e−4	0.24±1.67e−2	0.28±1.94e−2
	5 p	3.12e−6±2.16e−7	7.27e−3±5.05e−4	7.72e−2±5.36e−3	0.21±1.46e−2	0.45±3.11e−2
	15 p	0.00±0.00	2.31e−3±1.60e−4	3.99e−3±2.77e−4	0.41±2.82e−2	0.34±2.39e−2
E-PROGRESS (CME)	1 p	1.73±4.66e−3	3.22±3.45e−2	7.41±9.91e−2	28.62±0.54	56.64±0.85
	5 p	2.89±3.41e−2	7.73±0.13	13.99±1.29	32.06±0.56	61.68±0.93
	15 p	2.57±2.83e−2	8.11±0.53	17.87±2.52	37.89±1.31	57.79±1.25
E-ABRUPT (CME)	1 p	0.39±8.79e−3	2.64±0.11	8.42±0.20	35.71±0.41	65.59±0.72
	5 p	0.59±1.61e−2	3.30±0.16	13.24±0.35	41.10±0.71	83.15±1.01
	15 p	0.60±3.77e−2	6.22±0.97	13.69±0.39	47.27±2.14	74.40±1.03
E-ABRUPT (ABEBC)	1 p	3.16e−3±3.01e−3	0.37±0.11	2.06±0.30	10.49±0.30	27.12±0.52
	5 p	7.74e−3±3.02e−3	0.57±0.18	6.26±0.52	14.31±0.42	41.66±0.69
	15 p	6.76e−2±3.38e−2	3.51±0.95	7.49±0.44	20.67±2.15	36.45±0.80
E-ABRUPT (ITEL)	1 p	1.15±5.17e−2	9.82±0.47	36.04±0.82	184.29±2.87	200.00±0.00
	5 p	1.51±0.12	13.01±0.79	99.81±6.56	195.61±2.20	200.00±0.00
	15 p	1.17±6.83e−2	35.63±11.57	103.59±2.75	199.44±0.57	200.00±0.00
E-CHAOS (CME)	1 p	21.62±4.94e−2	42.35±0.11	74.45±0.27	174.45±1.10	257.20±1.42
	5 p	14.28±6.56e−2	38.83±0.17	67.80±0.39	158.09±1.88	226.83±1.84
	15 p	10.79±4.99e−2	33.18±0.14	59.15±0.32	132.32±1.16	195.85±2.58
E-CHAOS (ABEBC)	1 p	2.96±4.92e−2	22.24±0.14	59.15±0.28	166.48±1.04	251.33±1.47
	5 p	3.60±4.23e−2	23.72±0.18	56.41±0.39	151.59±1.77	222.02±1.81
	15 p	3.01±3.33e−2	21.19±0.16	49.70±0.26	127.25±1.15	191.93±2.53
E-CHAOS (ITEL)	1 p	1.37±1.38e−2	4.96±7.04e−3	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	1.37±1.56e−2	4.94±9.74e−3	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	1.10±1.07e−2	4.93±1.10e−2	5.00±0.00	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	3.34±1.07e−2	21.15±0.10	86.74±0.33	217.25±0.72	293.17±1.00
	5 p	4.30±8.20e−3	23.92±0.20	73.55±0.30	186.49±0.55	271.75±0.81
	15 p	4.73±2.11e−2	20.79±0.13	60.08±0.20	178.49±0.47	239.46±0.86



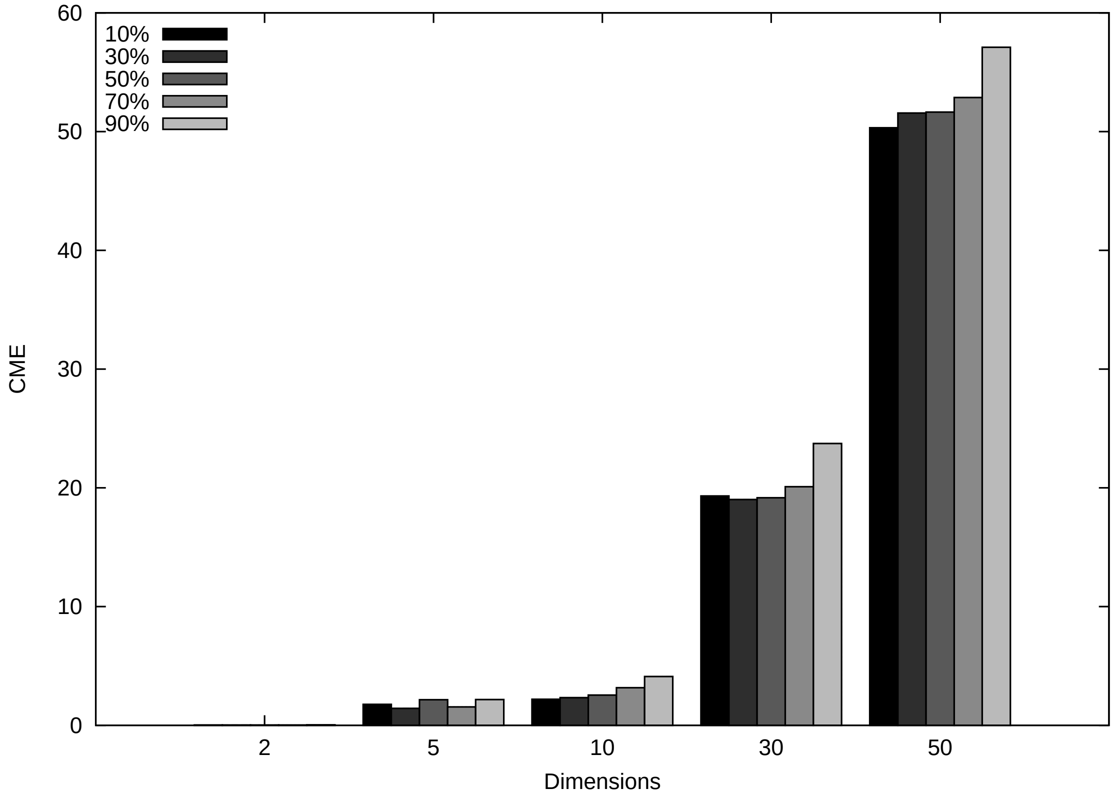
Graph 8.6: Reinitialising PSO with the various reinitialisation percentage for E-PATTERN with five peaks

8.3.2 Results Analysis

The p-values determining the significance of the difference between the results of the re-evaluating PSO and those of the APSO are listed in tables C.7 to C.11.

For all test environments the performance level of the APSO decreased as the dimensionality increased.

For E-STATIC, the results were close to those of the standard PSO. The CME tended to be lower for the unimodal problems than for the multimodal problems. For E-STATIC, swarms with 10% and 30% of charged particles gave the best results. Swarms with 50%



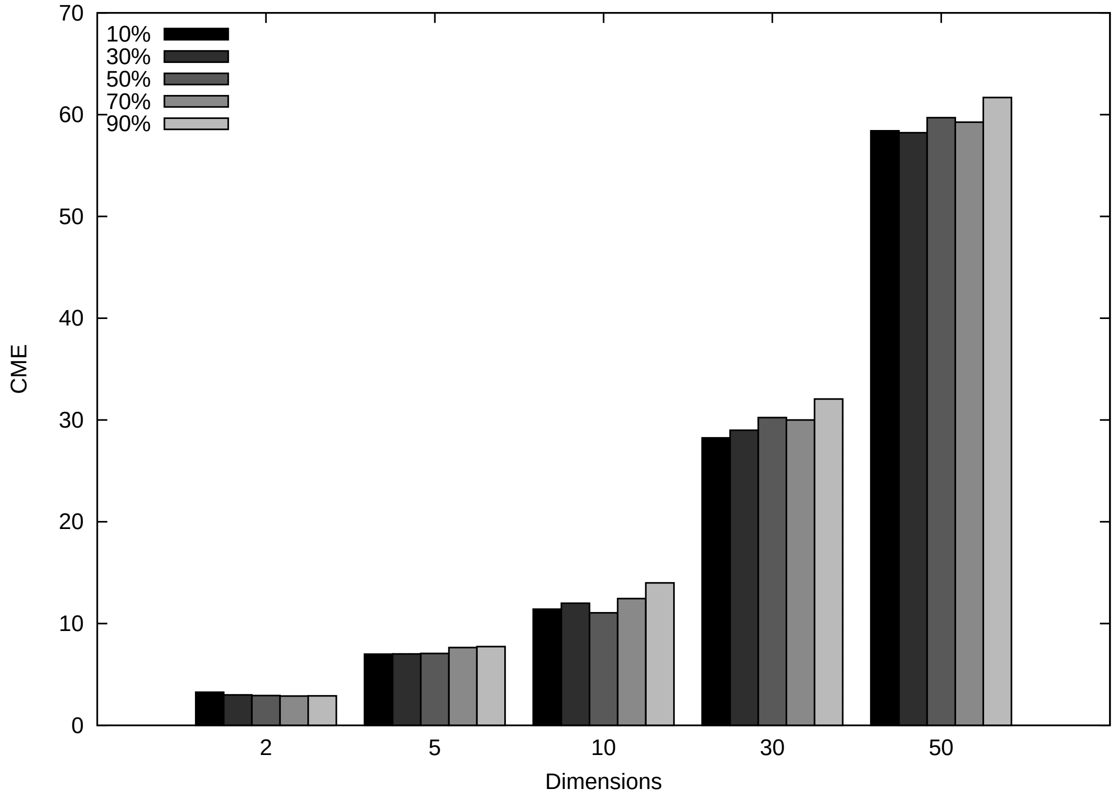
Graph 8.7: APSO with the various ratio of charged particles for E-STATIC with five peaks

of charged particles or less performed similarly to the standard PSO but the performance level tended to be poorer with a higher charged ratio. Indeed, both the CME and the final error show that the APSO with 90% of charged particles performed significantly worst than the standard PSO on most static problems and the APSO with 70% performed worst on some of the problems with a high number of dimensions. Graph 8.7 highlights the influence that the percentage of charged particles had on the performance for the E-STATIC environments with five peaks. Unlike the re-evaluating and reinitialising PSO, the APSO behaved differently to the standard PSO for static environments. The repulsion between the particles influences the particles' velocity at all time. Exploitation may therefore be slowed down when many particles are charged.

For all DEs, the performance level of the APSO (regardless of the charged ratio used) was better than that of the standard PSO.

For E-PROGRESS, the CME was higher than for E-STATIC. For the unimodal E-PROGRESS environments, the performance level was inversely proportional to the percentage of charged particles in the swarm. The re-evaluating PSO performed better than the APSO on these unimodal problems. For the multimodal environments, having a higher percentage of charged particles improved the performance on problems with low dimensionality but increased the error on high dimensionality problems. Regardless of the percentage of charged particles, the APSO performed significantly better than the re-evaluating PSO on two-dimensional multimodal E-PROGRESS environments. The APSO (70%) obtained the lowest CME on the two-dimensional problem with five peaks and the APSO (90%) obtained the lowest CME on the two-dimensional problem with 15 peaks. On the other problems, the performance level of the APSO (10%) was equivalent to that of the re-evaluating PSO but the CME tended to increase proportionally to the percentage of charged particles in the swarm. Graph 8.8 highlights how the charged ratio influenced the CME for the E-PROGRESS environments with five peaks.

For progressively changing environments, the charged particles cause the swarm to cover a larger area of the search space than with the re-evaluating PSO, but the exploration is limited to the areas surrounding the *gbest*. Figure 8.1 illustrates this dispersion pattern of the particles of a swarm with a charged ratio of 50% at iteration 542 for a two-dimensional E-PROGRESS environment with five peaks. The figure shows that most particles are located around the optimum and that as the distance from the optimum increases, less and less particles are present. Consequently, a peak appearing far from the *gbest* is less likely to be detected than a peak appearing closer. The better performance of the swarms with a higher charged ratio for the multimodal problems with low dimensionality can be explained by a better detection capacity. As shown in section 7.3.2, for E-PROGRESS environments with low dimensionality, the swarm of a re-evaluating PSO quickly converged towards a peak, then the area covered by the swarm remained small. In such condition, the additional diversity generated by the charged particles increases the detection capacity of the swarm. However, for environments with high dimensional-



Graph 8.8: APSO with the various ratio of charged particles for E-PROGRESS with five peaks

ity, the diversity of the re-evaluating PSO remained higher and having charged particles only slowed down the exploitation of the peaks.

For E-ABRUPT, the CME was higher than for E-STATIC but not necessarily higher than for E-PROGRESS. The CME was lower on unimodal problems but the algorithm did not perform systematically worse on problems with a higher number of peaks. APSO (50%) and (70%) obtained the lowest CME on problems with 10 dimensions or less. For environments with 30 and 50 dimensions, the influence on the CME of varying the number of charged particles was unclear from the results. However, the APSO (10%) performed best on most of the high dimensionality problems and the ABEBC tended to raise as the charged ratio increased. The APSO performed significantly better than

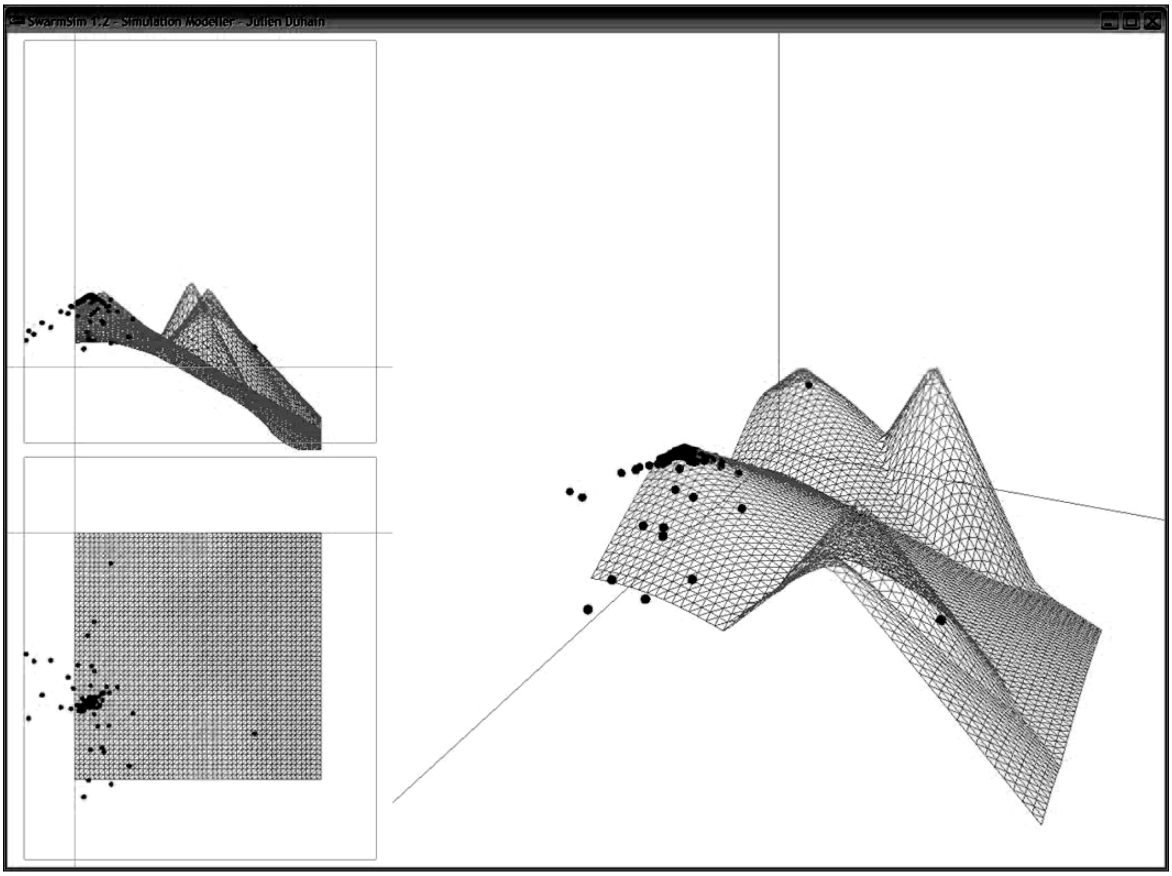
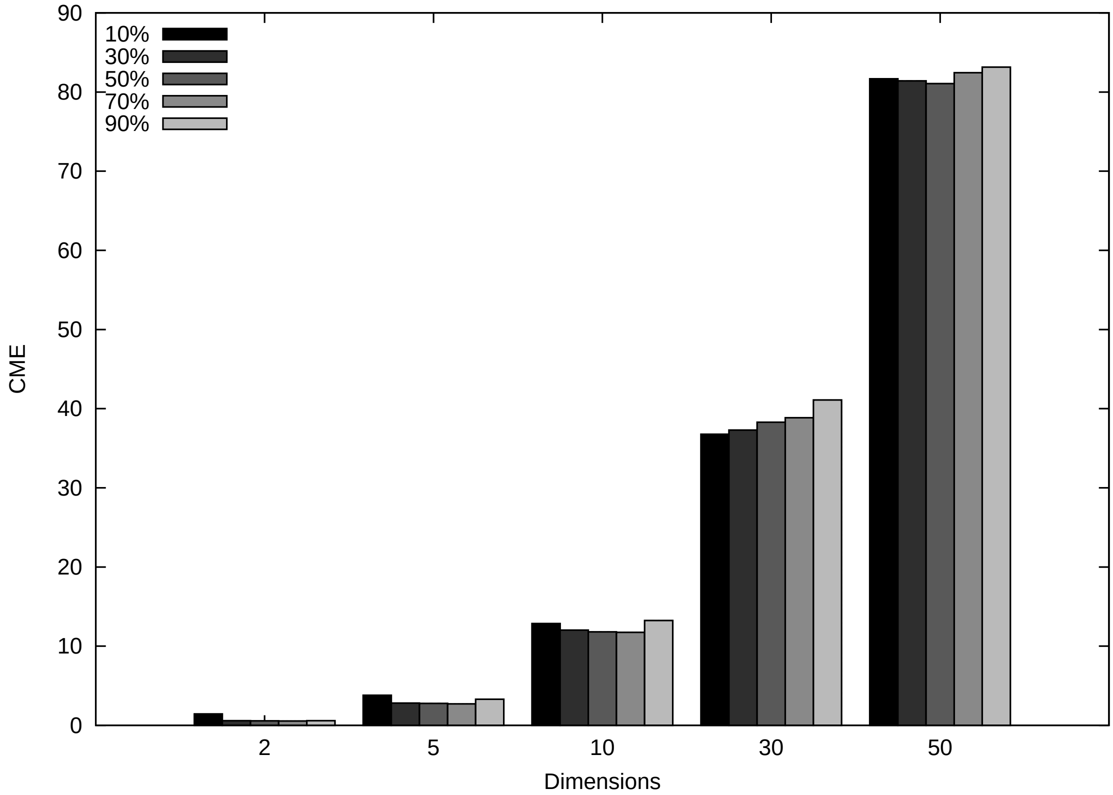


Figure 8.1: APSO for E-PROGRESS

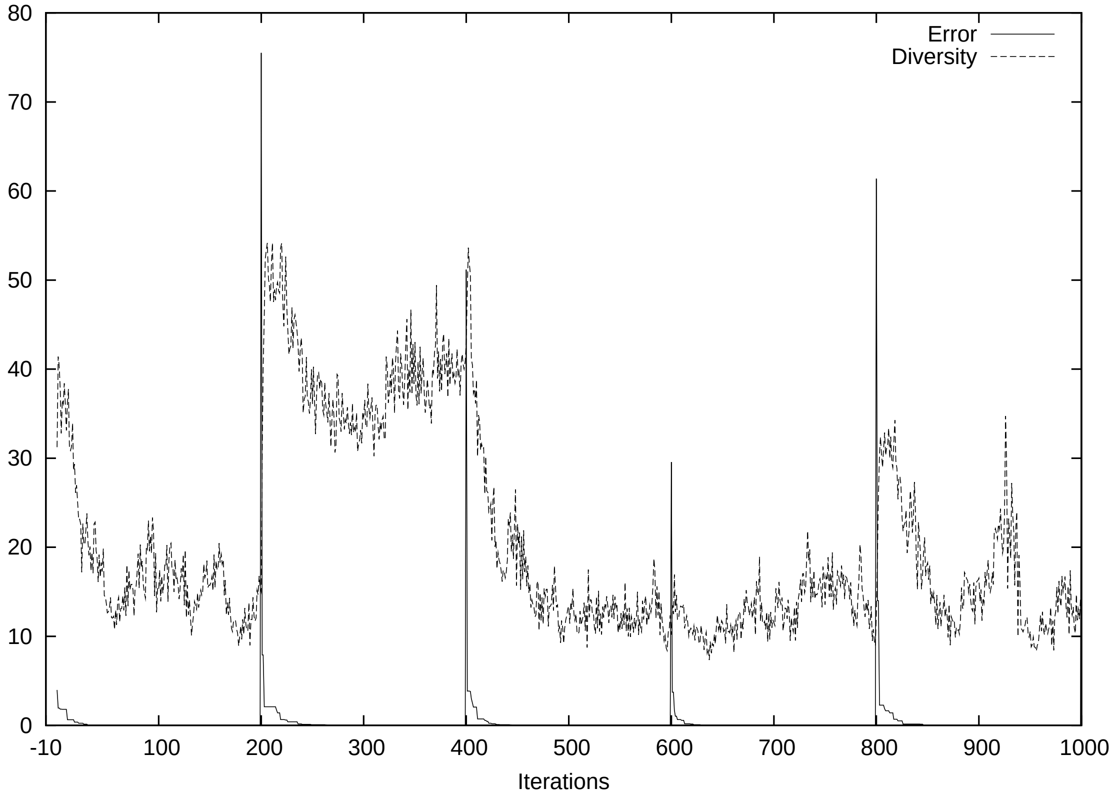
the re-evaluating PSO on low dimensionality problems (10 or less dimensions) but it was not necessarily the case for the rest of the environments. Graph 8.9 highlights the influence that the charged ratio had on the performance of the APSO for the E-ABRUPT environments with five peaks. For abruptly changing environments, the swarm never converges beyond a certain point while the environment remains static because of the repulsion between the charged particles. Since the APSO swarm always maintains diversity, the particles of the APSO have more velocity after a change than the particles of the re-evaluating PSO. This allows the APSO swarm to explore better and detect new peaks after a change. Graph 8.10 illustrates the progression of the error and diversity during a simulation where an APSO with 50% of charged particles is applied to a two-dimensional E-ABRUPT environment with five peaks. The diversity was maintained at



Graph 8.9: APSO with the various ratio of charged particles for E-ABRUPT with five peaks

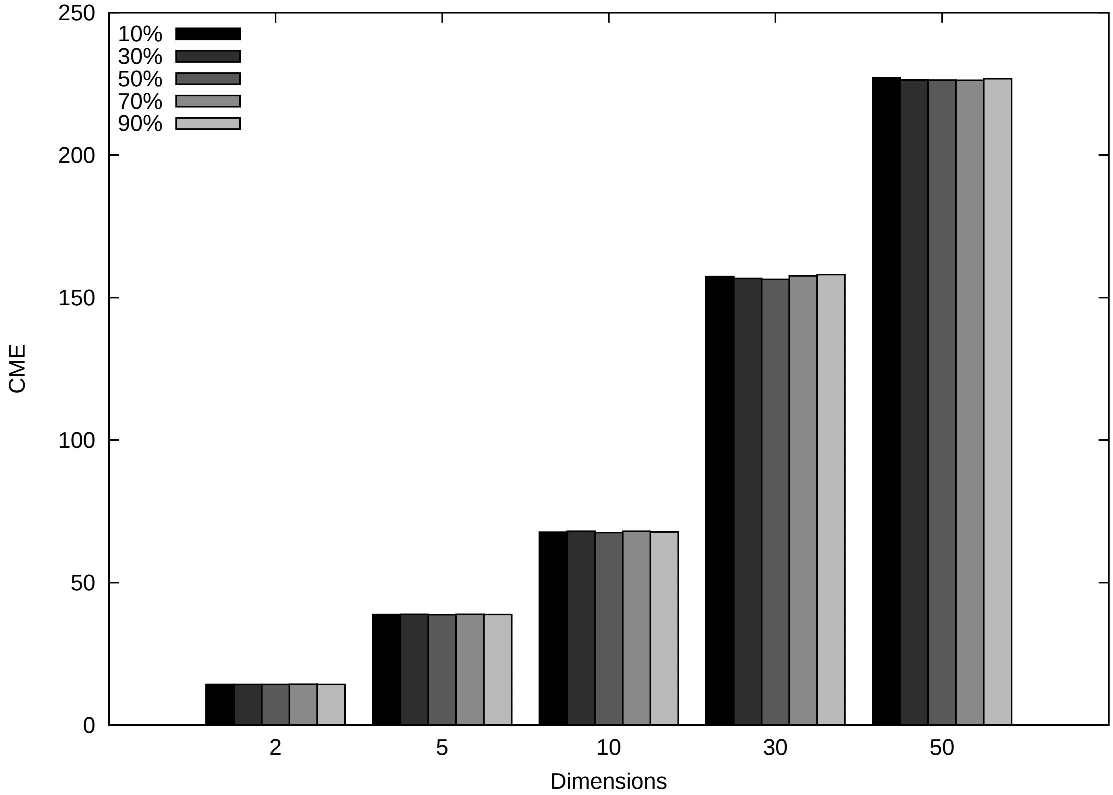
all times and fluctuated constantly due to the repulsion between the charged particles. It can be seen from the ITEL and ABEBC that a higher percentage of charged particles improved the recovery time of the algorithm at the expense of the exploitation capacity. For instance, when comparing APSO (50%) to APSO (90%) for the five-dimensional E-ABRUPT environments with one and five peaks, the CME and ABEBC of the APSO (50%) were lower but the ITEL of the APSO (50%) was significantly higher. This means the APSO (90%) reacted quickly to environmental changes and reached an error of 10 faster than the APSO (50%), but the APSO (50%) reached a lower error between two changes.

For E-CHAOS, the CME was higher than for E-PROGRESS and E-ABRUPT. The



Graph 8.10: APSO (50% charged) for E-ABRUPT

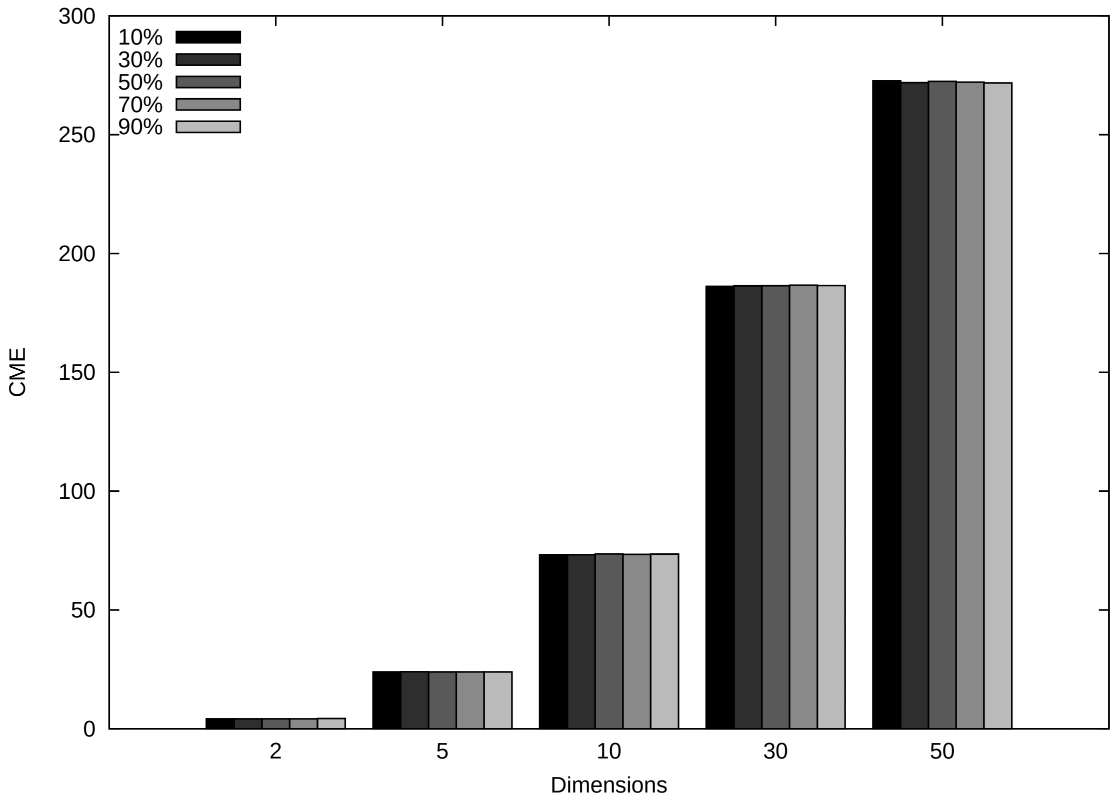
percentage of charged particles in the swarm had little influence on the performance level and the results were generally similar to those of the re-evaluating PSO. Graph 8.11 illustrates the lack of influence that the charged ratio had on the performance of the APSO for the E-CHAOS environments with five peaks. However, with a high percentage of charged particles, the ABEBC tended to be higher for the APSO than for the re-evaluating PSO for the environments of low dimensionality. This shows that in these environments, a high number of charged particles slowed down exploitation. As for the re-evaluating PSO, the CME was proportional to the number of peaks in the environment which indicates that the algorithm struggled to exploit the peaks and that a higher number of peaks increases the chance of a particle to randomly land on a good position. Because the frequent and severe changes in chaotically changing environments cause the diversity of the swarm to remain high, maintaining diversity through repulsion



Graph 8.11: APSO with the various ratio of charged particles for E-CHAOS with five peaks

is unnecessary and the charged ratio has little effect on the performance.

For E-PATTERN, the results were close to those of the re-evaluating PSO and the observations made for the re-evaluating PSO also apply to the APSO. From the results, the influence that the percentage of charged particles had on the performance level is unclear for the multimodal environments with more than two dimensions, as illustrated in graph 8.12 for the E-PATTERN environments with five peaks. However, on unimodal problems, the re-evaluating PSO performed better than the APSO and the CME increased proportionally to the charged ratio. Also, for the two-dimensional E-PATTERN with 15 peaks, the APSO (70%) obtained the lowest CME.



Graph 8.12: APSO with the various ratio of charged particles for E-PATTERN with five peaks

8.3.3 Summary of Strengths and Weaknesses

The APSO was able to track peaks and could detect the appearance of new optima better than the re-evaluating PSO. The results indicate that the APSO can outperform the re-evaluating PSO for progressively changing multimodal environments and for abruptly changing environments of low dimensionality. For static environments, the charged particles may slow down exploitation. For chaotically changing environments, the influence of the charged particles was minimal.

8.4 Quantum Swarm Optimisation

The QSO algorithm described in section 4.8.3 is evaluated in this section. Like the APSO, QSO uses diversity maintenance to overcome diversity loss, but instead of using inter-particles repulsion, the quantum particles of the QSO algorithm are uniformly distributed within a quantum radius. Quantum particles can be used to promote either exploitation or exploration depending on the size of the quantum radius. A small quantum radius keeps the quantum particles close to the *gbest* which means that the quantum particles immediately start exploiting the first optimum found. Figure 8.2 shows a QSO swarm with quantum radius of five and 50% of quantum particles converging after only two iterations in the E-STATIC environment with two dimensions and five peaks. On the other hand, a large quantum radius disperses the quantum particles evenly within a large area which promotes exploration. Figure 8.3 shows a QSO swarm with quantum radius 50 and 50% of quantum particles in a E-PROGRESS environment with five peaks after a 100 iterations. The figure illustrates that the swarm is still exploring a large area of the search space after 100 iterations. Two configurations of the QSO algorithm were therefore evaluated: one with a quantum radius of five and another one with a quantum radius of 50. The value five for the quantum radius was selected as it is small compared to the size of the domain of the search space. With quantum radius five, the radius of the hypersphere where the quantum particles are located is larger than the spatial severity for E-PROGRESS but smaller than the spatial severity for E-ABRUPT. This should allow the swarm to track a peak in the E-PROGRESS environments, but not necessarily in the E-ABRUPT environments. The value 50 was selected to allow the quantum particles to explore the majority of the search space, since the domain has a length of 100 which is equal to twice the quantum radius. For each configuration, QSO is evaluated with 10%, 30%, 50%, 70% and 90% of quantum particles in the swarm in order to evaluate the effect that the quantum ratio has on the performance.

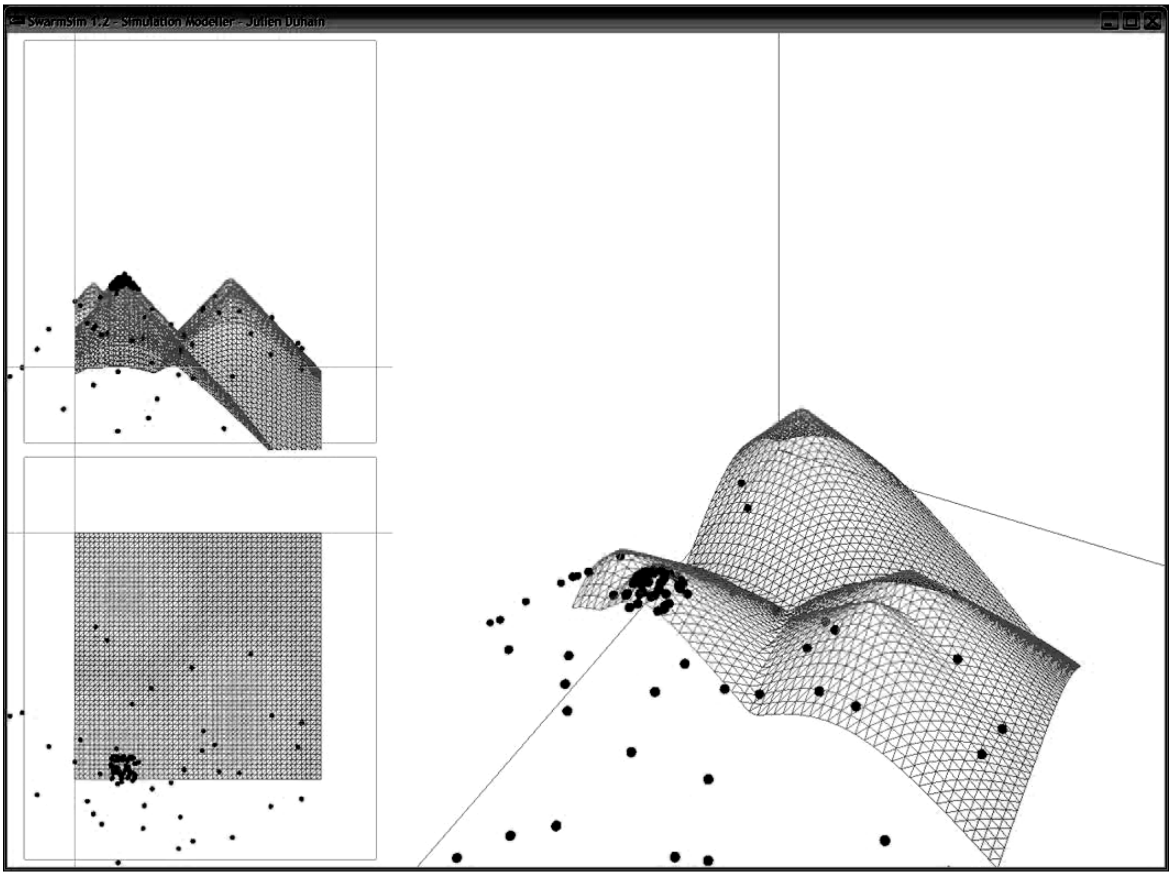


Figure 8.2: QSO (radius 5, 50% quantum) for E-STATIC

8.4.1 Results

The results in tables 8.11 to 8.20 were obtained by an implementation of algorithm 4.3 using the parameters from section 6.4. The ratio of quantum particles in the swarm and the size of the quantum radius are mentioned in the legend of the tables. The code for the QSO algorithm can be found in CIlib [74].

Table 8.11: Experimental results of QSO - quantum ratio 10%, quantum radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$1.69e{-2}\pm2.04e{-3}$	$0.25\pm1.83e{-2}$	$1.35\pm7.57e{-2}$	10.82 ± 0.50	25.84 ± 0.91
	5 p	$1.55e{-2}\pm2.20e{-3}$	3.30 ± 1.44	1.95 ± 1.06	12.15 ± 0.49	33.26 ± 2.36
	15 p	$2.94e{-2}\pm6.52e{-3}$	0.61 ± 0.44	7.92 ± 2.64	21.64 ± 2.32	36.02 ± 2.65
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$1.96e{-7}\pm1.36e{-8}$	$1.50e{-4}\pm1.04e{-5}$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$4.53e{-7}\pm3.15e{-8}$	$1.86e{-4}\pm1.29e{-5}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$2.84e{-15}\pm1.97e{-16}$	$0.28\pm1.97e{-2}$	$0.69\pm4.81e{-2}$
E-PROGRESS (CME)	1 p	$1.67\pm2.51e{-3}$	$2.13\pm1.92e{-2}$	$4.26\pm7.49e{-2}$	17.56 ± 0.53	37.48 ± 1.05
	5 p	4.39 ± 0.35	8.47 ± 1.38	13.96 ± 0.48	23.82 ± 1.05	43.65 ± 2.02
	15 p	6.59 ± 0.46	11.70 ± 0.81	18.72 ± 2.76	32.84 ± 1.16	41.31 ± 1.47
E-ABRUPT (CME)	1 p	$0.93\pm1.59e{-2}$	$2.62\pm5.17e{-2}$	5.37 ± 0.10	20.48 ± 0.41	39.83 ± 1.16
	5 p	4.88 ± 0.99	4.37 ± 0.23	12.38 ± 1.95	25.66 ± 1.95	57.20 ± 1.97
	15 p	7.09 ± 1.22	8.17 ± 1.47	10.65 ± 0.45	34.80 ± 3.31	47.77 ± 1.16
E-ABRUPT (ABEBC)	1 p	$1.78e{-9}\pm4.27e{-10}$	$1.89e{-5}\pm2.11e{-6}$	$6.05e{-3}\pm2.64e{-4}$	$1.34\pm4.10e{-2}$	5.27 ± 0.19
	5 p	3.87 ± 0.94	1.54 ± 0.27	6.49 ± 2.02	5.51 ± 1.88	20.35 ± 1.48
	15 p	6.50 ± 1.25	5.99 ± 1.40	5.81 ± 0.49	14.96 ± 3.52	13.72 ± 0.32
E-ABRUPT (ITEL)	1 p	$4.07\pm6.94e{-2}$	9.61 ± 0.30	19.57 ± 0.48	64.15 ± 1.03	114.98 ± 1.19
	5 p	59.90 ± 12.09	12.23 ± 0.47	87.30 ± 16.20	107.06 ± 9.61	184.00 ± 3.41
	15 p	48.53 ± 9.99	58.25 ± 15.88	84.83 ± 5.60	158.38 ± 17.04	165.13 ± 1.08
E-CHAOS (CME)	1 p	$20.56\pm4.35e{-2}$	39.73 ± 0.11	68.56 ± 0.23	158.15 ± 1.20	233.08 ± 2.42
	5 p	$13.02\pm7.85e{-2}$	36.79 ± 0.17	65.09 ± 0.67	147.22 ± 1.86	207.03 ± 2.20
	15 p	$9.69\pm4.39e{-2}$	31.16 ± 0.13	56.94 ± 0.35	130.71 ± 2.16	185.14 ± 3.79
E-CHAOS (ABEBC)	1 p	$1.21\pm1.43e{-2}$	16.58 ± 0.15	49.87 ± 0.28	148.03 ± 1.22	225.49 ± 2.43
	5 p	$1.94\pm5.15e{-2}$	19.42 ± 0.20	51.42 ± 0.67	138.63 ± 1.80	200.81 ± 2.23
	15 p	$1.67\pm4.34e{-2}$	17.01 ± 0.12	45.52 ± 0.38	123.75 ± 2.07	179.92 ± 3.69
E-CHAOS (ITEL)	1 p	$1.34\pm1.11e{-2}$	$4.77\pm1.56e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.29\pm1.96e{-2}$	$4.80\pm1.50e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.07\pm8.35e{-3}$	$4.75\pm1.89e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.16\pm3.95e{-3}$	$17.70\pm9.37e{-2}$	79.44 ± 0.56	217.15 ± 1.10	293.37 ± 1.45
	5 p	$4.14\pm1.74e{-2}$	21.54 ± 0.35	69.05 ± 0.59	186.90 ± 0.83	273.74 ± 1.41
	15 p	5.74 ± 0.29	18.70 ± 0.17	57.78 ± 0.28	179.53 ± 0.70	241.28 ± 1.16

Table 8.12: Experimental results of QSO - quantum ratio 30%, quantum radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.11e−2±1.46e−3	0.22±1.63e−2	1.18±9.05e−2	9.35±0.45	21.85±0.98
	5 p	1.16e−2±1.43e−3	4.06±1.37	8.00±4.26	11.77±1.79	31.29±2.95
	15 p	3.38e−2±1.62e−2	2.55±1.29	9.05±2.98	17.74±2.55	31.78±2.98
E-STATIC (Error)	1 p	0.00±0.00	0.00±0.00	0.00±0.00	6.56e−7±4.56e−8	2.58e−4±1.79e−5
	5 p	0.00±0.00	0.16±1.14e−2	0.00±0.00	7.57e−2±5.26e−3	0.90±6.24e−2
	15 p	0.00±0.00	0.00±0.00	0.47±3.27e−2	2.12e−7±1.47e−8	0.74±5.12e−2
E-PROGRESS (CME)	1 p	1.67±2.07e−3	2.13±1.74e−2	3.96±7.75e−2	15.40±0.48	32.47±1.41
	5 p	4.43±0.35	11.45±2.29	15.52±1.57	20.96±0.84	38.43±1.97
	15 p	6.96±0.66	13.17±0.91	22.19±2.64	30.49±1.90	41.71±2.71
E-ABRUPT (CME)	1 p	0.81±1.08e−2	2.32±3.77e−2	4.70±6.93e−2	17.57±0.33	33.12±0.77
	5 p	6.71±0.89	5.11±0.74	12.89±2.72	22.59±2.27	48.18±1.07
	15 p	9.19±0.98	9.57±1.87	13.08±2.23	30.56±3.25	40.78±1.46
E-ABRUPT (ABEBC)	1 p	2.96e−9±6.44e−10	2.43e−5±2.97e−6	5.89e−3±3.44e−4	1.05±2.94e−2	3.60±6.17e−2
	5 p	5.68±0.96	2.53±0.75	8.19±2.75	5.75±2.22	16.82±0.73
	15 p	8.69±1.04	7.68±1.86	8.85±2.18	14.60±3.38	12.25±1.19
E-ABRUPT (ITEL)	1 p	3.68±5.76e−2	8.72±0.19	16.73±0.26	48.56±0.62	80.82±1.11
	5 p	83.67±10.86	22.27±11.96	89.44±21.03	95.34±13.32	165.60±3.17
	15 p	69.33±8.67	75.67±20.63	107.77±16.44	150.43±18.42	148.01±5.36
E-CHAOS (CME)	1 p	20.55±6.13e−2	39.91±8.05e−2	66.85±0.38	146.86±1.33	216.79±2.09
	5 p	13.01±9.08e−2	37.13±0.26	67.08±1.15	140.62±1.80	194.01±2.13
	15 p	9.66±7.33e−2	31.46±0.19	56.90±0.36	127.79±2.49	174.54±2.65
E-CHAOS (ABEBC)	1 p	0.78±1.17e−2	14.83±0.12	46.45±0.39	135.49±1.36	208.26±2.12
	5 p	2.08±6.42e−2	18.23±0.23	51.06±0.99	130.82±1.77	186.91±2.07
	15 p	1.69±5.71e−2	16.11±0.18	44.07±0.34	119.48±2.34	168.47±2.57
E-CHAOS (ITEL)	1 p	1.41±1.30e−2	4.64±1.59e−2	5.00±3.41e−4	5.00±0.00	5.00±0.00
	5 p	1.39±1.98e−2	4.72±1.93e−2	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	1.10±1.40e−2	4.65±2.00e−2	5.00±0.00	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	3.17±2.60e−3	16.99±0.11	75.74±0.58	224.24±1.34	302.85±1.68
	5 p	4.15±3.42e−3	21.60±0.53	67.48±0.66	191.50±1.19	280.68±1.66
	15 p	6.38±0.32	19.30±0.28	56.64±0.40	183.06±1.10	244.71±1.27

Table 8.13: Experimental results of QSO - quantum ratio 50%, quantum radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.12e−2±1.75e−3	0.21±1.96e−2	1.16±8.80e−2	9.04±0.44	22.88±0.86
	5 p	1.47e−2±8.85e−3	7.77±2.09	10.50±3.79	10.64±1.08	32.59±3.35
	15 p	5.27e−2±4.89e−2	3.48±1.38	8.49±3.07	22.30±3.36	30.83±1.89
E-STATIC (Error)	1 p	0.00±0.00	0.00±0.00	2.37e−16±1.64e−17	2.62e−6±1.82e−7	1.19e−3±8.24e−5
	5 p	0.00±0.00	0.16±1.14e−2	4.74e−16±3.29e−17	7.57e−2±5.26e−3	0.40±2.75e−2
	15 p	0.00±0.00	0.32±2.23e−2	0.47±3.27e−2	0.28±1.97e−2	0.39±2.68e−2
E-PROGRESS (CME)	1 p	1.67±2.20e−3	2.14±1.66e−2	3.92±6.95e−2	14.65±0.47	30.68±1.03
	5 p	4.53±0.37	10.71±1.85	14.66±0.98	21.00±1.12	37.10±1.35
	15 p	7.12±0.66	14.04±0.89	19.93±3.13	28.82±1.92	42.32±1.65
E-ABRUPT (CME)	1 p	0.80±1.15e−2	2.28±4.06e−2	4.55±6.45e−2	17.01±0.39	32.46±0.73
	5 p	7.85±0.97	5.29±0.78	15.05±3.19	21.83±1.91	48.48±1.74
	15 p	10.15±0.81	10.20±1.91	11.98±1.77	32.53±3.17	41.43±1.00
E-ABRUPT (ABEBC)	1 p	6.96e−9±1.78e−9	4.70e−5±6.22e−6	1.06e−2±1.15e−3	1.13±3.72e−2	3.44±6.66e−2
	5 p	6.92±1.00	2.81±0.78	10.36±3.23	5.23±1.87	17.79±1.54
	15 p	9.73±0.87	8.33±1.91	7.70±1.71	16.72±3.27	12.61±0.65
E-ABRUPT (ITEL)	1 p	3.68±8.47e−2	8.75±0.18	16.13±0.21	44.93±0.49	72.63±0.90
	5 p	97.57±12.47	21.67±13.14	104.53±23.90	89.11±11.33	165.64±5.11
	15 p	75.15±5.37	83.73±19.25	97.55±16.34	159.24±16.84	149.93±7.63
E-CHAOS (CME)	1 p	21.21±6.65e−2	41.84±0.14	69.01±0.42	148.59±1.21	218.23±1.78
	5 p	13.61±0.12	38.73±0.27	68.18±1.27	143.03±1.75	192.17±2.11
	15 p	9.97±6.66e−2	33.02±0.19	58.91±0.58	128.53±2.34	173.75±3.47
E-CHAOS (ABEBC)	1 p	0.65±1.56e−2	16.08±0.15	48.45±0.43	137.21±1.24	209.71±1.78
	5 p	2.48±9.70e−2	19.28±0.27	52.01±1.03	133.15±1.69	185.12±2.07
	15 p	2.01±6.73e−2	17.22±0.19	45.78±0.52	120.20±2.20	167.63±3.37
E-CHAOS (ITEL)	1 p	1.57±1.73e−2	4.68±1.52e−2	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	1.54±2.38e−2	4.74±1.47e−2	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	1.18±1.88e−2	4.68±2.34e−2	5.00±0.00	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	3.17±4.06e−3	18.13±0.12	79.97±0.64	229.32±1.32	309.89±2.06
	5 p	4.14±2.63e−3	23.03±0.39	70.86±0.74	194.26±1.44	286.42±1.92
	15 p	7.72±0.31	20.34±0.30	58.24±0.65	186.99±0.87	251.58±1.39

Table 8.14: Experimental results of QSO - quantum ratio 70%, quantum radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$8.53e{-3}\pm1.51e{-3}$	$0.21\pm2.31e{-2}$	$1.32\pm8.73e{-2}$	10.15 ± 0.39	24.15 ± 0.65
	5 p	$1.19e{-2}\pm2.20e{-3}$	6.52 ± 1.76	8.86 ± 4.09	12.44 ± 1.35	31.61 ± 3.11
	15 p	0.54 ± 0.49	3.45 ± 1.40	8.20 ± 2.45	21.85 ± 3.18	34.30 ± 2.66
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	$1.09e{-14}\pm7.56e{-16}$	$2.13e{-5}\pm1.48e{-6}$	$1.91e{-3}\pm1.33e{-4}$
	5 p	0.00 ± 0.00	$0.31\pm2.12e{-2}$	$1.30e{-13}\pm9.01e{-15}$	$0.10\pm7.29e{-3}$	$0.34\pm2.37e{-2}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$3.72e{-3}\pm2.58e{-4}$	$0.84\pm5.85e{-2}$	$0.44\pm3.02e{-2}$
E-PROGRESS (CME)	1 p	$1.67\pm2.45e{-3}$	$2.17\pm1.88e{-2}$	4.05 ± 0.11	15.50 ± 0.62	33.09 ± 0.93
	5 p	4.63 ± 0.35	12.93 ± 2.43	17.34 ± 2.18	21.58 ± 1.20	39.33 ± 1.58
	15 p	6.57 ± 0.73	14.82 ± 1.62	24.22 ± 2.02	29.67 ± 1.34	44.54 ± 2.84
E-ABRUPT (CME)	1 p	$0.79\pm1.09e{-2}$	$2.36\pm2.90e{-2}$	4.95 ± 0.12	17.19 ± 0.45	33.44 ± 0.77
	5 p	8.77 ± 0.60	5.89 ± 1.19	15.34 ± 3.15	27.07 ± 3.54	52.02 ± 1.85
	15 p	10.82 ± 0.98	11.10 ± 2.46	13.53 ± 2.13	32.67 ± 2.65	43.54 ± 1.62
E-ABRUPT (ABEBC)	1 p	$2.45e{-8}\pm8.34e{-9}$	$3.11e{-4}\pm1.45e{-4}$	$4.71e{-2}\pm1.54e{-2}$	$1.46\pm6.91e{-2}$	$3.73\pm5.18e{-2}$
	5 p	7.87 ± 0.64	3.44 ± 1.20	10.53 ± 3.21	10.52 ± 3.61	19.35 ± 1.72
	15 p	10.47 ± 1.05	9.25 ± 2.51	9.11 ± 2.07	16.90 ± 2.76	13.56 ± 1.59
E-ABRUPT (ITEL)	1 p	$3.68\pm7.24e{-2}$	9.21 ± 0.15	17.09 ± 0.32	44.25 ± 0.48	70.89 ± 0.70
	5 p	103.53 ± 9.03	24.23 ± 14.86	108.95 ± 24.49	115.90 ± 20.98	172.85 ± 6.52
	15 p	75.17 ± 7.53	96.67 ± 23.70	105.20 ± 18.21	171.13 ± 15.30	150.17 ± 9.45
E-CHAOS (CME)	1 p	22.89 ± 0.12	46.11 ± 0.25	74.72 ± 0.56	157.86 ± 1.27	225.04 ± 1.97
	5 p	$14.90\pm9.67e{-2}$	42.34 ± 0.24	73.70 ± 1.35	148.67 ± 1.90	201.02 ± 2.61
	15 p	$10.77\pm7.65e{-2}$	35.66 ± 0.25	62.77 ± 0.67	134.79 ± 2.58	184.29 ± 3.32
E-CHAOS (ABEBC)	1 p	$0.74\pm2.72e{-2}$	20.12 ± 0.23	55.05 ± 0.61	147.22 ± 1.26	216.89 ± 1.97
	5 p	$3.43\pm9.88e{-2}$	22.65 ± 0.28	58.04 ± 1.18	139.35 ± 1.87	194.37 ± 2.57
	15 p	2.69 ± 0.10	19.69 ± 0.24	49.99 ± 0.59	126.86 ± 2.42	178.51 ± 3.26
E-CHAOS (ITEL)	1 p	$1.86\pm2.43e{-2}$	$4.79\pm1.26e{-2}$	$5.00\pm3.41e{-4}$	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.86\pm2.44e{-2}$	$4.84\pm9.52e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.39\pm2.67e{-2}$	$4.77\pm1.76e{-2}$	$5.00\pm5.70e{-4}$	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.17\pm4.04e{-3}$	21.63 ± 0.22	90.65 ± 0.84	235.15 ± 1.44	315.87 ± 1.74
	5 p	$4.15\pm4.06e{-3}$	26.28 ± 0.50	78.63 ± 0.69	199.45 ± 1.36	290.63 ± 1.43
	15 p	8.44 ± 0.30	21.19 ± 0.29	62.81 ± 0.73	191.12 ± 0.85	255.07 ± 1.61

Table 8.15: Experimental results of QSO - quantum ratio 90%, quantum radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$1.10e{-2}\pm2.00e{-3}$	$0.38\pm5.18e{-2}$	2.24±0.19	14.42±0.87	31.87±0.99
	5 p	$1.09e{-2}\pm3.74e{-3}$	7.52±2.52	15.27±4.38	16.65±1.72	38.24±2.85
	15 p	1.30±0.95	3.09±1.61	12.77±3.58	22.35±2.59	40.15±1.98
E-STATIC (Error)	1 p	0.00±0.00	$1.23e{-2}\pm8.57e{-4}$	$7.79e{-11}\pm5.41e{-12}$	$9.42e{-2}\pm6.54e{-3}$	$0.11\pm7.92e{-3}$
	5 p	$4.74e{-16}\pm3.29e{-17}$	$9.01e{-3}\pm6.25e{-4}$	$0.52\pm3.60e{-2}$	$0.17\pm1.15e{-2}$	$0.11\pm7.87e{-3}$
	15 p	0.00±0.00	$0.21\pm1.48e{-2}$	$0.20\pm1.41e{-2}$	$0.89\pm6.17e{-2}$	$0.79\pm5.51e{-2}$
E-PROGRESS (CME)	1 p	$1.68\pm2.93e{-3}$	$2.22\pm3.46e{-2}$	4.41±0.19	18.03±0.66	39.54±1.30
	5 p	5.29±0.24	14.43±2.46	15.71±1.54	22.84±1.05	46.21±2.36
	15 p	6.46±0.75	13.83±1.39	20.32±2.77	31.43±2.08	49.37±2.93
E-ABRUPT (CME)	1 p	$0.83\pm1.21e{-2}$	$2.72\pm5.90e{-2}$	5.92±0.15	21.32±0.65	39.77±1.30
	5 p	9.26±0.31	6.82±1.58	14.47±2.72	27.56±2.73	56.27±1.88
	15 p	10.86±0.87	13.39±2.37	15.16±2.42	32.22±2.82	49.65±2.23
E-ABRUPT (ABEBC)	1 p	$4.67e{-4}\pm5.30e{-4}$	$0.13\pm4.80e{-2}$	0.71±0.13	$2.61\pm9.71e{-2}$	$4.61\pm7.15e{-2}$
	5 p	8.33±0.30	4.12±1.56	9.43±2.80	9.38±2.91	18.66±1.49
	15 p	10.49±0.99	11.47±2.44	10.44±2.34	14.37±2.99	14.47±1.85
E-ABRUPT (ITEL)	1 p	$3.95\pm9.14e{-2}$	10.40±0.18	18.94±0.34	48.98±0.64	75.16±0.89
	5 p	106.13±7.33	30.35±17.44	104.60±20.39	107.03±17.73	169.21±4.44
	15 p	76.13±7.15	117.51±22.68	114.43±19.71	146.69±18.85	147.10±7.19
E-CHAOS (CME)	1 p	31.09±0.27	60.13±0.50	94.47±0.97	183.93±1.15	259.40±1.72
	5 p	20.16±0.32	53.13±0.42	88.24±1.63	169.78±2.13	223.42±2.52
	15 p	14.21±0.19	44.02±0.46	74.29±0.66	150.59±3.24	203.25±4.81
E-CHAOS (ABEBC)	1 p	4.27±0.19	37.01±0.55	78.54±0.99	175.35±1.16	252.90±1.71
	5 p	7.33±0.32	35.09±0.44	75.66±1.45	162.39±2.11	218.03±2.50
	15 p	5.55±0.18	29.07±0.45	63.77±0.61	144.51±3.12	198.63±4.71
E-CHAOS (ITEL)	1 p	$2.95\pm2.80e{-2}$	$4.93\pm8.68e{-3}$	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	$2.88\pm4.82e{-2}$	$4.95\pm7.19e{-3}$	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	$2.31\pm4.54e{-2}$	$4.93\pm7.49e{-3}$	5.00±0.00	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	$3.17\pm2.87e{-3}$	36.71±0.52	120.02±1.07	240.93±1.58	319.54±2.03
	5 p	$4.15\pm5.58e{-3}$	36.73±0.72	99.22±0.62	206.95±1.47	296.11±1.61
	15 p	9.13±0.45	26.55±0.49	75.52±0.64	199.62±1.43	259.90±1.73

Table 8.16: Experimental results of QSO - quantum ratio 10%, quantum radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$2.69e{-2}\pm3.05e{-3}$	$0.39\pm2.32e{-2}$	$1.73\pm6.46e{-2}$	10.60 ± 0.37	24.75 ± 0.52
	5 p	$3.01e{-2}\pm5.11e{-3}$	1.48 ± 0.94	2.38 ± 1.03	12.40 ± 1.16	32.48 ± 2.39
	15 p	$4.71e{-2}\pm7.73e{-3}$	0.64 ± 0.42	6.49 ± 2.43	19.26 ± 2.81	31.81 ± 2.54
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$2.12e{-6}\pm1.47e{-7}$	$5.20e{-4}\pm3.61e{-5}$
	5 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$7.57e{-2}\pm5.26e{-3}$	$0.34\pm2.35e{-2}$
	15 p	0.00 ± 0.00	$0.20\pm1.41e{-2}$	$0.47\pm3.27e{-2}$	$1.82e{-6}\pm1.26e{-7}$	$0.57\pm3.94e{-2}$
E-PROGRESS (CME)	1 p	$1.68\pm3.15e{-3}$	$2.27\pm2.11e{-2}$	$4.68\pm7.11e{-2}$	17.43 ± 0.40	33.18 ± 0.60
	5 p	$2.93\pm2.03e{-2}$	6.89 ± 0.12	13.54 ± 0.76	22.95 ± 0.82	41.00 ± 0.86
	15 p	3.05 ± 0.19	9.07 ± 0.33	18.47 ± 2.56	33.54 ± 1.45	41.92 ± 2.26
E-ABRUPT (CME)	1 p	$0.41\pm1.35e{-2}$	$2.09\pm5.03e{-2}$	6.02 ± 0.10	26.45 ± 0.35	47.29 ± 0.53
	5 p	$0.58\pm2.04e{-2}$	2.71 ± 0.18	14.03 ± 2.56	34.48 ± 3.25	62.59 ± 0.80
	15 p	$0.59\pm5.39e{-2}$	5.24 ± 1.08	12.24 ± 0.76	41.75 ± 3.08	55.87 ± 1.53
E-ABRUPT (ABEBC)	1 p	$2.54e{-9}\pm6.37e{-10}$	$2.60e{-5}\pm2.25e{-6}$	$1.22e{-2}\pm8.23e{-4}$	$3.85\pm9.50e{-2}$	15.56 ± 0.38
	5 p	$1.45e{-3}\pm1.40e{-3}$	0.16 ± 0.17	7.54 ± 2.64	11.30 ± 3.20	28.55 ± 0.57
	15 p	$0.11\pm5.65e{-2}$	2.88 ± 1.06	6.23 ± 0.73	19.54 ± 3.22	24.53 ± 1.19
E-ABRUPT (ITEL)	1 p	1.59 ± 0.12	10.73 ± 0.47	33.58 ± 0.82	131.55 ± 1.16	199.73 ± 0.28
	5 p	2.44 ± 0.24	14.84 ± 0.95	101.37 ± 16.87	164.63 ± 7.49	200.00 ± 0.00
	15 p	1.61 ± 0.13	33.97 ± 11.17	106.78 ± 7.49	187.85 ± 6.75	199.91 ± 0.19
E-CHAOS (CME)	1 p	$21.44\pm4.18e{-2}$	40.52 ± 0.15	66.68 ± 0.23	140.80 ± 0.75	197.65 ± 0.99
	5 p	$14.14\pm6.17e{-2}$	37.12 ± 0.15	62.08 ± 0.57	135.16 ± 1.12	178.61 ± 1.64
	15 p	$10.68\pm4.44e{-2}$	31.68 ± 0.11	54.39 ± 0.26	116.39 ± 1.89	159.71 ± 1.63
E-CHAOS (ABEBC)	1 p	$2.67\pm3.86e{-2}$	20.67 ± 0.17	49.29 ± 0.24	128.67 ± 0.74	187.69 ± 0.99
	5 p	$3.40\pm5.09e{-2}$	21.93 ± 0.17	48.77 ± 0.52	124.67 ± 1.06	170.46 ± 1.60
	15 p	$2.81\pm3.22e{-2}$	19.37 ± 0.11	43.12 ± 0.27	107.99 ± 1.76	152.68 ± 1.54
E-CHAOS (ITEL)	1 p	$1.35\pm1.52e{-2}$	$4.94\pm9.65e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.36\pm2.03e{-2}$	$4.93\pm1.11e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.10\pm9.87e{-3}$	$4.90\pm1.66e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.17\pm4.05e{-3}$	$20.01\pm9.18e{-2}$	70.22 ± 0.39	205.57 ± 0.89	283.13 ± 1.31
	5 p	$4.15\pm3.27e{-3}$	22.83 ± 0.23	61.96 ± 0.38	175.81 ± 0.94	262.30 ± 0.98
	15 p	$4.96\pm2.75e{-2}$	20.03 ± 0.12	53.66 ± 0.26	169.16 ± 0.75	230.65 ± 0.92

Table 8.17: Experimental results of QSO - quantum ratio 30%, quantum radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$2.54e{-2}\pm3.37e{-3}$	$0.39\pm2.15e{-2}$	$1.45\pm6.02e{-2}$	8.70 ± 0.21	19.52 ± 0.43
	5 p	$3.18e{-2}\pm4.44e{-3}$	3.88 ± 1.53	10.14 ± 4.21	12.68 ± 2.08	28.00 ± 2.40
	15 p	$6.08e{-2}\pm1.16e{-2}$	1.93 ± 1.05	8.25 ± 2.97	17.60 ± 2.78	26.45 ± 2.47
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	$4.79e{-6}\pm3.33e{-7}$	$1.21e{-3}\pm8.42e{-5}$
	5 p	0.00 ± 0.00	$0.31\pm2.12e{-2}$	$0.61\pm4.26e{-2}$	$7.57e{-2}\pm5.26e{-3}$	$0.34\pm2.36e{-2}$
	15 p	0.00 ± 0.00	$1.56e{-2}\pm1.09e{-3}$	$3.72e{-3}\pm2.58e{-4}$	$0.28\pm1.97e{-2}$	$8.49e{-4}\pm5.90e{-5}$
E-PROGRESS (CME)	1 p	$1.68\pm3.29e{-3}$	$2.34\pm2.05e{-2}$	$4.60\pm6.37e{-2}$	15.44 ± 0.20	28.51 ± 0.41
	5 p	$2.88\pm7.53e{-3}$	8.98 ± 1.88	17.68 ± 1.88	21.48 ± 0.72	36.50 ± 1.35
	15 p	$2.74\pm3.46e{-2}$	9.00 ± 0.49	20.75 ± 2.89	29.48 ± 2.71	39.42 ± 2.24
E-ABRUPT (CME)	1 p	$0.37\pm6.64e{-3}$	$1.86\pm4.40e{-2}$	$5.50\pm8.53e{-2}$	24.83 ± 0.30	43.94 ± 0.42
	5 p	$0.56\pm2.06e{-2}$	2.92 ± 0.33	16.57 ± 3.27	32.55 ± 3.04	60.70 ± 1.35
	15 p	$0.61\pm6.05e{-2}$	5.76 ± 1.25	12.71 ± 1.57	41.26 ± 3.01	53.13 ± 1.46
E-ABRUPT (ABEBC)	1 p	$4.14e{-9}\pm1.39e{-9}$	$4.45e{-5}\pm5.29e{-6}$	$1.74e{-2}\pm1.08e{-3}$	$4.27\pm9.40e{-2}$	15.77 ± 0.31
	5 p	$5.78e{-3}\pm2.06e{-3}$	0.57 ± 0.30	10.87 ± 3.28	11.22 ± 3.07	30.78 ± 1.49
	15 p	$0.15\pm5.53e{-2}$	3.56 ± 1.24	7.42 ± 1.53	20.98 ± 3.16	25.35 ± 1.51
E-ABRUPT (ITEL)	1 p	$1.22\pm7.18e{-2}$	9.49 ± 0.55	34.19 ± 0.80	137.08 ± 1.63	200.00 ± 0.00
	5 p	1.75 ± 0.15	16.01 ± 2.88	121.25 ± 21.34	166.67 ± 6.45	200.00 ± 0.00
	15 p	1.38 ± 0.12	41.06 ± 14.88	106.39 ± 12.05	190.03 ± 5.10	200.00 ± 0.00
E-CHAOS (CME)	1 p	$21.21\pm5.42e{-2}$	38.48 ± 0.13	59.86 ± 0.19	117.03 ± 0.58	161.07 ± 0.72
	5 p	$14.10\pm7.51e{-2}$	35.37 ± 0.13	58.30 ± 0.80	114.97 ± 1.44	149.62 ± 1.68
	15 p	$10.62\pm5.38e{-2}$	30.34 ± 0.14	50.92 ± 0.28	107.03 ± 2.70	137.90 ± 4.74
E-CHAOS (ABEBC)	1 p	$2.74\pm3.60e{-2}$	18.97 ± 0.13	41.07 ± 0.21	102.14 ± 0.55	148.58 ± 0.74
	5 p	$3.40\pm5.02e{-2}$	20.26 ± 0.14	43.66 ± 0.73	102.01 ± 1.31	139.35 ± 1.58
	15 p	$2.79\pm3.35e{-2}$	17.88 ± 0.15	38.57 ± 0.28	95.55 ± 2.35	128.70 ± 4.39
E-CHAOS (ITEL)	1 p	$1.25\pm1.62e{-2}$	$4.91\pm1.31e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.35\pm1.41e{-2}$	$4.87\pm1.27e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.09\pm8.96e{-3}$	$4.83\pm1.55e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.19\pm4.30e{-3}$	$19.10\pm8.01e{-2}$	57.66 ± 0.29	199.91 ± 0.85	282.62 ± 1.34
	5 p	$4.17\pm4.85e{-3}$	22.22 ± 0.35	51.97 ± 0.60	167.81 ± 0.80	258.90 ± 1.20
	15 p	$4.80\pm2.79e{-2}$	19.93 ± 0.15	47.93 ± 0.28	163.96 ± 0.65	225.49 ± 1.09

Table 8.18: Experimental results of QSO - quantum ratio 50%, quantum radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	2.81e−2±2.82e−3	0.37±2.49e−2	1.43±3.92e−2	8.46±0.20	19.06±0.48
	5 p	3.03e−2±2.65e−3	5.15±1.48	10.07±4.04	11.38±1.74	25.23±2.54
	15 p	7.32e−2±2.24e−2	1.47±0.79	11.75±3.15	19.43±3.34	26.76±2.58
E-STATIC (Error)	1 p	0.00±0.00	0.00±0.00	0.00±0.00	7.90e−6±5.49e−7	4.28e−3±2.97e−4
	5 p	0.00±0.00	0.00±0.00	4.74e−16±3.29e−17	0.54±3.78e−2	0.34±2.37e−2
	15 p	0.00±0.00	0.00±0.00	0.47±3.27e−2	5.51e−6±3.82e−7	0.39±2.70e−2
E-PROGRESS (CME)	1 p	1.69±4.02e−3	2.48±2.70e−2	4.89±6.20e−2	15.84±0.24	28.27±0.64
	5 p	2.87±9.06e−3	11.11±2.44	17.69±1.94	22.69±0.90	35.99±0.80
	15 p	2.65±1.84e−2	9.88±0.93	20.61±2.35	31.82±2.11	40.82±2.34
E-ABRUPT (CME)	1 p	0.37±8.81e−3	1.89±5.23e−2	5.66±9.73e−2	26.02±0.36	44.85±0.54
	5 p	0.56±1.65e−2	3.85±1.09	18.69±3.50	31.67±2.12	62.78±1.81
	15 p	0.62±4.45e−2	5.62±1.10	13.67±1.93	42.88±3.07	54.36±1.50
E-ABRUPT (ABEBC)	1 p	7.46e−9±1.69e−9	9.91e−5±1.11e−5	3.30e−2±2.62e−3	5.73±0.25	18.49±0.61
	5 p	7.23e−3±1.94e−3	1.38±1.10	12.82±3.52	10.71±2.11	34.93±1.84
	15 p	0.18±4.99e−2	3.48±1.05	8.22±1.89	22.99±3.11	28.51±1.52
E-ABRUPT (ITEL)	1 p	1.11±6.42e−2	9.83±0.66	36.20±0.98	153.13±3.10	200.00±0.00
	5 p	1.61±0.12	31.21±16.20	134.85±21.71	175.43±3.85	200.00±0.00
	15 p	1.21±8.89e−2	39.31±13.44	118.89±14.42	193.63±3.61	200.00±0.00
E-CHAOS (CME)	1 p	21.08±3.61e−2	37.19±0.11	56.58±0.14	108.57±0.42	147.91±0.60
	5 p	14.08±6.70e−2	34.63±0.18	57.88±1.01	107.41±1.51	139.17±1.51
	15 p	10.68±5.59e−2	30.05±0.15	49.98±0.43	99.81±2.31	127.36±4.57
E-CHAOS (ABEBC)	1 p	2.80±3.67e−2	17.84±0.13	37.54±0.17	92.28±0.37	134.28±0.58
	5 p	3.44±5.65e−2	19.42±0.18	42.48±0.89	93.26±1.33	127.92±1.41
	15 p	2.87±3.68e−2	17.55±0.17	37.31±0.34	87.57±1.91	117.35±4.16
E-CHAOS (ITEL)	1 p	1.20±1.18e−2	4.90±1.28e−2	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	1.34±1.67e−2	4.83±1.66e−2	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	1.09±1.09e−2	4.78±1.78e−2	5.00±3.41e−4	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	3.22±7.31e−3	18.71±8.50e−2	52.16±0.31	197.99±0.90	283.96±1.06
	5 p	4.20±6.87e−3	22.33±0.48	46.98±0.64	164.08±0.77	259.83±1.07
	15 p	4.78±3.06e−2	20.06±0.27	45.72±0.34	162.96±0.66	224.63±0.99

Table 8.19: Experimental results of QSO - quantum ratio 70%, quantum radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	3.51e−2±3.91e−3	0.40±2.47e−2	1.57±8.29e−2	9.60±0.57	24.74±1.58
	5 p	3.56e−2±6.86e−3	5.64±1.95	12.40±4.42	12.63±1.62	36.73±4.18
	15 p	7.52e−2±2.35e−2	2.14±1.02	8.01±2.90	22.25±3.23	31.57±3.22
E-STATIC (Error)	1 p	0.00±0.00	0.00±0.00	1.47e−10±1.02e−11	5.53e−4±3.84e−5	0.32±2.22e−2
	5 p	0.00±0.00	0.16±1.14e−2	0.61±4.26e−2	7.58e−2±5.26e−3	1.08e−2±7.50e−4
	15 p	0.00±0.00	1.62e−7±1.12e−8	0.47±3.27e−2	0.86±5.95e−2	3.56e−2±2.47e−3
E-PROGRESS (CME)	1 p	1.70±3.69e−3	3.01±7.45e−2	6.16±0.26	18.82±0.86	33.11±1.17
	5 p	2.88±7.44e−3	15.45±2.86	20.79±2.25	28.71±1.75	43.20±1.87
	15 p	2.69±4.25e−2	11.10±0.90	26.60±2.88	36.45±3.24	43.85±2.18
E-ABRUPT (CME)	1 p	0.38±7.68e−3	2.06±5.33e−2	6.37±0.20	29.06±0.48	48.40±0.52
	5 p	0.58±4.19e−2	3.93±0.68	15.30±2.70	37.31±3.43	64.48±2.26
	15 p	0.70±4.49e−2	8.57±1.42	13.32±1.57	45.52±3.08	57.23±1.73
E-ABRUPT (ABEBC)	1 p	5.02e−8±1.79e−8	9.30e−4±3.59e−4	0.18±7.95e−2	10.53±0.80	25.05±0.88
	5 p	8.48e−3±1.66e−3	1.41±0.69	8.80±2.75	17.79±3.45	40.32±2.52
	15 p	0.23±4.93e−2	6.21±1.39	7.22±1.49	27.26±3.20	34.03±1.94
E-ABRUPT (ITEL)	1 p	1.06±4.00e−2	10.51±0.80	42.64±1.51	186.85±4.40	200.00±0.00
	5 p	1.77±0.60	25.87±10.88	115.60±16.21	192.99±2.99	200.00±0.00
	15 p	1.35±0.13	69.03±16.15	111.06±12.03	198.27±1.53	200.00±0.00
E-CHAOS (CME)	1 p	21.05±3.75e−2	36.37±8.38e−2	54.63±0.19	102.97±0.32	140.89±0.53
	5 p	14.25±8.14e−2	34.58±0.17	56.90±1.16	102.99±1.27	133.74±1.48
	15 p	10.71±6.60e−2	30.05±0.20	50.18±0.53	99.13±3.55	123.61±5.42
E-CHAOS (ABEBC)	1 p	2.90±3.84e−2	17.13±9.16e−2	35.37±0.17	86.28±0.27	126.56±0.51
	5 p	3.63±4.83e−2	19.31±0.19	41.19±0.91	88.17±1.12	121.82±1.35
	15 p	3.03±5.34e−2	17.57±0.20	37.21±0.47	85.83±2.97	112.96±4.83
E-CHAOS (ITEL)	1 p	1.17±1.18e−2	4.87±1.33e−2	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	1.39±1.72e−2	4.81±2.13e−2	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	1.11±1.36e−2	4.73±1.77e−2	5.00±0.00	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	3.29±8.91e−3	18.40±6.98e−2	49.12±0.23	194.97±0.76	283.96±0.99
	5 p	4.26±1.06e−2	22.18±0.64	45.62±0.78	161.29±0.80	260.31±1.05
	15 p	4.91±3.41e−2	19.84±0.23	44.66±0.53	162.47±0.71	225.17±1.08

Table 8.20: Experimental results of QSO - quantum ratio 90%, quantum radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$5.36e{-}2{\pm}6.43e{-}3$	2.00±0.52	8.19±1.32	27.14±2.30	41.79±1.91
	5 p	$5.82e{-}2{\pm}1.32e{-}2$	5.67±1.49	16.34±4.71	28.51±2.75	50.10±4.12
	15 p	0.57±0.63	5.83±1.82	14.64±3.39	38.57±3.57	50.25±2.42
E-STATIC (Error)	1 p	0.00±0.00	$2.89e{-}5{\pm}2.01e{-}6$	$0.33{\pm}2.26e{-}2$	$0.90{\pm}6.22e{-}2$	$0.90{\pm}6.22e{-}2$
	5 p	$2.77e{-}12{\pm}1.92e{-}13$	$4.32e{-}3{\pm}3.00e{-}4$	$0.42{\pm}2.88e{-}2$	$1.10{\pm}7.64e{-}2$	1.53±0.11
	15 p	0.00±0.00	$0.30{\pm}2.08e{-}2$	$0.21{\pm}1.45e{-}2$	$1.14{\pm}7.93e{-}2$	$1.11{\pm}7.72e{-}2$
E-PROGRESS (CME)	1 p	$1.76{\pm}7.65e{-}3$	4.79±0.13	11.22±0.44	29.48±0.47	42.91±0.63
	5 p	$2.94{\pm}1.18e{-}2$	14.41±2.93	27.18±2.13	41.34±2.95	53.61±1.94
	15 p	$2.81{\pm}8.05e{-}2$	12.58±0.85	28.24±3.07	45.98±3.51	53.87±1.64
E-ABRUPT (CME)	1 p	$0.44{\pm}1.13e{-}2$	3.73±0.32	11.72±0.79	36.85±0.61	54.76±0.71
	5 p	1.29±0.39	6.12±1.09	24.10±3.45	45.64±3.55	70.93±1.24
	15 p	0.84±0.11	8.94±1.17	20.98±2.18	50.83±2.96	63.40±1.33
E-ABRUPT (ABEBC)	1 p	$6.46e{-}3{\pm}4.22e{-}3$	1.44±0.41	6.73±1.28	25.87±1.13	38.02±1.47
	5 p	0.61±0.40	3.46±1.17	18.69±3.78	33.53±4.13	54.01±1.44
	15 p	0.34±0.11	6.47±1.13	16.14±2.34	38.43±3.23	46.81±1.36
E-ABRUPT (ITEL)	1 p	$1.03{\pm}2.83e{-}2$	13.23±1.24	106.59±14.28	200.00±0.00	200.00±0.00
	5 p	9.90±6.31	41.41±14.42	168.91±12.94	200.00±0.00	200.00±0.00
	15 p	1.21±0.14	71.01±15.86	164.81±11.84	200.00±0.00	200.00±0.00
E-CHAOS (CME)	1 p	$21.00{\pm}3.87e{-}2$	35.78±0.12	53.34±0.12	99.68±0.28	135.43±0.43
	5 p	$14.51{\pm}9.37e{-}2$	34.82±0.24	57.27±1.27	101.49±1.38	128.91±1.70
	15 p	$10.95{\pm}7.57e{-}2$	31.10±0.28	50.88±0.64	95.38±3.25	117.05±3.59
E-CHAOS (ABEBC)	1 p	$2.98{\pm}4.52e{-}2$	16.60±0.11	33.87±0.14	82.31±0.29	120.55±0.43
	5 p	$4.12{\pm}7.96e{-}2$	19.68±0.24	40.87±0.90	86.08±1.18	116.60±1.52
	15 p	$3.34{\pm}6.49e{-}2$	18.50±0.25	37.53±0.45	81.87±2.63	106.28±3.24
E-CHAOS (ITEL)	1 p	$1.14{\pm}1.16e{-}2$	$4.85{\pm}1.79e{-}2$	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	$1.49{\pm}2.41e{-}2$	$4.78{\pm}1.63e{-}2$	$5.00{\pm}3.41e{-}4$	5.00±0.00	5.00±0.00
	15 p	$1.19{\pm}1.77e{-}2$	$4.73{\pm}2.06e{-}2$	$5.00{\pm}3.41e{-}4$	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	$3.50{\pm}1.21e{-}2$	$18.29{\pm}5.22e{-}2$	47.09±0.20	192.69±0.69	278.04±0.79
	5 p	$4.45{\pm}1.34e{-}2$	22.77±0.76	46.28±1.52	159.33±0.86	257.75±0.75
	15 p	$5.28{\pm}7.38e{-}2$	20.96±0.36	43.78±0.57	161.89±0.86	222.30±0.72

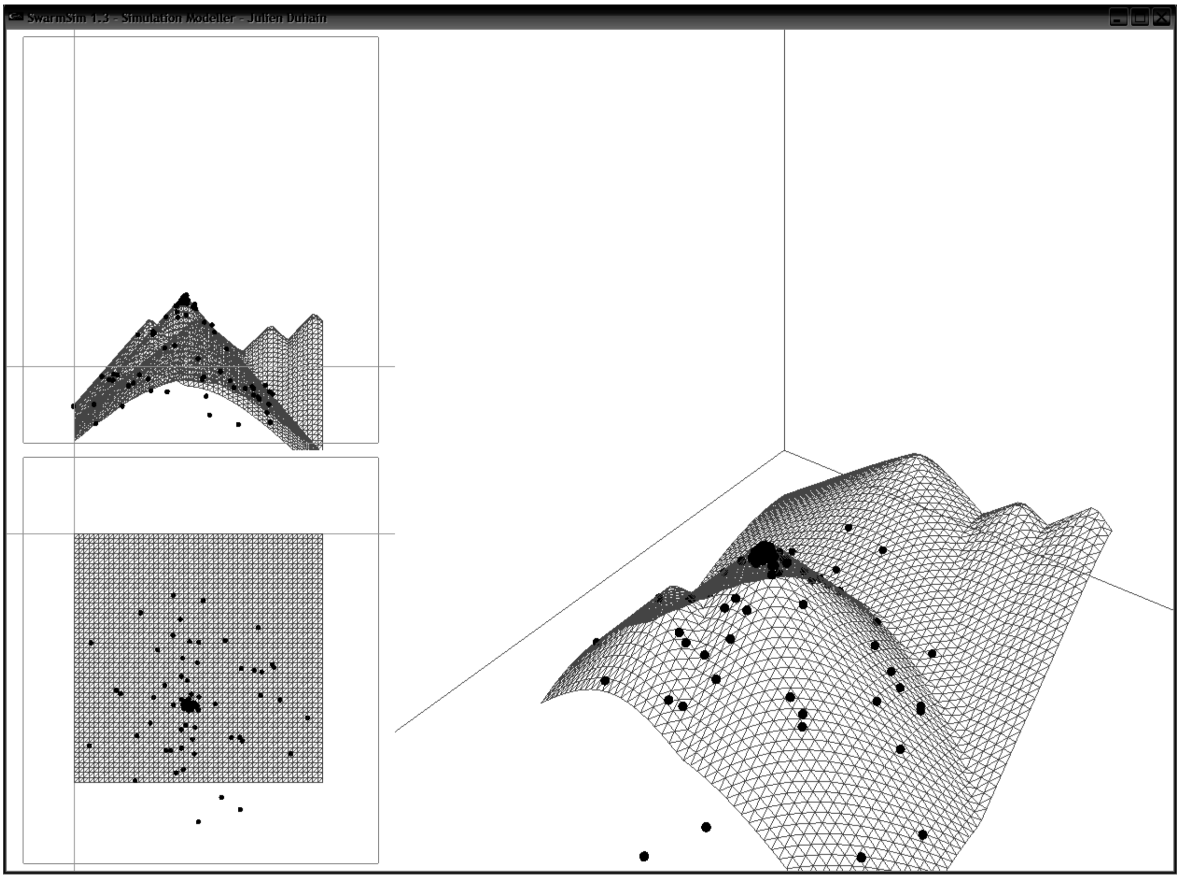


Figure 8.3: QSO (radius 50, 50% quantum) for E-PROGRESS

8.4.2 Analysis of Results

As can be seen in figure 8.3, the quantum particles are evenly distributed within the quantum radius. The swarm therefore has a different behaviour from the APSO where charged particles are more likely to be found near the *gbest* than away from it. The p-values determining the significance of the difference between the results of the re-evaluating PSO and those of the QSO algorithm are listed in tables C.12 to C.21. This section analyses the results for QSO with radius five separately from those of QSO with radius 50.

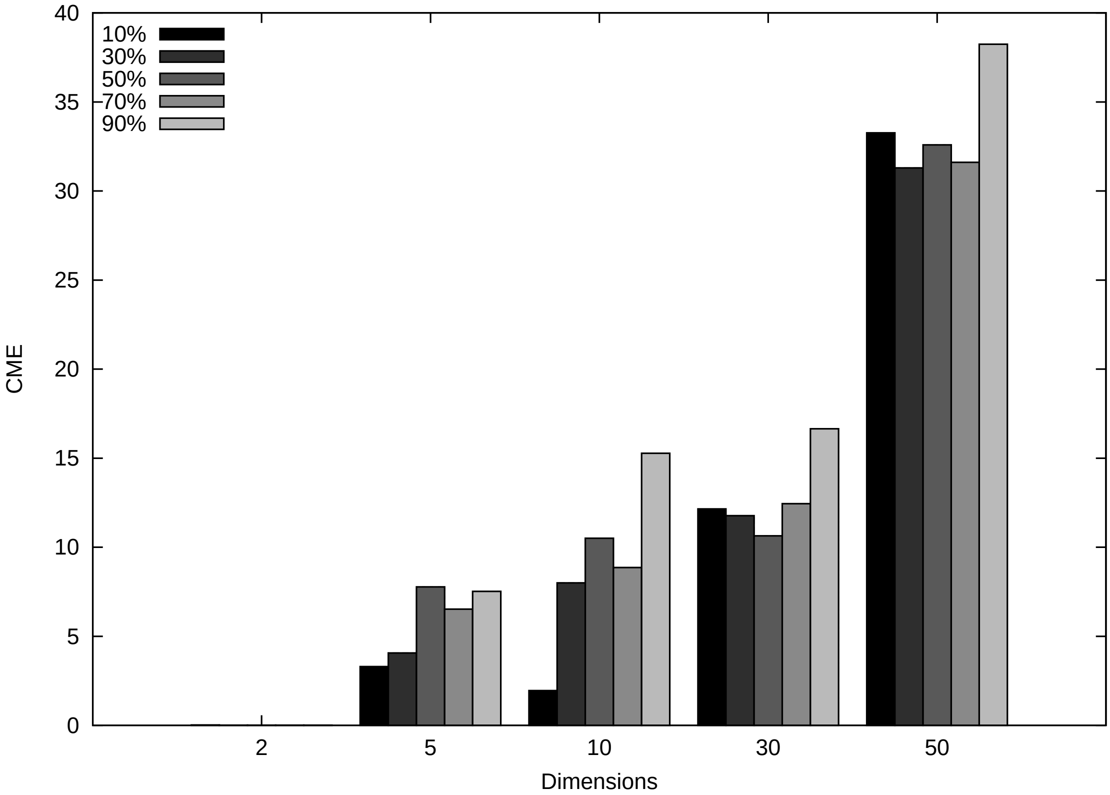
QSO with Radius 5

With a small quantum radius, the quantum particles always exploit around the *gbest* while the neutral particles are left to explore the search space. However, if the neutral particles converge within the quantum radius, the neutral particles can exploit the optimum while the quantum particles maintain a minimum level of diversity in a swarm.

For all environments and for all percentages of quantum particles evaluated, the performance of the QSO algorithm decreased as the number of dimensions increased.

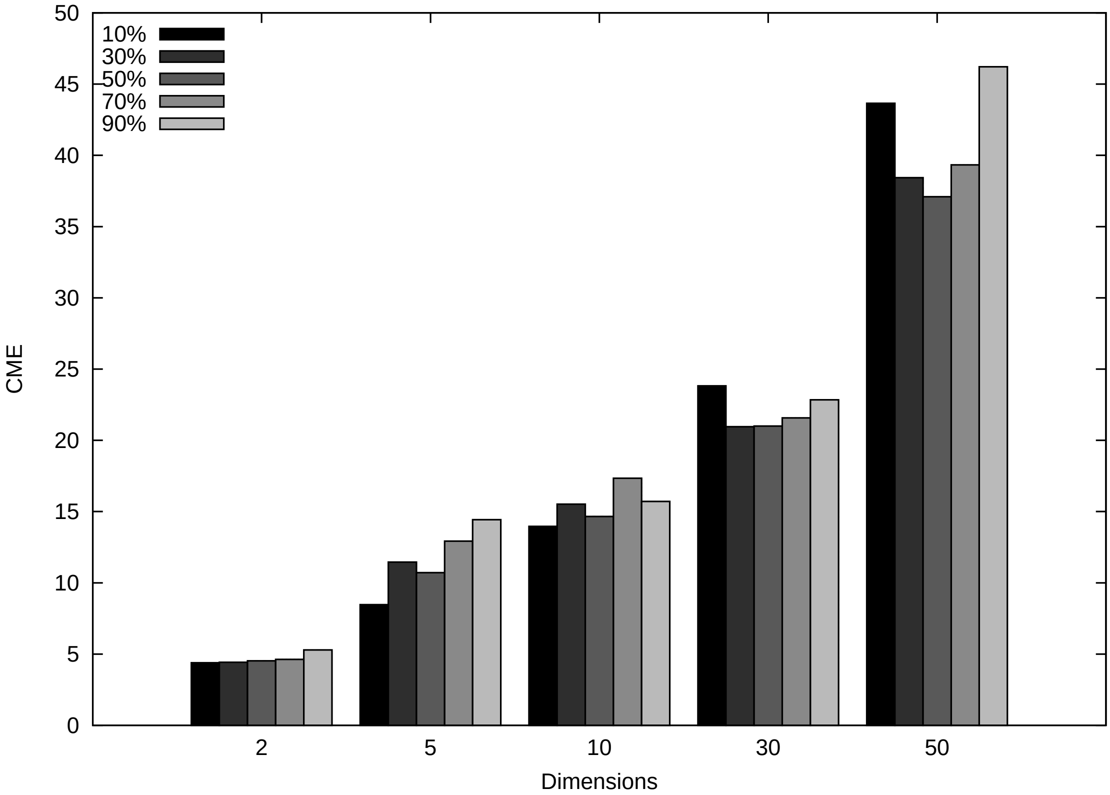
For E-STATIC, QSO (10%) generally gave the best results on multimodal problems of 10 dimensions or less. For the rest of the problems, QSO (30%) and QSO (50%) generally gave the best results. Graph 8.13 illustrates the effect that the quantum ratio has on the CME for E-STATIC environments with five peaks. For the unimodal environments and for the environments of 30 or 50 dimensions, QSO obtained a lower CME than the standard PSO. However, the error obtained by the standard PSO for these environments was not necessarily higher than that of QSO, indicating that QSO found good solutions faster that the standard PSO but did not necessarily find better solutions.

For E-PROGRESS environments, the CME was higher than for E-STATIC, and as for the re-evaluating PSO, the CME tended to rise proportionally to the number of peaks. For environments with less than 10 dimensions and for multimodal environments with 10 dimensions, the percentage of quantum particles influenced the performance negatively. However, for the E-PROGRESS environments with more than 10 dimensions and for the 10-dimensional unimodal E-PROGRESS environment, QSO (50%) generally gave the best results. Graph 8.14 illustrates the effect that the quantum ratio had on the CME for E-PROGRESS environments with five peaks. With the most efficient quantum ratio, QSO outperformed the re-evaluating PSO for all but the multimodal environments with 10 dimensions or less. Because the quantum particles are always located around the *gbest*, the swarm quickly started exploiting the area within the quantum radius and the neutral particles were likely to rapidly converge towards the quantum cloud but further exploitation within the quantum cloud was not necessarily helped by the quan-



Graph 8.13: QSO with quantum radius of five and various quantum ratios for E-STATIC with five peaks

tum particles as these keep being randomly relocated within the quantum radius. It is therefore possible that, for environments of low dimensionality, the re-evaluating PSO could exploit the optimum further than QSO because all particles of the re-evaluating PSO were allowed to participate in the exploitation. Also, if the neutral particles have converged within the quantum radius, a global optimum that appears outside the quantum radius is likely to go undetected as no particle is left to explore outside the (small) quantum radius. However, for environments of high dimensionality where the swarm of the re-evaluating PSO converged at a slower pace, the quantum particles allowed the QSO algorithm to exploit a peak faster than the re-evaluating PSO. Furthermore, in high dimensionality environments, the neutral particles take longer to converge and can therefore explore outside the quantum radius while the quantum particle exploit the best

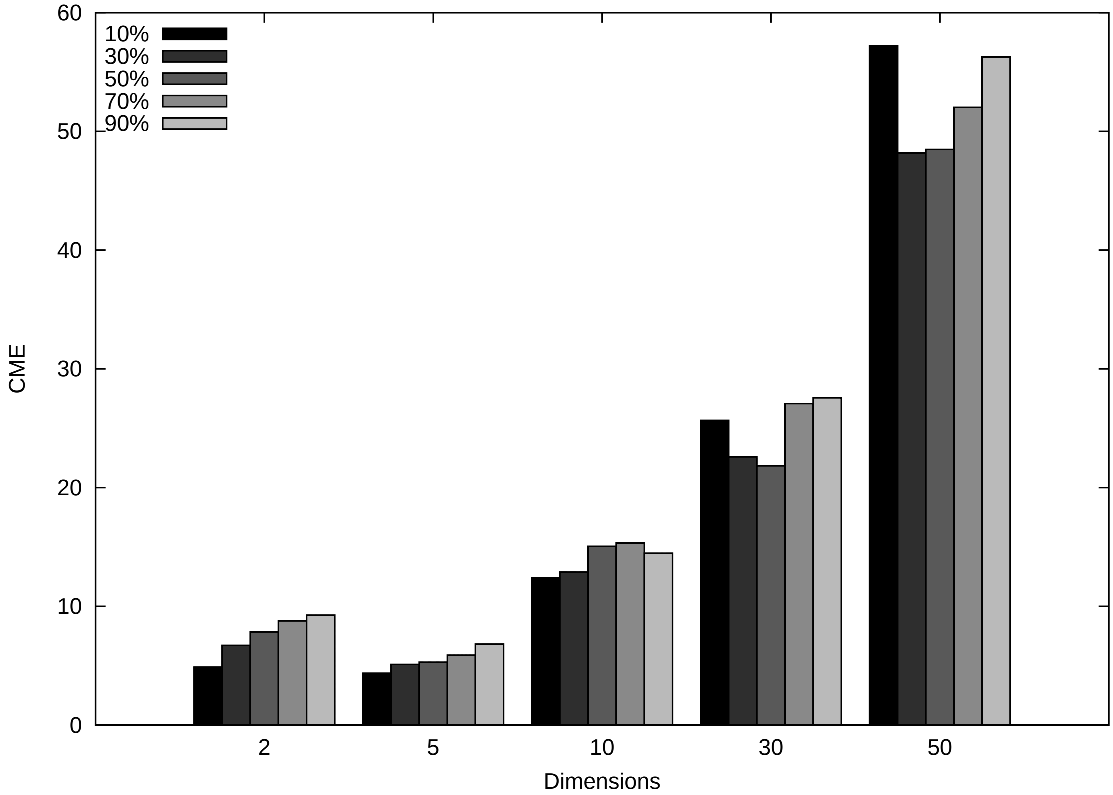


Graph 8.14: QSO with quantum radius of five and various quantum ratios for E-PROGRESS with five peaks

known peak.

For E-ABRUPT, the CME was higher than for E-STATIC but not necessarily higher than for E-PROGRESS. QSO (10%) obtained the lowest CME on multimodal problems of 10 or less dimensions but QSO with a greater ratio of quantum particles obtained a lower CME on unimodal problems and multimodal problems with 30 and 50 dimensions. Graph 8.15 illustrates the effect that the quantum ratio had on the CME for E-ABRUPT environments with five peaks. For the multimodal E-ABRUPT environments with 10 dimensions or less, the QSO algoritm's CME was lower than that of the re-evaluating PSO but the ABEBC of the re-evaluating PSO was lower, showing that QSO reacted quickly after a change but did not exploit as much as the re-evaluating PSO. Also, for these environments, the ITEL measurements of the QSO algorithm were lower that those of the re-evaluating PSO only for the E-ABRUPT environments with five dimensions and five peaks and with 10 dimensions and 15 peaks. This can indicate that the QSO swarm is more prone to becoming trapped in a small local optimum, in which case the error limit of 10 cannot be reached. For most of the unimodal problems and multimodal problems with 30 and 50 dimensions, the CME was generally lowest with 50% of quantum particles. For these environments, QSO performed significantly better than the re-evaluating PSO with reference to CME, ABEBC and ITEL. After a change in an abruptly changing environment, the swarm moves quickly to the top of the closest peak as all quantum particles are immediately placed next to the *gbest*. Graph 8.16 illustrates the progression of the error and diversity for an E-ABRUPT environment with two dimensions and five peaks. The error dropped quickly after a change but remained relatively far from zero between iteration 200 and 600 because the swarm became trapped in a local optimum after iteration 200 and again after iteration 400. The swarm's diversity never fell below a certain level due to the quantum particles never converging. This allowed the swarm to quickly recover from a change and climb up the closest peak. However, less diversity was generated after a change for QSO than for the re-evaluating PSO because the quantum particles always remain within the quantum radius. Only the neutral particles can explore outside the quantum radius which makes the QSO swarm prone to becoming trapped in a local optimum.

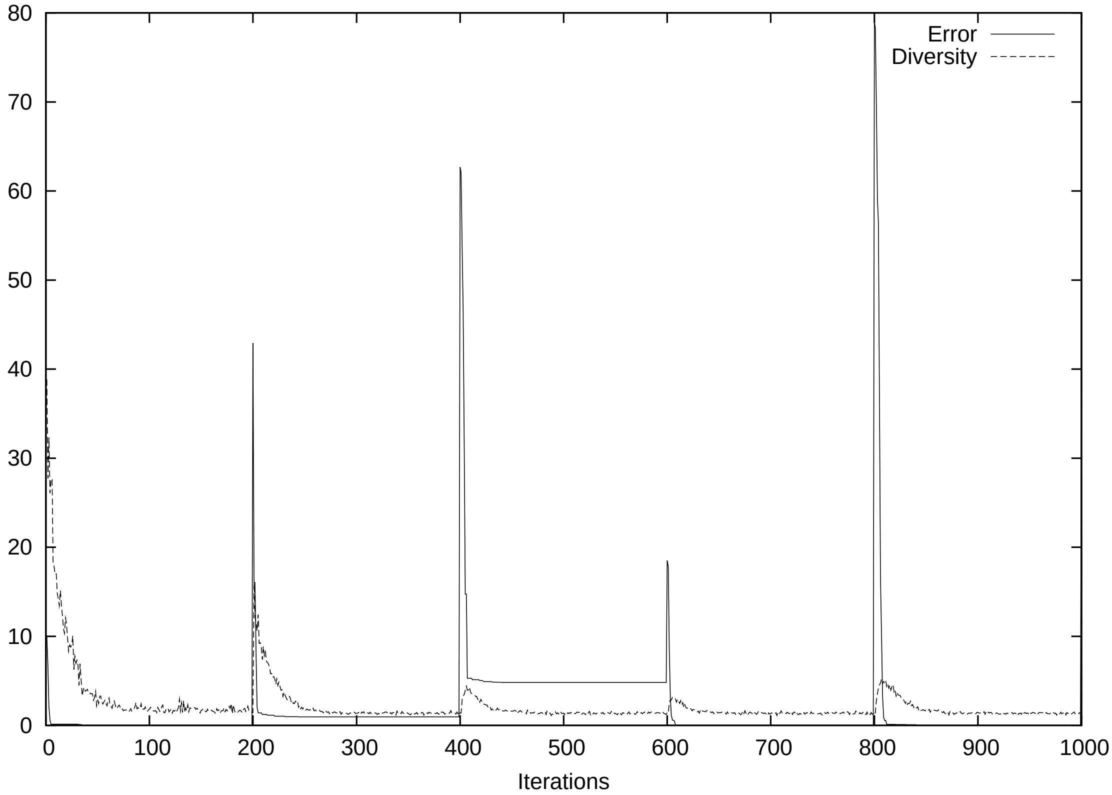
For E-CHAOS, the CME was higher than for E-PROGRESS and E-ABRUPT, and the performance generally improved proportionally to the number of peaks in the environment. QSO (30%) gave the best results on most problems although QSO (10%) had a slightly lower CME on the five-dimensional problems and the 10-dimensional problems with five peaks and QSO (50%) had a lower CME on 50-dimensional, multimodal problems. Graph 8.17 illustrates the effect that the quantum ratio had on the CME for E-CHAOS environments with five peaks. QSO (30%) outperformed the re-evaluating PSO for all E-CHAOS environments with reference to CME, ABEBC and ITEL (where applicable). In chaotically changing environments, the swarm is given a small amount of time to locate and exploit the optimum, with the quantum particles always located around the *gbest*, QSO can quickly exploit a peak while some of the neutral particles are



Graph 8.15: QSO with quantum radius of five and various quantum ratios for E-ABRUPT with five peaks

still exploring the search space.

For E-PATTERN, the CME increased proportionally to the number of peaks for the two-dimensional environments and increased in inverse proportion to the number of peaks for the environments with 10 or more dimensions. For E-PATTERN with two dimensions, the performance level was not significantly influenced by the ratio of quantum particles and was similar to that of the re-evaluating PSO for the environments with one and five peaks. However, for two-dimensional E-PATTERN with 15 peaks the CME rose with the quantum ratio and was higher than the re-evaluating PSO's CME. For E-PATTERN with 5 and 10 dimensions, QSO (10%) and (30%) obtained the lowest CME and outperformed the re-evaluating PSO. For E-PATTERN with 30 and 50 dimensions,

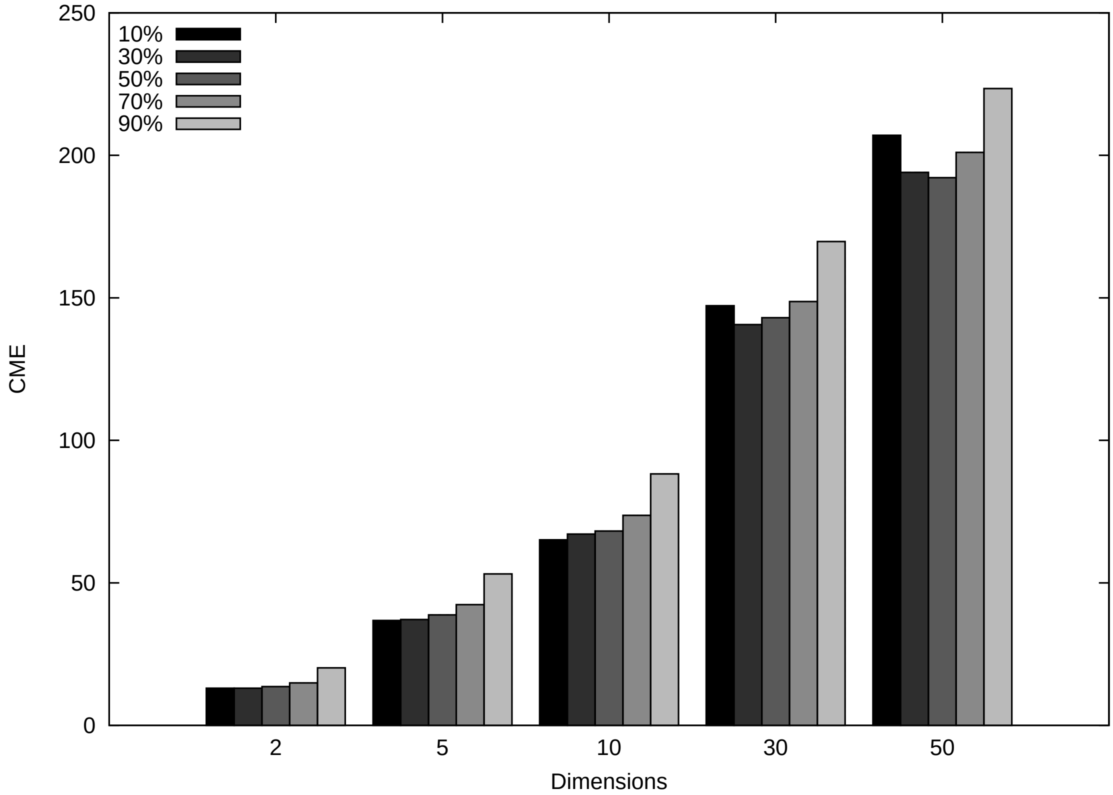


Graph 8.16: QSO with quantum radius of five and quantum ratio of 50% for E-ABRUPT

better performance were achieved with the lowest ratio of quantum particles and the re-evaluating PSO generally outperformed QSO. Graph 8.18 illustrates the effect that the quantum ratio had on the CME for E-PATTERN environments with five peaks.

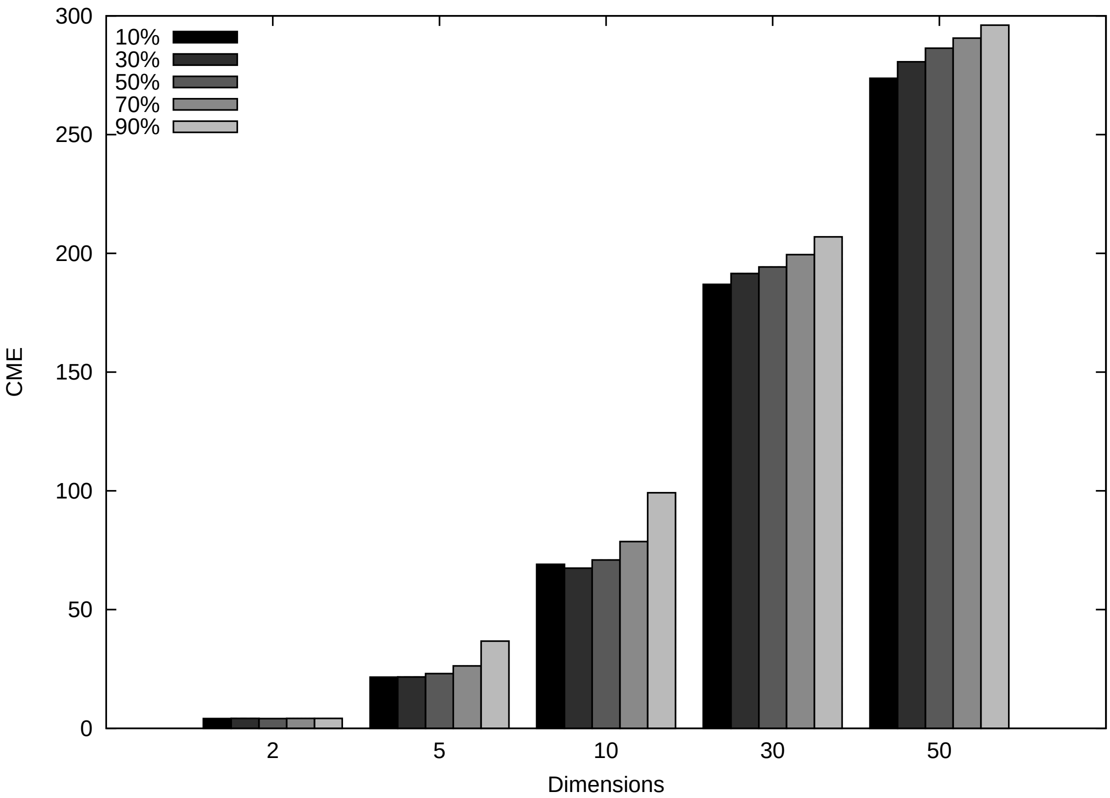
QSO with Radius 50

With a large quantum radius, the diversity always remains high which promotes exploration as illustrated in graph 8.19 for an E-ABRUPT environment with two dimensions and five peaks. This helps QSO to find the global optimum but slows down the exploitation as quantum particles do not converge. The comparison between graphs 8.19 and 8.10 is shown in graph 8.20 and illustrates that the diversity level is more stable when



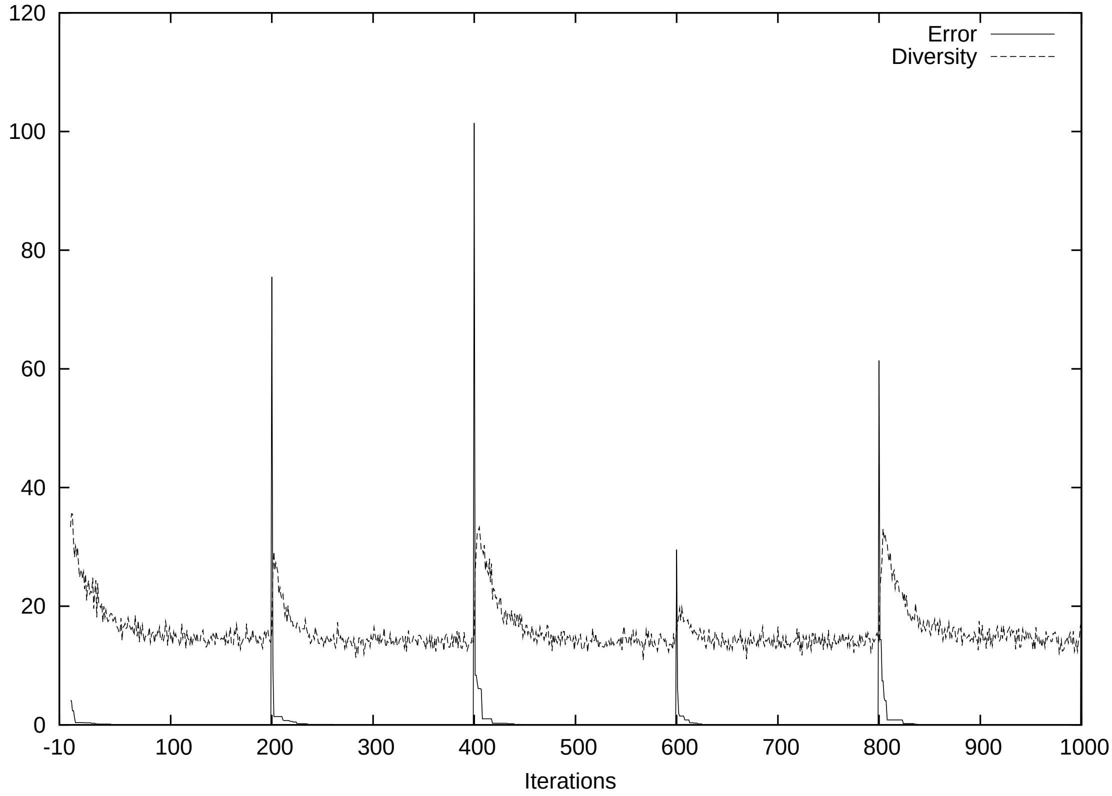
Graph 8.17: QSO with quantum radius of five and various quantum ratios for E-CHAOS with five peaks

QSO is used than for APSO. The quantum particles of the QSO algorithm are always uniformly distributed within an hypershpere of pre-defined radius. On the other hand, the charged particles of the APSO are free to travel across the search space and their velocity is influenced by the surrounding charged particles. Therefore, the average distance between the quantum particles being only dependent on a uniform distribution tends to be more constant than the distance between the charged particles. The diversity of the QSO swarm being based on the average distance between the particles is therefore more stable through the simulation. In contrast with QSO with a small quantum radius, the quantum particles of QSO with a large radius only promote exploration, and exploitation is left to the neutral particle.



Graph 8.18: QSO with quantum radius of five and various quantum ratios for E-PATTERN with five peaks

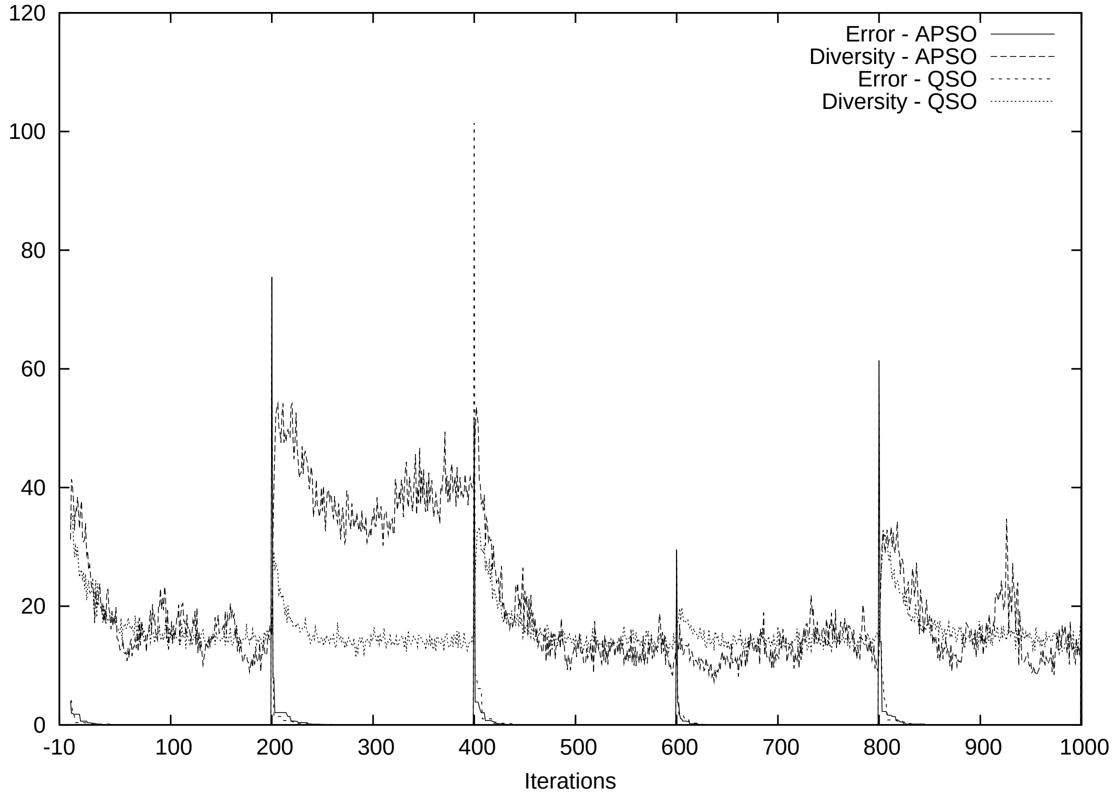
For E-STATIC, the performance level varied with the ratio of quantum particles in the swarm. For the multimodal environments with 10 dimensions or less, the lowest CME was obtained with 10% of quantum particles and the standard PSO outperformed QSO. For the unimodal environments and for environments with five peaks and 30 or 50 dimensions, QSO (50%) obtained the lowest CME and outperformed the standard PSO. For environments with 15 peaks and 30 or 50 dimensions, QSO (30%) obtained the lowest CME and outperformed the standard PSO. However, the final error of the QSO algorithm was generally higher than that of the standard PSO and tended to increase with the quantum ratio. This confirms that exploitation is limited with a larger quantum ratio since only the neutral particles can exploit the optimum. Graph 8.21 illustrates the effect that the quantum ratio had on the CME for the E-STATIC environments with



Graph 8.19: QSO with quantum radius of 50 and quantum ratio of 50% for E-ABRUPT

five peaks.

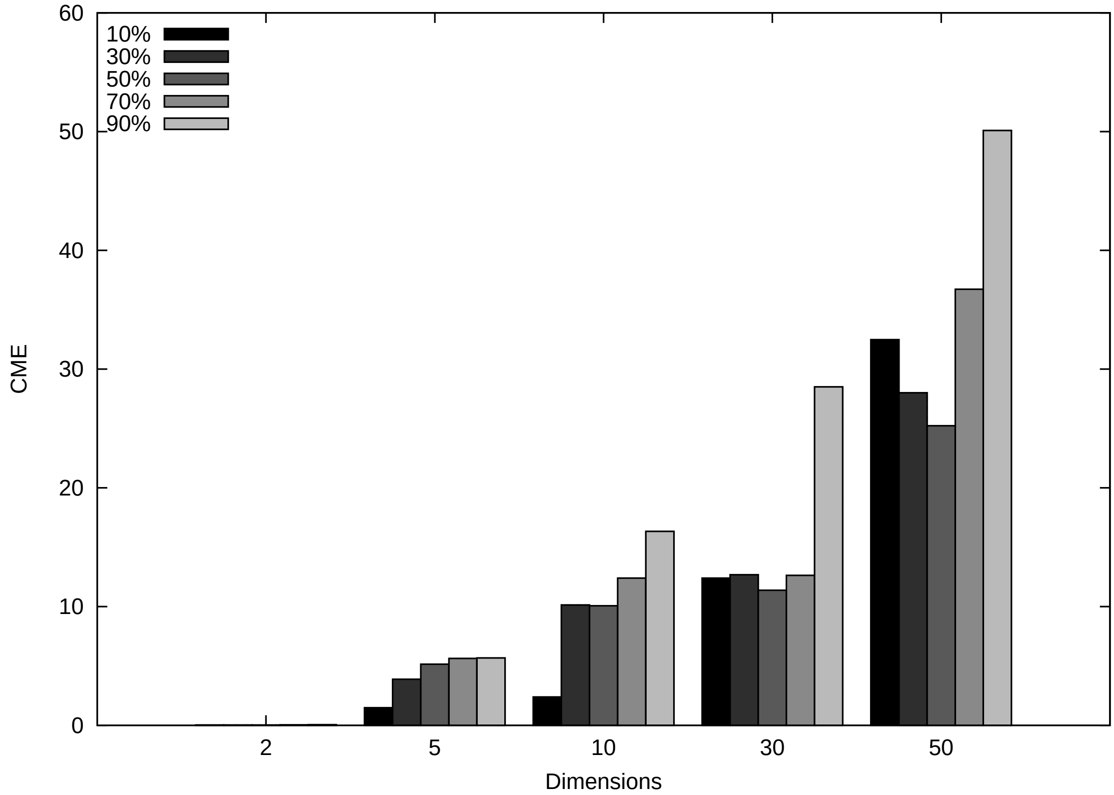
For E-PROGRESS, the performance level was lower than for E-STATIC. The lowest CME was obtained with 50% of quantum particles or less, and, with the most efficient quantum ratio, QSO outperformed the re-evaluating PSO for all but the 10-dimensional multimodal problems. Graph 8.22 illustrates the effect that the quantum ratio had on the CME for the E-PROGRESS environments with five peaks. However, the selection of the most effective quantum ratio seems to be dependent on a combination of the dimensionality and the number of peaks in the environment. QSO performed better on unimodal environments but, unlike the re-evaluating PSO, the CME of the QSO algorithm did not necessarily increase proportionally to the number of peaks in the environment. The better performance of QSO compared to the re-evaluating PSO can



Graph 8.20: Comparison between QSO and APSO for E-ABRUPT

be explained by the better peak detection capacity of QSO with a large quantum radius.

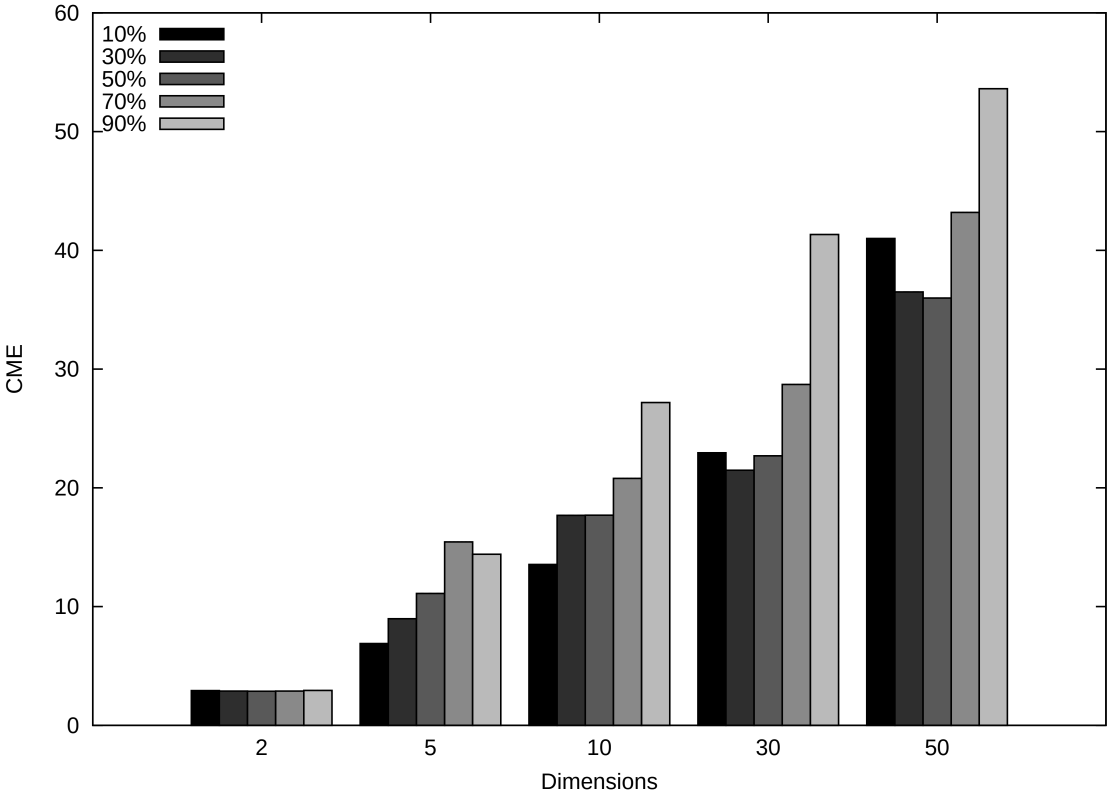
For E-ABRUPT, the CME was higher than for E-STATIC. QSO performed better than for E-PROGRESS for environments with two or five dimensions but worse for environments with 30 or 50 dimensions. QSO (10%) generally gave the best results on multimodal problems with 10 dimensions or less and QSO (30%) generally gave the best results on unimodal problems and multimodal problems with 30 or 50 dimensions. Graph 8.23 illustrates the effect that the quantum ratio had on the CME for the E-ABRUPT environments with five peaks. With the most effective ratio of quantum particles, QSO obtained a lower CME than the re-evaluating PSO. For most E-ABRUPT environments, the ABEBC and ITEL of QSO were lower than that of the re-evaluating PSO. However, for environments with 15 peaks and five to 30 dimensions and environments with five



Graph 8.21: QSO with quantum radius of 50 and various quantum ratios for E-STATIC with five peaks

peaks and 10 or 30 dimensions, the ABEBC and ITEL were lower for the re-evaluating PSO. This indicates that for these environments, although QSO reacted faster to a change by quickly exploiting a peak, QSO could not exploit as well as the re-evaluating PSO, possibly because of the smaller number of (neutral) particles taking part in the exploitation.

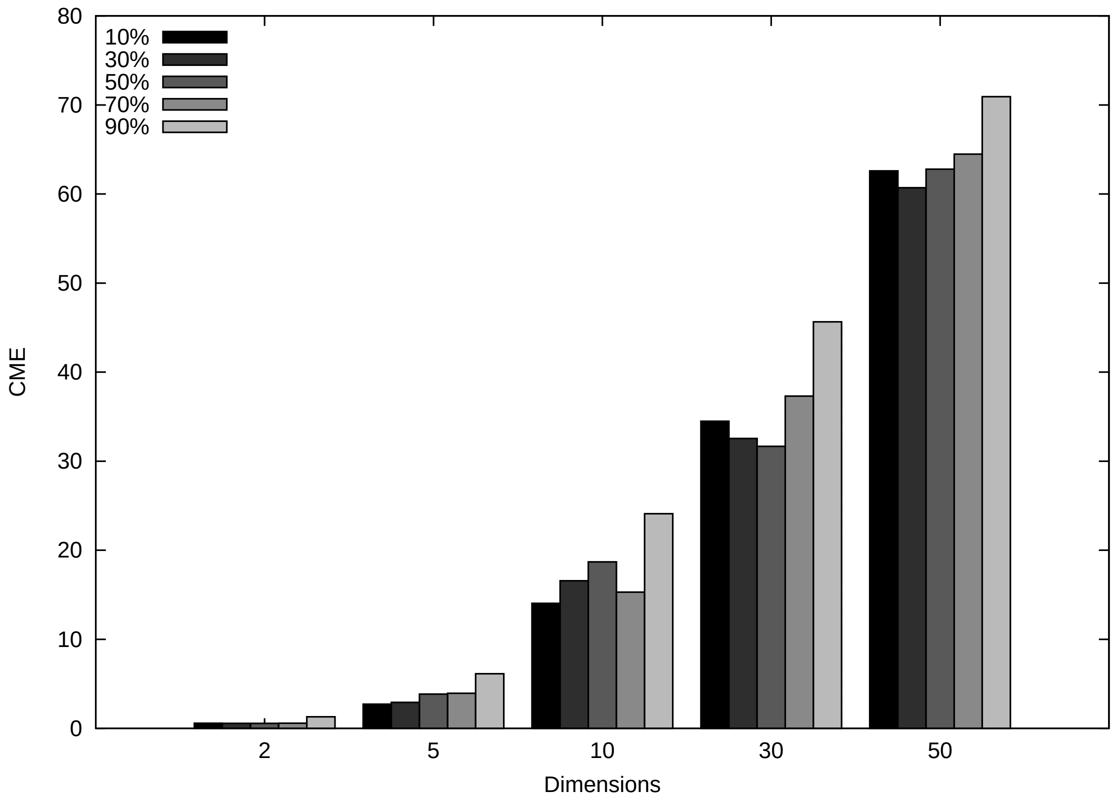
For E-CHAOS, the CME was higher than for E-PROGRESS and E-ABRUPT. The performance generally improved proportionally to the ratio of quantum particles. QSO (90%) obtained the lowest CME for unimodal environments and multimodal environments with 30 and 50 dimensions. For the multimodal E-CHAOS with 10 or less dimensions, a quantum ratio of 30% to 70% gave the lowest CME. Graph 8.24 illustrates



Graph 8.22: QSO with quantum radius of 50 and various quantum ratios for E-PROGRESS with five peaks

the effect that the quantum ratio had on the CME for the E-CHAOS environments with five peaks. With the most efficient quantum ratio, QSO outperformed the re-evaluating PSO with regards to CME, ABEBC and ITEL. After a change, the large quantum radius promotes exploration and allows a quick detection of the new optimum. Unlike the re-evaluating PSO, the QSO algorithm's performance did not necessarily improve proportionally to the number of peaks.

For E-PATTERN, a low quantum ratio gave the best results on problems with few dimensions while using a higher percentage of quantum particles was more efficient for environments with high dimensionality. QSO (90%) obtained the best results for most of the E-PATTERN environments. Graph 8.25 illustrates the effect that the quantum

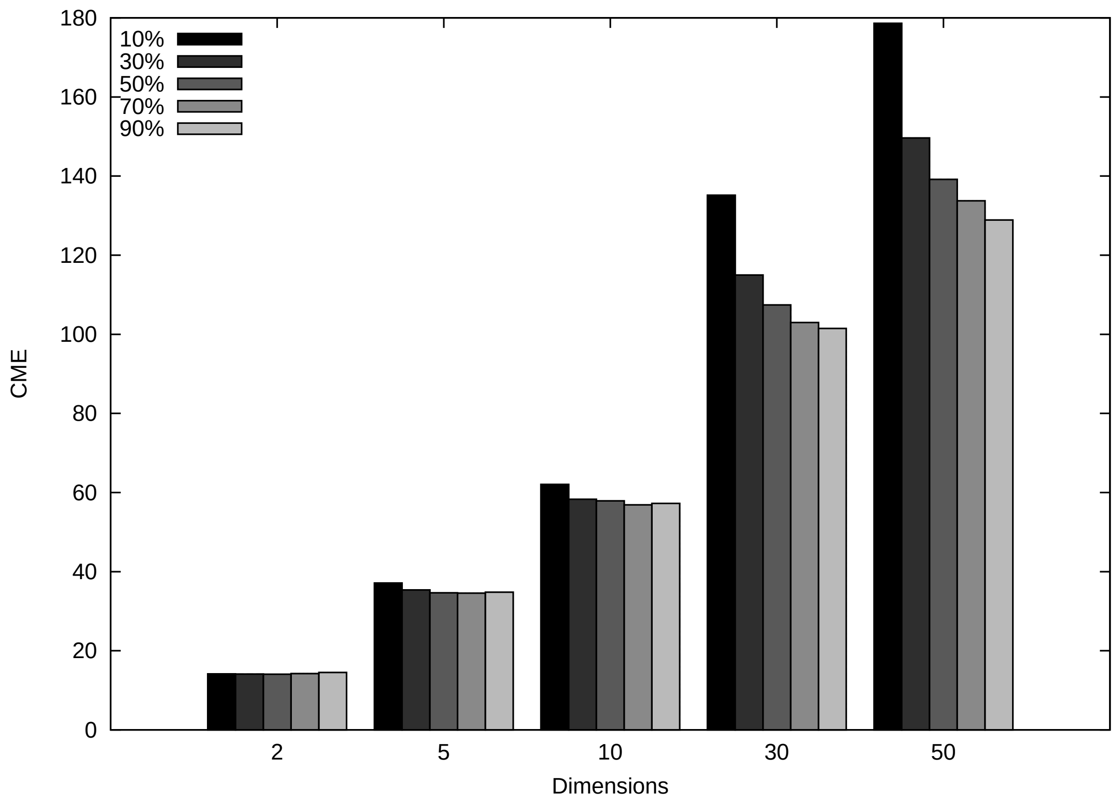


Graph 8.23: QSO with quantum radius of 50 and various quantum ratios for E-ABRUPT with five peaks

ratio had on the performance for the E-PATTERN environments with five peaks. With the most efficient quantum ratio, QSO outperformed the re-evaluating PSO for all E-PATTERN environments.

8.4.3 Summary of Strengths and Weaknesses

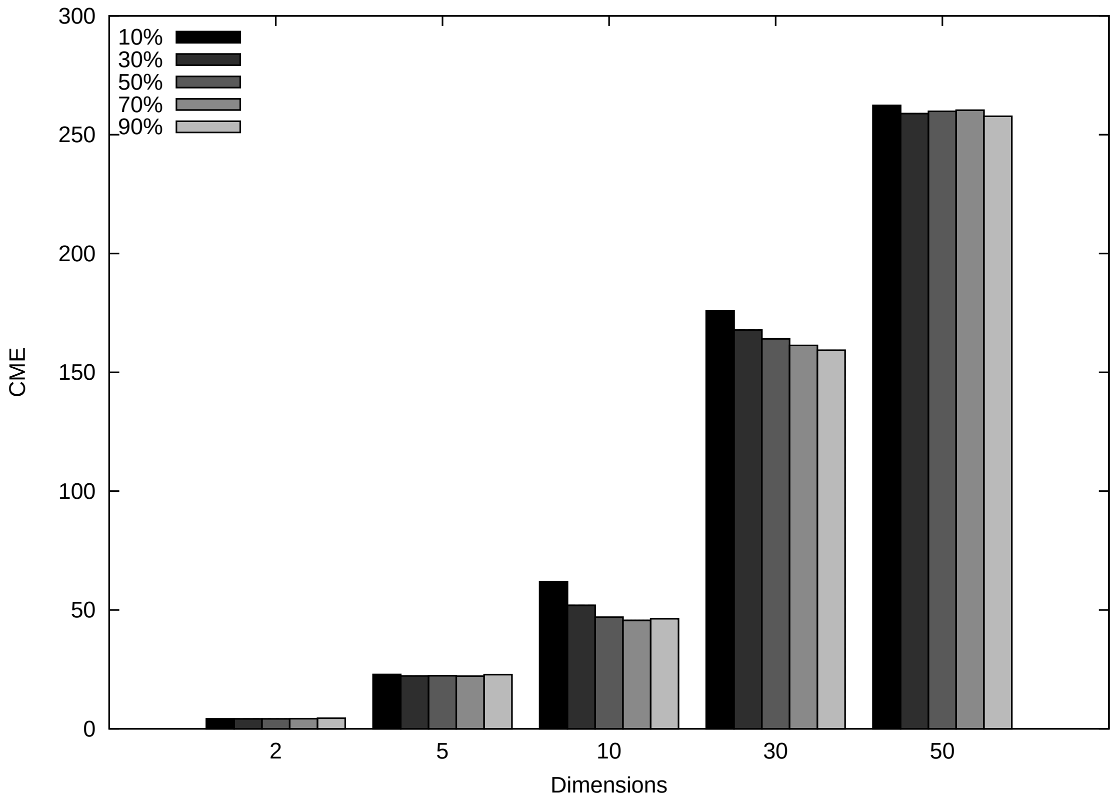
With a small quantum radius, QSO was able to track peaks, but sometimes failed to detect the appearance of new optima. The results indicate that the QSO algorithm quickly exploited the first peak found and outperformed the re-evaluating PSO for unimodal problems of all types. QSO also outperformed the re-evaluating PSO for progressively



Graph 8.24: QSO with quantum radius of 50 and various quantum ratios for E-CHAOS with five peaks

changing environments of high dimensionality. For abruptly changing environments, QSO had a better reactivity than the re-evaluating PSO, but for multimodal environments with low dimensionality, the re-evaluating PSO had a better exploitation capacity. QSO showed to be very effective for chaotically changing environments in comparison to the re-evaluating PSO.

With a large radius, QSO was able to track peaks and detect the appearance of new optima. The results indicate that, for static environments, a large quantum radius can slow down exploitation. For progressively changing environments, QSO was particularly effective for multimodal environments and high dimensional environments in comparison to the re-evaluating PSO. For abruptly changing environments, QSO showed good peak



Graph 8.25: QSO with quantum radius of 50 and various quantum ratios for E-PATTERN with five peaks

detection capacity and good reactivity but the exploitation capacity of QSO was sometime worse than that of the re-evaluating PSO. QSO outperformed the re-evaluating PSO for chaotically changing environments.

8.5 Multi-swarm

This section provides and analyses the results of applying the multi-swarm algorithm described in section 4.8.4 to the test environments. Multi-swarm uses swarm sub-division and parallel tracking of optima to overcome diversity loss. Two configurations of the multi-swarm algorithm are evaluated. The first configuration uses five sub-swarms of 20 particles and the second configuration uses 10 sub-swarms of 10 particles so that both multi-swarms have a total population of 100 particles. Blackwell and Branke [11] have shown that using quantum swarms as sub-swarms is preferable to using CPSO or APSO. QSO sub-swarms are therefore used in the experiments. The QSO sub-swarms are implemented as in algorithm 4.3 with 50% of quantum particles and use the parameters selected in 6.4, however with a different population size. Ideally, each sub-swarm should converge quickly on a different peak, exploration being handled by the anti-convergence operator. It has been shown in section 8.4.2 that QSO with a quantum radius of five tends to converge quickly and is prone to become trapped in local optima (which is desired in this case). Therefore, a quantum radius of five was selected for the QSO sub-swarms.

Having an exclusion radius smaller than the quantum radius would allow two sub-swarms to have their quantum clouds overlap which makes little sense since peaks appearing within the quantum radius of a sub-swarm have a good chance to be detected by the quantum particles. On the other hand, having an exclusion radius larger than the quantum radius is not a good idea either. Indeed, with $r_{cloud} < r_{excl}$, a peak can appear within the exclusion radius of a sub-swarm after the neutral particles of that sub-swarm have converged within its quantum radius. In such scenarios, if another sub-swarm attempts to explore the appearing peak, one of the two sub-swarms will be reinitialised, even though the sub-swarms are exploring different peaks. The exclusion radius was therefore set to five to match the quantum radius.

Blackwell and Branke [8, 12] suggested the anti-convergence radius be smaller or equal to the exclusion radius but large enough to keep at least one sub-swarm "patrolling" at all time. According to Blackwell and Branke, the lowest possible value for r_{excl} should be based on the number of evaluations needed for the swarm to converge on a peak. However, since environments differ in change severity and dimensionality, the speed of convergence of a swarm varies for the various test environments making a lower bound for r_{excl} difficult to define. To match the exclusion radius, the anti-convergence radius was therefore set to five for the experiments conducted in this section.

8.5.1 Results

The results in tables 8.21 and 8.22 were obtained by an implementation of algorithm 4.4 using the configurations described above. The number of sub-swarms used is mentioned in the legend of the tables. The code for multi-swarm can be found in CIlib [74].

8.5.2 Analysis of Results

Additionally to being compared to the results of the benchmark algorithms, the results of multi-swarm are also compared to those of the QSO algorithm with quantum radius five and quantum ratio 50% since this QSO uses the same configuration as the sub-swarms and could be considered as the special case of a multi-swarm with a single sub-swarm of 100 particles. In the text, the QSO algorithm with a quantum radius of five and a quantum ratio of 50% is referred to as QSO (5,50%). Multi-swarm with five sub-swarms of 20 particles is referred to as multi-swarm (5) and the configuration with 10 sub-swarms of 10 particles is referred to as multi-swarm (10).

The p-values determining the significance of the difference between the results of the re-evaluating PSO and those of multi-swarm are listed in tables C.22 and C.23. The p-values determining the significance of the difference between the results of QSO (5,50%) and those of multi-swarm are listed in tables C.24 and C.25.

For all test environments, the performance level worsened as the number of dimensions increased.

For E-STATIC, the number of peaks did not clearly affect the performance positively or negatively. The multi-swarm (5) performed better than the multi-swarm (10) for all E-STATIC environments except for the 10-dimensional E-STATIC environment with 15 peaks. The multi-swarm (5) also performed as well as or better than the standard PSO for all E-STATIC environments. The QSO (5,50%) obtained a lower CME for the unimodal environment. For the multimodal environments, the multi-swarm algorithm's

Table 8.21: Experimental results of multi-swarms with five sub-swarms

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$1.91e{-}2\pm2.97e{-}3$	$0.28\pm2.50e{-}2$	1.70 ± 0.11	13.65 ± 0.55	33.78 ± 0.89
	5 p	$2.09e{-}2\pm3.07e{-}3$	$0.43\pm5.75e{-}2$	1.71 ± 0.14	12.86 ± 0.42	36.78 ± 1.34
	15 p	$2.92e{-}2\pm6.31e{-}3$	$0.28\pm2.44e{-}2$	2.20 ± 1.03	16.50 ± 1.21	35.62 ± 1.14
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	$9.47e{-}16\pm6.58e{-}17$	$7.04e{-}5\pm4.89e{-}6$	$2.51e{-}2\pm1.74e{-}3$
	5 p	0.00 ± 0.00	0.00 ± 0.00	$9.47e{-}16\pm6.58e{-}17$	$9.08e{-}4\pm6.31e{-}5$	$4.99e{-}2\pm3.47e{-}3$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$7.61e{-}13\pm5.28e{-}14$	$9.63e{-}3\pm6.69e{-}4$	$1.83e{-}2\pm1.27e{-}3$
E-PROGRESS (CME)	1 p	$1.75\pm6.32e{-}3$	$2.57\pm3.26e{-}2$	$5.00\pm8.71e{-}2$	20.35 ± 0.49	42.72 ± 1.36
	5 p	$2.04\pm7.03e{-}3$	$3.20\pm5.45e{-}2$	5.85 ± 0.14	19.86 ± 0.48	42.34 ± 1.41
	15 p	$2.29\pm8.86e{-}2$	6.28 ± 0.86	9.02 ± 1.12	23.80 ± 0.83	41.23 ± 1.31
E-ABRUPT (CME)	1 p	$0.34\pm7.70e{-}3$	$1.74\pm5.36e{-}2$	5.31 ± 0.13	23.74 ± 0.39	45.90 ± 0.93
	5 p	$0.42\pm1.09e{-}2$	$2.58\pm9.45e{-}2$	6.90 ± 0.20	25.85 ± 0.59	55.90 ± 1.87
	15 p	0.58 ± 0.11	3.04 ± 0.41	7.07 ± 0.48	27.54 ± 0.99	49.75 ± 1.23
E-ABRUPT (ABEBC)	1 p	$1.84e{-}7\pm9.56e{-}8$	$4.09e{-}4\pm8.71e{-}5$	$4.63e{-}2\pm7.65e{-}3$	$2.37\pm7.98e{-}2$	7.16 ± 0.30
	5 p	$4.30e{-}7\pm2.18e{-}7$	$4.34e{-}4\pm1.18e{-}4$	$5.65e{-}2\pm1.14e{-}2$	2.90 ± 0.34	13.48 ± 1.45
	15 p	0.13 ± 0.10	0.82 ± 0.37	1.15 ± 0.48	5.41 ± 0.76	11.45 ± 0.98
E-ABRUPT (ITEL)	1 p	$1.17\pm6.83e{-}2$	7.64 ± 0.29	20.91 ± 0.52	70.15 ± 0.68	118.03 ± 1.35
	5 p	1.45 ± 0.12	10.55 ± 0.42	27.19 ± 1.77	79.09 ± 3.88	163.87 ± 8.06
	15 p	1.60 ± 0.18	10.87 ± 1.09	34.92 ± 5.62	105.21 ± 9.21	152.26 ± 6.24
E-CHAOS (CME)	1 p	$20.29\pm3.79e{-}2$	44.54 ± 0.19	75.34 ± 0.38	169.65 ± 1.05	247.75 ± 1.61
	5 p	$12.70\pm6.36e{-}2$	40.26 ± 0.19	68.52 ± 0.49	145.66 ± 1.80	205.18 ± 3.24
	15 p	$9.58\pm3.90e{-}2$	33.67 ± 0.19	58.68 ± 0.30	126.43 ± 1.41	176.25 ± 3.19
E-CHAOS (ABEBC)	1 p	$1.20\pm1.94e{-}2$	21.44 ± 0.19	58.27 ± 0.41	160.81 ± 1.07	241.32 ± 1.61
	5 p	$1.12\pm2.60e{-}2$	22.36 ± 0.22	55.45 ± 0.46	138.59 ± 1.73	200.07 ± 3.15
	15 p	$1.12\pm2.14e{-}2$	19.13 ± 0.17	47.75 ± 0.30	120.79 ± 1.36	172.04 ± 3.12
E-CHAOS (ITEL)	1 p	$1.28\pm1.16e{-}2$	$4.87\pm1.27e{-}2$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.17\pm1.60e{-}2$	$4.88\pm1.36e{-}2$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.05\pm5.64e{-}3$	$4.84\pm1.55e{-}2$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.34\pm1.19e{-}2$	22.77 ± 0.19	88.90 ± 0.41	216.83 ± 0.92	297.35 ± 0.87
	5 p	$4.20\pm1.83e{-}2$	23.45 ± 0.23	74.43 ± 0.42	185.45 ± 0.75	274.09 ± 1.02
	15 p	$4.50\pm1.56e{-}2$	18.83 ± 0.20	59.90 ± 0.31	177.91 ± 0.54	241.12 ± 0.76

Table 8.22: Experimental results of multi-swarms with 10 sub-swarms

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	$3.16e{-2}\pm3.66e{-3}$	$0.40\pm3.95e{-2}$	2.23 ± 0.10	17.55 ± 0.78	44.25 ± 0.98
	5 p	$2.68e{-2}\pm3.98e{-3}$	$0.47\pm5.15e{-2}$	2.08 ± 0.18	17.53 ± 0.60	46.22 ± 1.36
	15 p	$3.37e{-2}\pm5.53e{-3}$	$0.35\pm3.75e{-2}$	1.98 ± 0.15	18.30 ± 0.74	45.04 ± 1.09
E-STATIC (Error)	1 p	0.00 ± 0.00	0.00 ± 0.00	$3.95e{-8}\pm2.74e{-9}$	$4.73e{-2}\pm3.28e{-3}$	$9.75e{-2}\pm6.77e{-3}$
	5 p	0.00 ± 0.00	$1.42e{-15}\pm9.87e{-17}$	$5.45e{-8}\pm3.78e{-9}$	$6.37e{-2}\pm4.42e{-3}$	$0.10\pm6.96e{-3}$
	15 p	0.00 ± 0.00	0.00 ± 0.00	$7.74e{-8}\pm5.37e{-9}$	$7.10e{-3}\pm4.93e{-4}$	$9.99e{-2}\pm6.94e{-3}$
E-PROGRESS (CME)	1 p	$1.83\pm9.61e{-3}$	$2.94\pm3.82e{-2}$	6.00 ± 0.15	25.53 ± 0.68	52.76 ± 1.24
	5 p	$2.12\pm1.22e{-2}$	$3.53\pm5.00e{-2}$	6.53 ± 0.14	22.92 ± 0.53	50.93 ± 1.51
	15 p	$2.20\pm1.26e{-2}$	3.82 ± 0.12	7.00 ± 0.30	25.14 ± 0.96	44.47 ± 1.21
E-ABRUPT (CME)	1 p	$0.37\pm9.78e{-3}$	$1.99\pm6.65e{-2}$	6.77 ± 0.18	30.65 ± 0.66	57.16 ± 1.03
	5 p	$0.44\pm1.39e{-2}$	$2.37\pm6.81e{-2}$	8.01 ± 0.26	32.47 ± 0.80	68.65 ± 1.68
	15 p	$0.41\pm1.10e{-2}$	$2.56\pm5.75e{-2}$	7.25 ± 0.18	31.97 ± 0.71	58.42 ± 0.89
E-ABRUPT (ABEBC)	1 p	$1.26e{-6}\pm7.77e{-7}$	$1.46e{-2}\pm6.62e{-3}$	$0.26\pm4.07e{-2}$	3.70 ± 0.10	14.97 ± 0.64
	5 p	$5.41e{-5}\pm9.61e{-5}$	$1.25e{-2}\pm9.70e{-3}$	$0.35\pm6.52e{-2}$	4.11 ± 0.14	23.23 ± 1.53
	15 p	$1.78e{-4}\pm3.36e{-4}$	$7.72e{-2}\pm2.04e{-2}$	$0.27\pm4.92e{-2}$	4.95 ± 0.32	16.42 ± 0.73
E-ABRUPT (ITEL)	1 p	$1.18\pm7.44e{-2}$	9.09 ± 0.40	26.89 ± 0.68	96.31 ± 1.48	146.99 ± 1.28
	5 p	1.29 ± 0.11	11.15 ± 0.49	32.23 ± 1.30	102.83 ± 2.15	184.99 ± 3.97
	15 p	$1.19\pm9.72e{-2}$	11.67 ± 0.31	29.95 ± 0.75	108.61 ± 3.28	166.94 ± 3.70
E-CHAOS (CME)	1 p	$20.82\pm5.63e{-2}$	48.11 ± 0.16	84.37 ± 0.46	188.78 ± 1.12	272.83 ± 1.35
	5 p	$13.06\pm4.66e{-2}$	42.42 ± 0.18	73.68 ± 0.40	155.13 ± 1.18	217.07 ± 2.75
	15 p	$9.86\pm4.74e{-2}$	35.32 ± 0.19	62.16 ± 0.25	133.06 ± 1.24	189.16 ± 2.00
E-CHAOS (ABEBC)	1 p	$1.71\pm2.16e{-2}$	26.78 ± 0.24	69.70 ± 0.48	181.77 ± 1.16	267.65 ± 1.34
	5 p	$1.53\pm2.55e{-2}$	25.67 ± 0.20	62.48 ± 0.37	149.57 ± 1.14	212.98 ± 2.69
	15 p	$1.40\pm2.24e{-2}$	21.52 ± 0.16	52.34 ± 0.25	128.34 ± 1.23	185.72 ± 1.99
E-CHAOS (ITEL)	1 p	$1.34\pm1.75e{-2}$	$4.93\pm8.87e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	5 p	$1.17\pm1.05e{-2}$	$4.93\pm8.82e{-3}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
	15 p	$1.05\pm7.03e{-3}$	$4.89\pm1.10e{-2}$	5.00 ± 0.00	5.00 ± 0.00	5.00 ± 0.00
E-PATTERN (CME)	1 p	$3.58\pm1.18e{-2}$	26.53 ± 0.21	95.09 ± 0.53	212.73 ± 0.75	290.58 ± 0.98
	5 p	$4.44\pm2.23e{-2}$	26.50 ± 0.23	78.80 ± 0.38	182.52 ± 0.72	268.98 ± 0.91
	15 p	$4.73\pm2.48e{-2}$	20.87 ± 0.20	61.80 ± 0.28	175.67 ± 0.67	236.75 ± 0.67

CME was lower than that of the QSO (5,50%) for environments with 5 or 10 dimensions, but not necessarily for the other multimodal environments. However, the final error of multi-swarm was either equal to or lower than that of the QSO (5,50%). Multi-swarm is designed to track multiple peaks concurrently. However, with a small peaks to sub-swarms ratio, the sub-swarms are likely to frequently compete for a peak, which causes the weakest sub-swarm to be re-initialised. For unimodal environments, only one sub-swarm was therefore allowed to exploit the peak. Consequently, the QSO (5,50%) performed better because its population size is greater than that of a sub-swarm.

For E-PROGRESS, the CME was higher than for E-STATIC. For the E-PROGRESS environments with 10 dimensions or less, the multi-swarm algorithm's performance level was inversely proportional to the number of peaks in the environment. The multi-swarm (5) performed significantly better than the multi-swarm (10) for most E-PROGRESS environments but multi-swarm (10) obtained a lower CME for the E-PROGRESS with 15 peaks and 10 dimensions or less. The multi-swarm (5) outperformed the re-evaluating PSO for most E-PROGRESS test cases. The re-evaluating PSO obtained a lower CME only for the unimodal environments with less than 10 dimensions. As for E-STATIC, the QSO (5,50%) obtained a lower CME than multi-swarm for the unimodal environments. However, the multi-swarm (5) outperformed QSO for all multimodal E-PROGRESS environments except for the 50-dimensional environment with 5 peaks.

For E-ABRUPT, the CME was higher than for E-STATIC but not necessarily higher than for E-PROGRESS. The multi-swarm (5) obtained a lower CME than the multi-swarm (10) for all E-ABRUPT environments except for the five-dimensional multimodal environments and the two-dimensional E-ABRUPT environment with 15 peaks. The multi-swarm (10) obtained a lower ABEBC for E-ABRUPT environments with 15 peaks and less than 50 dimensions. For the multi-swarm (10), a larger number of sub-swarms were able to converge on a peak between changes due to the lower peaks to sub-swarms ratio. The larger number of exploring particles of the multi-swarm (10) could occasionally improve the CME by landing on a peak immediately after a change. For all E-ABRUPT environments, the CME, ABEBC and the ITEL of multi-swarm were lower than those of the re-evaluating PSO. Multi-swarm generally performed better than the

QSO (5,50%) on problems with lower dimensionality and higher number of peaks while the QSO (5,50%) was more efficient for environments with less peaks and more dimensions.

For E-CHAOS, the CME was higher than for E-PROGRESS and E-ABRUPT and the performance level improved proportionally to the number of peaks. The multi-swarm (5) performed significantly better than the multi-swarm (10) for all E-CHAOS environments. The multi-swarm (5)'s CME was lower than that of the re-evaluating PSO for the environments with more than 10 dimensions, the 10-dimensional environment with 15 peaks, and the two-dimensional environments. However, the ABEBC and ITEL (where applicable) of multi-swarm were slightly lower than that of the re-evaluating PSO for all E-CHAOS environments. The fact that the sub-swarms quickly exploit the optima can explain the low ABEBC and ITEL. However, the fast convergence of the sub-swarms causes the particles to cluster on the peaks which in turn may cause a larger increase in error after a change. This explains why, for the environments with 5 or 10 dimensions, multi-swarm had a higher CME than the re-evaluating PSO but a lower ABEBC and ITEL. The multi-swarm (5) obtained a lower CME than the QSO (5,50%) for the two-dimensional E-CHAOS and for the environments with 15 peaks and 10 or 30 dimensions. However, multi-swarm only obtained a lower ABEBC for multimodal E-CHAOS environments with two dimensions.

For E-PATTERN, the CME was inversely proportional to the number of peaks for environments with 10 dimensions or more. The multi-swarm (5) performed significantly better than the multi-swarm (10) for the environments with less than 30 dimensions but the multi-swarm (10) obtained a lower CME than the multi-swarm (5) for environments with high dimensionality. Multi-swarm achieved a CME lower than that of the re-evaluating PSO for only the 30 and 50-dimensional environments, for the multimodal environments with 5 dimensions and for the two dimensional environment with 15 peaks. Multi-swarm also outperformed the QSO (5, 50%) for the environments with 30 or 50 dimensions and for the E-PATTERN environments with 15 peaks and two or five dimensions.

8.5.3 Summary of Strengths and Weaknesses

Multi-swarm was able to track peaks and detect new optima. The results indicate that the algorithm performed well for static multimodal environments but could not always obtain a final error as low as that of the standard PSO. For progressively and abruptly changing environments, multi-swarm was most efficient for multimodal environments with less than 50 dimensions in comparison to the re-evaluating PSO and QSO (5,50%). For chaotically changing environments, multi-swarm generally outperformed the re-evaluating PSO but was generally outperformed by the QSO (5,50%).

8.6 Self-adapting Multi-swarm

Like multi-swarm, SAMS uses swarm sub-division and parallel tracking of optima with the major difference that SAMS uses a mechanism to decide if a new sub-swarm should be introduced in the search space. This section presents and analyses the results obtained after applying the SAMS algorithm described in section 4.8.5 to the test environments. SAMS uses sub-swarms of 20 particles. The sub-swarms use QSO with a quantum radius of five, a quantum ratio of 50%, and use the parameters selected in section 6.4. The algorithm starts with a single sub-swarm in the search space and an n_{excess} of three. There is no upper limit to the number of sub-swarms that can be introduced. If all the sub-swarms have converged, the algorithm is always allowed to introduce an additional sub-swarm into the search space. This means that if more than five sub-swarms are introduced there will be more than 100 particles in the environment.

Even though the varying number of particles in the search space makes it difficult to conduct a fair comparison with the other algorithms, it is relevant to evaluate the ability of the algorithm to generate a sufficient number of sub-swarms to allow effective peak tracking and peak detection. It is also valuable to observe how variations in modality, dimensionality, change frequency and change severity affect the ability of the algorithm to select the appropriate number of sub-swarms to use.

8.6.1 Results

The results in table 8.23 are obtained by an implementation of algorithm 4.5 with the configuration described above. Because the varying number of sub-swarms determines the population size and consequently influences the computational cost of the algorithm, it is important to gather information about the number of sub-swarms generated. To this effect, the average number of sub-swarms and the maximum number of sub-swarms present concurrently in the search space are calculated. Table 8.24 lists the number of sub-swarms that were generated for the various environments. In this table, *Average number of swarms* refers to the number of sub-swarms present in the environment on average, calculated as

$$Average\ number\ of\ swarms = \frac{\sum_{m=1}^{n_m}\left(\frac{\sum_{t=1}^{n_t} C_{m,t}}{n_t}\right)}{n_m} \tag{8.1}$$

where n_m is the number of simulations, n_t is the maximum number of iterations and $C_{m,t}$ is the number of swarms in the environment at iteration t of simulation m. *Maximum swarm count* refers to the maximum number of sub-swarms present in the environment during a simulation, calculated as

$$Maximum\ swarm\ count = \frac{\sum_{m=1}^{M} C_m}{M} \tag{8.2}$$

where C_m is the maximum number of sub-swarms present concurrently in the search space during simulation m.

8.6.2 Analysis of Results

The p-values determining the significance of the difference between the results of the re-evaluating PSO and those of SAMS are listed in table C.26. The p-values determining the significance of the difference between the results of multi-swarm and those of SAMS are listed in table C.27.

Table 8.23: Experimental results of SAMS

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	3.99e−2±8.20e−3	0.45±5.88e−2	2.28±0.18	15.74±0.87	40.46±2.09
	5 p	0.11±6.35e−2	1.22±0.28	3.10±0.52	17.01±0.91	46.31±2.14
	15 p	0.27±0.13	1.95±0.45	3.39±0.57	21.22±1.62	45.62±2.15
E-STATIC (Error)	1 p	0.00±0.00	0.00±0.00	6.39e−15±4.44e−16	2.30e−4±1.60e−5	7.11e−3±4.94e−4
	5 p	0.00±0.00	0.00±0.00	5.16e−14±3.59e−15	1.90e−4±1.32e−5	6.20e−2±4.31e−3
	15 p	0.00±0.00	1.00e−10±6.96e−12	4.07e−13±2.83e−14	6.54e−3±4.54e−4	0.38±2.65e−2
E-PROGRESS (CME)	1 p	2.44±3.09e−2	4.02±6.88e−2	7.66±0.20	28.41±1.22	59.26±2.18
	5 p	4.03±9.47e−2	8.27±0.23	14.13±0.61	33.32±1.07	64.02±2.56
	15 p	4.00±7.02e−2	8.45±0.24	13.87±0.42	34.71±0.96	59.96±3.30
E-ABRUPT (CME)	1 p	0.43±1.95e−2	2.76±8.62e−2	7.38±0.23	31.98±2.07	72.02±3.76
	5 p	1.15±8.05e−2	8.65±0.45	20.87±1.20	54.79±2.27	93.26±2.85
	15 p	1.27±0.10	8.56±0.49	21.23±0.78	55.19±1.91	90.71±2.60
E-ABRUPT (ABEBC)	1 p	1.86e−7±7.14e−8	6.17e−4±2.48e−4	4.12e−2±6.42e−3	8.64±4.22	31.70±6.81
	5 p	6.10e−2±2.14e−2	4.58±1.40	15.61±5.52	31.87±5.24	55.19±5.65
	15 p	0.29±0.26	5.40±1.52	15.50±3.46	34.36±4.95	58.03±4.86
E-ABRUPT (ITEL)	1 p	1.89±0.17	11.89±0.33	25.97±0.61	82.94±6.54	179.15±10.88
	5 p	1.89±0.16	15.35±1.28	55.93±8.13	146.35±10.12	197.25±2.44
	15 p	1.94±0.31	19.20±3.97	51.22±7.50	161.49±9.67	195.59±2.98
E-CHAOS (CME)	1 p	25.65±1.64	54.71±0.96	93.54±0.73	196.25±2.03	280.74±3.59
	5 p	19.21±1.18	52.33±0.74	89.39±0.59	184.28±2.06	248.57±4.21
	15 p	14.48±0.90	43.56±0.73	75.28±0.66	156.92±1.88	219.39±5.46
E-CHAOS (ABEBC)	1 p	6.89±5.57	37.60±3.68	84.54±3.03	190.19±2.08	275.66±3.72
	5 p	11.05±5.58	36.52±3.30	81.59±2.80	178.32±2.17	244.47±4.53
	15 p	9.92±3.85	31.13±3.04	67.01±1.86	152.60±2.09	215.46±5.36
E-CHAOS (ITEL)	1 p	1.81±7.50e−2	4.96±1.16e−2	5.00±0.00	5.00±0.00	5.00±0.00
	5 p	1.86±7.64e−2	4.96±5.60e−3	5.00±0.00	5.00±0.00	5.00±0.00
	15 p	1.47±7.18e−2	4.96±8.73e−3	5.00±0.00	5.00±0.00	5.00±0.00
E-PATTERN (CME)	1 p	6.19±0.13	49.56±0.28	122.72±3.26	248.79±1.64	329.30±2.34
	5 p	14.86±0.22	41.96±0.35	96.98±0.81	211.34±1.39	304.48±2.30
	15 p	10.54±0.12	33.29±0.30	80.34±0.56	202.74±1.28	265.15±1.91

Table 8.24: Number of sub-swarms for SAMS

DIMENSIONS		2	5	10	30	50
E-STATIC (Average number of swarms)	1 p	1.99±3.24e−3	1.98±3.39e−3	1.96±2.52e−3	2.25±0.11	2.59±2.88e−2
	5 p	2.91±5.07e−2	4.51±9.67e−2	5.08±7.04e−2	3.31±0.22	3.15±0.15
	15 p	3.61±0.10	5.18±0.59	7.44±0.51	4.50±0.20	3.73±0.18
E-STATIC (Maximum swarm count)	1 p	2.00±0.00	2.00±0.00	2.00±0.00	2.77±0.16	3.43±0.19
	5 p	3.07±9.49e−2	5.00±0.00	6.00±0.00	4.43±0.36	4.93±0.26
	15 p	4.00±0.00	8.73±0.74	11.83±0.60	7.33±0.47	6.47±0.38
E-PROGRESS (Average number of swarms)	1 p	1.99±2.50e−3	1.98±2.55e−3	1.96±3.83e−3	2.30±0.10	2.60±2.57e−2
	5 p	2.94±2.47e−2	3.82±8.27e−2	4.43±0.14	3.43±0.15	2.82±8.27e−2
	15 p	2.94±2.99e−2	4.38±0.16	5.05±0.32	3.87±0.21	3.02±9.99e−2
E-PROGRESS (Maximum swarm count)	1 p	2.00±0.00	2.00±0.00	2.00±0.00	2.83±0.14	3.63±0.18
	5 p	4.23±0.16	5.40±0.19	5.73±0.17	5.00±0.24	4.43±0.21
	15 p	4.10±0.11	7.30±0.31	8.43±0.44	6.53±0.43	5.13±0.24
E-ABRUPT (Average number of swarms)	1 p	1.99±2.24e−3	1.97±2.81e−3	1.96±4.85e−3	2.37±0.12	2.62±2.44e−2
	5 p	3.01±2.48e−2	3.23±6.45e−2	3.42±7.15e−2	2.79±4.38e−2	2.62±2.97e−2
	15 p	3.02±4.41e−2	3.20±7.76e−2	3.40±6.66e−2	2.82±5.96e−2	2.65±2.64e−2
E-ABRUPT (Maximum swarm count)	1 p	2.00±0.00	2.00±0.00	2.00±0.00	2.77±0.16	3.23±0.16
	5 p	4.20±0.15	5.20±0.23	5.40±0.19	4.07±9.49e−2	4.00±0.00
	15 p	4.57±0.23	5.43±0.25	5.77±0.16	4.10±0.11	4.03±6.83e−2
E-CHAOS (Average number of swarms)	1 p	1.98±7.82e−3	1.91±2.32e−2	1.74±8.30e−2	1.46±0.13	1.23±0.11
	5 p	1.98±6.36e−2	1.77±7.71e−2	1.75±9.41e−2	1.40±0.12	1.41±0.12
	15 p	1.94±3.67e−2	1.79±4.96e−2	1.80±7.21e−2	1.52±0.12	1.24±0.10
E-CHAOS (Maximum swarm count)	1 p	2.00±0.00	2.00±0.00	2.00±0.00	1.73±0.17	1.50±0.19
	5 p	2.07±9.49e−2	2.00±0.00	1.97±6.83e−2	1.73±0.17	1.77±0.16
	15 p	2.07±9.49e−2	2.00±0.00	1.97±6.83e−2	1.87±0.13	1.57±0.19
E-PATTERN (Average number of swarms)	1 p	1.99±2.42e−3	1.97±7.99e−3	1.19±0.12	1.00±0.00	1.00±0.00
	5 p	2.86±3.56e−2	1.96±2.04e−2	1.38±0.14	1.00±0.00	1.00±0.00
	15 p	2.88±2.52e−2	2.01±6.34e−2	1.68±7.71e−2	1.00±0.00	1.00±0.00
E-PATTERN (Maximum swarm count)	1 p	2.00±0.00	2.00±0.00	1.37±0.18	1.00±0.00	1.00±0.00
	5 p	3.00±0.00	2.07±9.49e−2	1.60±0.19	1.00±0.00	1.00±0.00
	15 p	3.00±0.00	2.07±9.49e−2	2.00±0.00	1.00±0.00	1.00±0.00

For all test environments, the performance level decreased as dimensionality increased. SAMS was outperformed by multi-swarm for all environments with regard to CME.

For E-STATIC, the CME increased with the number of dimensions and generally decreased with the number of peaks. Compared to the standard PSO, SAMS's CME was lower for the environments with 30 and 50 dimensions but higher for the two-dimensional environments. The number of sub-swarms generated was proportional to the number of peaks in the environment. It was shown that the capacity of QSO to converge quickly on a peak gives QSO an advantage over the standard PSO for environments of high dimensionality. The lower performance of SAMS for the two dimensional environments can be explained by the lower number of particles present in the environment in comparison to the population size of the standard PSO. For the unimodal environments, the original sub-swarm could converge quickly within the dynamic convergence radius, which triggered the introduction of a second sub-swarm. Because only one peak was present, only one of the sub-swarms could converge around the peak. The other sub-swarm was reinitialised as soon as the sub-swarms entered each other's dynamic convergence radius, which prevented the introduction of an extra sub-swarm. However, for the unimodal E-STATIC environment with 30 and 50-dimensions, the results show that more than two swarms were introduced. This indicates that the first two sub-swarms converged before locating the optimum. For the multimodal E-STATIC environments, the number of sub-swarms introduced varied with the dimensionality. SAMS generated most sub-swarms for the 10-dimensional environments. For multimodal environments, both the average number of sub-swarms and the maximum number of sub-swarms were generally lower than the number of peaks in the environment. However, for the 10-dimensional E-STATIC environment with five peaks, both the average number of swarms and the maximum swarm count were greater than five. The reason why more sub-swarms were introduced for the 10-dimensional multimodal environments is not obvious. However, the number of sub-swarms that are introduced in the environment is influenced by the ability of all sub-swarms to converge simultaneously on different peaks. Aside from the modality of the environment, the ability of the sub-swarms to converge simultaneously is influenced by three factors: the number of iterations needed by a sub-swarm to converge, the size of the convergence radius, and the likelihood of the sub-swarms to be reinitialised by the exclusion operator of SAMS. It has been shown that the dimensionality influences the number of iterations needed for a swarm to converge. Also, while the size of the quantum radius remains constant, it can be seen from equation (4.5) that the size of the dynamic convergence radius increases with the number of dimensions. Since the dynamic convergence radius also defines the exclusion radius, a high dimensionality can cause the sub-swarms to be often reinitialised and thereby prevent the introduction of new sub-swarms.

For E-PROGRESS, the CME was higher than for E-STATIC. The performance was poorer than that of the re-evaluating PSO for all but the multimodal two-dimensional E-PROGRESS environments and the five-dimensional E-PROGRESS environment with 15 peaks. The number of sub-swarms in the search space was lower for the multimodal E-PROGRESS environments than for E-STATIC environments except for the five peaks environments with two and 30 dimensions. This can be explained by the fact that sub-swarms take longer to converge when the peaks are moving. Except for the 10-dimensional E-PROGRESS environment with 15 peaks, less than five sub-swarms were present on average in the environment. This means that most of the time, less than 100 particles were were present in the search space and that SAMS was therefore having a smaller population that the other algorithms evaluated.

For E-ABRUPT, the CME was higher than for E-STATIC. The CME was lower than for E-PROGRESS for the two dimensional environments and for the unimodal environment with five or 10 dimensions. For these environments and for the unimodal environment with 30 dimensions, SAMS outperformed the re-evaluating PSO. The ABEBC of SAMS was lower than that of the re-evaluating PSO for the multimodal environments with two dimensions. The ITEL was however lower than that of the re-evaluating PSO for all environments. For the unimodal environments, SAMS generated about the same number of sub-swarms for E-ABRUPT and E-PROGRESS environments. For the two dimensional multimodal environments, SAMS created more sub-swarms for the E-ABRUPT environments than for the E-PROGRESS environments. However, for the other multimodal environments, more sub-swarms were created for the E-PROGRESS environments.

For E-CHAOS, the CME was higher than for E-PROGRESS and E-ABRUPT. The CME was inversely proportional to the number of peaks in the environments. The ABEBC was lower than the CME showing that optimisation took place while the environment stayed static. The re-evaluating PSO outperformed SAMS for all E-CHAOS environments. For E-CHAOS, few sub-swarms were created. The frequent and severe changes made it difficult for the sub-swarms to converge within the dynamic convergence radius which prevented the introduction of new sub-swarms. This indicates that when a swarm sub-division approach is used, the mechanism that generates sub-swarms may be disrupted by frequent and severe changes.

For E-PATTERN, the CME was inversely proportional to the number of peaks for all but the two-dimensional environments. The re-evaluating PSO outperformed SAMS for all environments. For E-PATTERN, the average number of sub-swarms in the search space was lower that for E-CHAOS for environments with 10 or more dimensions. For E-PATTERN environments with 30 or 50 dimensions, no extra sub-swarms were introduced showing that the frequent and severe changes of these environments prevented the original sub-swarm from converging.

8.6.3 Summary of Strengths and Weaknesses

SAMS was able to track peaks and detect the appearance of new optima. SAMS did not perform as well as multi-swarm but a fair comparison was made difficult due to the varying number of particles. More sub-swarms were generally introduced for static environments than for DEs. Also, less sub-swarms were introduced on average for chaotically changing environments than for progressively or abruptly changing environments. In comparison with the standard PSO, SAMS tended to perform better for static environments with high dimensionality. For progressively changing environments, the performance of SAMS tended to be poorer than that of the re-evaluating PSO. For abruptly changing environments, SAMS performed better for problems of low dimensionality compared to the re-evaluating PSO with regard to CME and ABEBC. However, SAMS showed a better reactivity than the re-evaluating PSO. For chaotically changing environments, the sub-swarms struggled to converge making it difficult for SAMS to introduce new sub-swarms. SAMS is therefore not suitable for chaotically changing environments.

8.7 General Comparison

As mentioned previously, none of the algorithms evaluated had been optimised for any of the environments. Therefore the results presented here cannot be used to make any definitive conclusion about the superiority of an algorithm for a specific environment type. However, a number of observations can be derived from the experiments and suggestions can be made about which approach should be considered or avoided for the various types of environment.

Tables 8.25 to 8.29 provide rankings of the algorithms for each environment. In these tables, *StPSO* refers to the standard PSO, *RiPSO* refers to the reinitialising PSO, *Multi 5* refers to multi-swarm with five sub-swarms of 20 particles, and *Multi 10* refers to multi-swarm with 10 sub-swarms of 10 particles. The APSO, QSO and SAMS are referred to using the acronyms previously defined for them. For the reinitialising PSO, the percentage refers to the reinitialisation ratio. For the APSO, the percentage refers to the charged ratio. For QSO, the percentage refers to the quantum ratio and R refers to the size of the quantum radius. The algorithms are listed in a top-down manner from best to worst for each performance measure. Only the best configurations of each of the algorithms were considered for the ranking. Table 8.30 summarises the algorithms that performed best for each test environment and each measure used.

For E-STATIC environments, all algorithms performed well. The standard PSO, re-evaluating PSO and reinitialising PSO behaved identically and gave similar results. The best results were achieved by QSO and multi-swarm as can be seen in tables 8.25 and 8.30. QSO had the lowest CME for the unimodal environments. A small quantum radius was best for problems with low dimensionality, while a large quantum radius obtained lower CME for the environments with high dimensionality. Multi-swarm gave good results for the multimodal environments. However, QSO with quantum radius 50 was the best for all environments with 50 dimensions. With reference to the final error, the standard PSO and reinitialising PSO are listed as best in table 8.30 for the environments with 10 or less dimensions, but a number of other algorithms achieved an equal error level (0.00) as can be seen in tables 7.1 to 8.23. Table 8.25 lists those algorithms that achieved a final error of 0.00 with '(0)' next to their name to indicate that they share the first place in the ranking. For the environments of high dimensionality, QSO obtained the lowest final error while multi-swarm and SAMS tended to rank lower than the other algorithms. The results indicate that, even for static environments, the modified PSO algorithms can outperform the standard PSO. The ability of QSO with a small radius to exploit quickly has shown to be very effective for unimodal environments as these environments do not have local optima where the QSO's swarm could become trapped. However, for multimodal environments the multi-swarm's capacity to exploit multiple peaks concurrently is preferable. The reason why QSO with a large radius had a lower CME for environments with a high dimensionality may be the ability of that algorithm to quickly locate the optimum with quantum particles that are exploring a large portion of the search space. However, QSO with a small radius generally obtained a lower final error. The approaches taken by the modified PSO algorithms to overcome diversity loss did not prevent these algorithms from performing well for static environments. Detection and response is meaningless in static environments because no change can be detected. However, diversity maintenance and swarm sub-division have shown to be beneficial with regards to CME and final error.

Table 8.26 provides the ranking for E-PROGRESS environmnents. For E-PROGRESS environments, the standard, re-evaluating and reinitialising PSO were outperformed by the algorithms that maintain diversity at all time. As shown in table 8.30, the algorithms that obtained the lowest CME for E-STATIC environments (QSO and multi-swarm) were also those that performed best for E-PROGRESS environments. However, for E-PROGRESS environments, multi-swarm was best for all multimodal environments with less than 50 dimensions and the multi-swarm (10) outperformed the multi-swarm

Table 8.25: Ranking for E-STATIC environments

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	QSO - 70%, R: 5 Multi 5 RePSO APSO - 10% StPSO RiPSO - 10% SAMS	QSO - 50%, R: 5 Multi 5 RePSO APSO - 10% SAMS RiPSO - 70% StPSO	QSO - 50%, R: 5 Multi 5 APSO - 30% StPSO SAMS RePSO RiPSO - 30%	QSO - 50%, R: 50 Multi 5 SAMS RePSO RiPSO - 30% APSO - 30% StPSO	QSO - 50%, R: 50 Multi 5 SAMS RiPSO - 50% APSO - 10% RePSO StPSO
	5 p	QSO - 90%, R: 5 Multi 5 APSO - 30% RePSO StPSO RiPSO - 10% SAMS	Multi 5 SAMS StPSO APSO - 30% QSO - 10%, R: 50 RiPSO - 50% RePSO	Multi 5 QSO - 10%, R: 5 APSO - 10% StPSO RiPSO - 30% RePSO SAMS	QSO - 50%, R: 5 Multi 5 SAMS RePSO RiPSO - 70% APSO - 30% StPSO	QSO - 50%, R: 50 Multi 5 SAMS RiPSO - 50% APSO - 10% RePSO StPSO
	15 p	Multi 5 QSO - 10%, R: 5 APSO - 30% StPSO RePSO RiPSO - 50% SAMS	Multi 5 RePSO APSO - 10% RiPSO - 70% StPSO QSO - 10%, R: 5 SAMS	Multi 10 APSO - 10% RiPSO - 90% RePSO SAMS StPSO QSO - 10%, R: 50	Multi 5 QSO - 30%, R: 50 SAMS RiPSO - 90% RePSO APSO - 10% StPSO	QSO - 30%, R: 50 Multi 5 SAMS APSO - 10% RePSO RiPSO - 50% StPSO
E-STATIC (Error)	1 p	APSO - 10% (0) Multi 5 (0) QSO - 10%, R: 5 (0) RePSO (0) RiPSO - 10% (0) SAMS (0) StPSO (0)	APSO - 10% (0) Multi 5 (0) QSO - 10%, R: 5 (0) RePSO (0) RiPSO - 10% (0) SAMS (0) StPSO (0)	APSO - 10% (0) QSO - 10%, R: 5 (0) RePSO (0) RiPSO - 10% (0) StPSO (0) Multi 5 SAMS	QSO - 10%, R: 5 StPSO RiPSO - 70% RePSO APSO - 10% Multi 5 SAMS	QSO - 10%, R: 5 StPSO RiPSO - 90% RePSO APSO - 30% SAMS Multi 5
	5 p	APSO - 10% (0) Multi 5 (0) QSO - 10%, R: 5 (0) RePSO (0) RiPSO - 10% (0) SAMS (0) StPSO (0)	APSO - 10% (0) Multi 5 (0) QSO - 10%, R: 5 (0) RiPSO - 10% (0) SAMS (0) StPSO (0) RePSO	APSO - 10% (0) QSO - 10%, R: 5 (0) RePSO (0) RiPSO - 10% (0) StPSO (0) Multi 5 SAMS	QSO - 10%, R: 5 RiPSO - 90% SAMS Multi 5 APSO - 10% RePSO StPSO	QSO - 10%, R: 5 RiPSO - 10% APSO - 30% RePSO StPSO Multi 5 SAMS
	15 p	APSO - 10% (0) Multi 5 (0) QSO - 10%, R: 5 (0) RePSO (0) RiPSO - 10% (0) SAMS (0) StPSO (0)	APSO - 10% (0) Multi 5 (0) QSO - 10%, R: 5 (0) RePSO (0) RiPSO - 10% (0) StPSO (0) SAMS	RiPSO - 10% (0) QSO - 10%, R: 5 SAMS Multi 5 StPSO RePSO APSO - 10%	QSO - 30%, R: 5 RiPSO - 50% APSO - 10% SAMS Multi 10 StPSO RePSO	QSO - 30%, R: 50 APSO - 50% Multi 5 RiPSO - 50% StPSO RePSO SAMS

(5) for most environments with 15 peaks. The reinitialising PSO (with 10% reinitialisation) ranked well for low dimensional environments but showed to perform poorly for E-PROGRESS environments of higher dimensionality. The re-evaluating PSO, APSO, QSO, multi-swarm and SAMS were able to track peaks, which is sufficient to perform well for unimodal environments that change progressively. However, the capacity of multi-swarm and QSO with a large radius to detect appearing peaks made them preferable for the progressively changing multimodal environments.

For progressively changing environments, a detection and response strategy can only be effective for environments with low dimensionality. For environments with more dimensions, the response is triggered too often and too much diversity is generated to allow exploitation. Maintaining diversity at all times allows peak tracking and peak detection, and is effective for progressively changing environments. The higher the level of diversity is maintained, the better the peak detection capacity of the algorithm will be. Swarm sub-division and parallel tracking of optima has shown to be a particularly effective strategy for multimodal environments that change progressively. A sub-swarm is often exploiting the local optimum before it becomes the global optimum which eliminates the need to detect the new global optimum.

Table 8.26: Ranking for E-PROGRESS environments

DIMENSIONS		2	5	10	30	50
E-PROGRESS (CME)	1 p	QSO - 30%, R: 5	QSO - 30%, R: 5	QSO - 50%, R: 5	QSO - 50%, R: 5	QSO - 50%, R: 50
		APSO - 10%	APSO - 10%	Multi 5	Multi 5	Multi 5
		RePSO	RePSO	APSO - 10%	RePSO	APSO - 10%
		RiPSO - 10%	Multi 5	RePSO	APSO - 10%	RePSO
		Multi 5	SAMS	SAMS	SAMS	SAMS
		SAMS	RiPSO - 10%	RiPSO - 10%	StPSO	RiPSO - 10%
		StPSO	StPSO	StPSO	RiPSO - 10%	StPSO
	5 p	Multi 5	Multi 5	Multi 5	Multi 5	QSO - 50%, R: 50
		RiPSO - 10%	RiPSO - 10%	RePSO	QSO - 30%, R: 5	Multi 5
		QSO - 50%, R: 50	QSO - 10%, R: 50	APSO - 50%	RePSO	RePSO
		APSO - 70%	RePSO	QSO - 10%, R: 50	APSO - 10%	APSO - 30%
		SAMS	APSO - 10%	SAMS	SAMS	SAMS
		RePSO	SAMS	RiPSO - 10%	StPSO	StPSO
		StPSO	StPSO	StPSO	RiPSO - 10%	RiPSO - 10%
	15 p	Multi 10	Multi 10	Multi 10	Multi 5	QSO - 30%, R: 50
		RiPSO - 10%	APSO - 90%	RePSO	QSO - 50%, R: 5	Multi 5
		APSO - 90%	SAMS	APSO - 10%	APSO - 30%	APSO - 10%
		QSO - 50%, R: 50	RiPSO - 10%	SAMS	RePSO	RePSO
		SAMS	QSO - 30%, R: 50	QSO - 10%, R: 50	SAMS	SAMS
		RePSO	RePSO	RiPSO - 10%	StPSO	StPSO
		StPSO	StPSO	StPSO	RiPSO - 10%	RiPSO - 10%

Table 8.27 provides the ranking for E-ABRUPT environments. For E-ABRUPT environments, the best performance for each environment was obtained by either the reinitialising PSO, QSO, or multi-swarm. However, the algorithms were ranked differently depending on the performance measure used. Table 8.30 shows that the E-ABRUPT environments where QSO (with a quantum radius of five) obtained the lowest CME are grouped on the top right, i.e. these environments tend to have a lower number of peaks and higher number of dimensions. The re-initialising PSO had the lowest CME

for the bottom left environment, which has the lowest dimensionality and highest number of peaks. The rest of the E-ABRUPT environments, where multi-swarm obtained the lowest CME, are also grouped together between the top right and the bottom left. The ABEBC measurements indicate which algorithms obtained the lowest error before change. The reinitialising PSO had the lowest ABEBC for the environments with two dimensions. For the rest of the E-ABRUPT environments, QSO with quantum radius five had the lowest ABEBC for the unimodal environments and multi-swarm had the lowest ABEBC for the multimodal environments (except for the environment with five peaks and 10 dimensions for which the reinitialising PSO was better). Algorithms that obtained the lowest ABEBC (first rank in table 8.27) often also obtained a low ITEL. Indeed, with regards to the ITEL, the reinitialising PSO was the best for two out of three environments with two dimensions, QSO was the best for four out of five unimodal environments, and multi-swarm was the best for seven out of eight multimodal environments with more than two dimensions. The best algorithm to use for an abruptly changing environment depends on the intent of the practitioner: if the aim is to maintain the lowest error on average, the algorithms that obtained the best CME should be used. If the aim is to obtain the lowest error before a change occurs, the algorithms with the best ABEBC should be used. If the aim is to reach an acceptable error level after a change as quickly as possible, the algorithms that obtained the best ITEL should be used.

The reinitialising PSO generally ranked well for environments of low dimensionality with regard to all performance measures. This indicates that detection and response is an effective strategy for abruptly changing environments but that this strategy does not scale well. The diversity maintenance strategy of QSO showed good scalability and was most effective for unimodal environments. However, the APSO generally ranked lower for high dimensionality environments than for environments with less dimensions. This contrast between the performance of QSO and the APSO shows that the effectiveness of a diversity maintenance strategy in abruptly changing environments is dependent on the implementation. The ranking of multi-swarm showed that swarm sub-division performed generally well, particularly for multimodal environments where having multiple sub-swarms increases the chances of finding the global optimum between two changes.

Table 8.27: Ranking for E-ABRUPT environments

DIMENSIONS		2	5	10	30	50
E-ABRUPT (CME)	1 p	Multi 5 RiPSO - 90% QSO - 30%, R: 50 APSO - 70% SAMS RePSO StPSO	Multi 5 QSO - 30%, R: 50 APSO - 70% RiPSO - 50% SAMS RePSO StPSO	QSO - 50%, R: 5 Multi 5 APSO - 50% SAMS RiPSO - 50% RePSO StPSO	QSO - 50%, R: 5 Multi 5 SAMS RePSO APSO - 10% RiPSO - 10% StPSO	QSO - 50%, R: 5 Multi 5 RePSO RiPSO - 10% APSO - 10% SAMS StPSO
	5 p	Multi 5 RiPSO - 70% APSO - 70% QSO - 50%, R: 50 SAMS RePSO StPSO	Multi 10 RiPSO - 50% APSO - 70% QSO - 10%, R: 50 RePSO SAMS StPSO	Multi 5 RiPSO - 90% APSO - 70% QSO - 10%, R: 5 RePSO SAMS StPSO	QSO - 50%, R: 5 Multi 5 APSO - 10% RiPSO - 10% RePSO SAMS StPSO	QSO - 30%, R: 5 Multi 5 RePSO APSO - 50% RiPSO - 30% SAMS StPSO
	15 p	RiPSO - 90% Multi 10 APSO - 50% QSO - 10%, R: 50 SAMS RePSO StPSO	Multi 10 RiPSO - 90% APSO - 70% QSO - 10%, R: 50 RePSO SAMS StPSO	Multi 5 RiPSO - 90% QSO - 10%, R: 5 APSO - 50% RePSO SAMS StPSO	Multi 5 QSO - 30%, R: 5 RePSO RiPSO - 30% APSO - 90% SAMS StPSO	QSO - 30%, R: 5 Multi 5 APSO - 10% RePSO RiPSO - 10% SAMS StPSO
E-ABRUPT (ABEBC)	1 p	RiPSO - 50% QSO - 10%, R: 5 APSO - 30% RePSO Multi 5 SAMS StPSO	QSO - 10%, R: 5 RiPSO - 10% APSO - 10% RePSO Multi 5 SAMS StPSO	QSO - 30%, R: 5 APSO - 10% RiPSO - 30% RePSO SAMS Multi 5 StPSO	QSO - 30%, R: 5 Multi 5 RePSO RiPSO - 10% APSO - 10% SAMS StPSO	QSO - 50%, R: 5 Multi 5 RePSO RiPSO - 10% APSO - 10% SAMS StPSO
	5 p	RiPSO - 90% Multi 5 APSO - 30% QSO - 10%, R: 50 SAMS RePSO StPSO	RiPSO - 30% Multi 5 APSO - 70% QSO - 10%, R: 50 RePSO SAMS StPSO	Multi 5 RiPSO - 90% RePSO APSO - 70% QSO - 10%, R: 5 SAMS StPSO	Multi 5 QSO - 50%, R: 5 RiPSO - 10% APSO - 10% RePSO SAMS StPSO	Multi 5 QSO - 30%, R: 5 RiPSO - 30% RePSO APSO - 30% SAMS StPSO
	15 p	RiPSO - 50% Multi 10 APSO - 50% QSO - 10%, R: 50 SAMS RePSO StPSO	Multi 10 RiPSO - 90% APSO - 70% RePSO QSO - 10%, R: 50 SAMS StPSO	Multi 10 RiPSO - 90% APSO - 10% RePSO QSO - 10%, R: 5 SAMS StPSO	Multi 10 QSO - 90%, R: 5 RiPSO - 90% RePSO APSO - 10% SAMS StPSO	Multi 5 QSO - 30%, R: 5 RePSO RiPSO - 30% APSO - 10% SAMS StPSO
E-ABRUPT (ITEL)	1 p	QSO - 90%, R: 50 RiPSO - 90% APSO - 90% Multi 5 SAMS RePSO StPSO	Multi 5 QSO - 30%, R: 5 APSO - 90% SAMS RiPSO - 50% RePSO StPSO	QSO - 50%, R: 5 Multi 5 SAMS APSO - 50% RiPSO - 30% RePSO StPSO	QSO - 70%, R: 5 Multi 5 SAMS RePSO APSO - 10% RiPSO - 10% StPSO	QSO - 70%, R: 5 Multi 5 SAMS RePSO APSO - 10% RiPSO - 10% StPSO
	5 p	RiPSO - 70% Multi 10 APSO - 90% QSO - 50%, R: 50 SAMS RePSO StPSO	Multi 5 QSO - 10%, R: 5 APSO - 90% RiPSO - 50% SAMS RePSO StPSO	Multi 5 SAMS RiPSO - 90% APSO - 30% RePSO QSO - 10%, R: 5 StPSO	Multi 5 QSO - 50%, R: 5 SAMS RePSO RiPSO - 10% APSO - 10% StPSO	Multi 5 QSO - 30%, R: 5 SAMS APSO - 10% RePSO RiPSO - 10% StPSO
	15 p	RiPSO - 90% APSO - 90% Multi 10 QSO - 50%, R: 50 SAMS RePSO StPSO	Multi 5 RiPSO - 50% APSO - 70% SAMS RePSO QSO - 10%, R: 50 StPSO	Multi 10 SAMS RiPSO - 90% QSO - 10%, R: 5 APSO - 70% RePSO StPSO	Multi 5 QSO - 90%, R: 5 SAMS RiPSO - 30% RePSO APSO - 30% StPSO	QSO - 90%, R: 5 Multi 5 SAMS APSO - 10% RePSO RiPSO - 10% StPSO

Table 8.28 provides the ranking for E-CHAOS environments. For E-CHAOS environments, QSO with a quantum radius of 50 and a high quantum ratio obtained the lowest CME for all environments, except for those with two dimensions where multi-swarm performed better. QSO also obtained the best ABEBC for most environments. However, the QSO (5, 30%) was the best for the 10-dimensional E-CHAOS environments and the unimodal environment with two dimensions. Multi-swarm obtained the lowest ABEBC for the multimodal environments with two dimensions. QSO obtained the lowest ITEL for most E-CHAOS environments with 10 dimensions or less. None of the algorithms evaluated could reach an error of 10 or less at any point for E-CHAOS environments with 30 or 50 dimensions, and the QSO was the only algorithm that could occasionally reach the error threshold for the 10-dimensional environments. The reinitialising PSO obtained the lowest ITEL for the multimodal E-CHAOS environments with two dimensions. For E-CHAOS environments, the algorithms have obtained poorer results than for E-PROGRESS and E-ABRUPT environments. Also, the algorithms tended to obtain better results for those E-CHAOS environments that contain more peaks, because a higher number of peaks increases the chance of a particle being randomly located on a good position. The fact that the QSO algorithms with the largest quantum radius and the largest quantum ratio performed best for E-CHAOS environments of high dimensionality indicates that the random repositioning of a large portion of the swarm within a large quantum radius helped the performance. In E-CHAOS environments, the peaks are repositioned to a distance of 50 after a change, and this distance is equal to the size of the large quantum radius. If after a change, the global optimum remained located on the same peak, the location of new optimum was likely to be within the quantum radius. QSO therefore had the best chance of detecting the new optimum if a large number of quantum particles were patrolling within the quantum radius. In other words, QSO with a large radius was able to *track* the optimum in a chaotically changing environment. However, having more quantum particles to explore the search space implies having less neutral particles to exploit the optimum, which explains why QSO with a small radius and lower quantum ratio obtained a better ABEBC and ITEL than QSO with large radius for the E-CHAOS environments of lower dimensionality. SAMS was ranked in second last place (only above the standard PSO) for all E-CHAOS environments due to the limitations of the sub-swarm generation mechanism. From the ranking of the algorithms, it appears that diversity maintenance is generally more effective than swarm sub-division, itself more effective than detection and response. However, in chaotically changing environments the particles rarely have the time to converge between two changes and diversity loss is therefore less of an issue than for the other types of environment. Also, the effect of frequent and severe changes on the sub-swarm generation mechanisms must be taken into account if a swarm sub-division approach is used.

Table 8.29 provides the ranking for E-PATTERN environments. For E-PATTERN environments, QSO obtained the lowest CME for all environments except for the two-dimensional environment with 15 peaks where multi-swarm was better. QSO with a small quantum radius was best for the E-PATTERN environments with two or five dimensions, but a quantum radius of 50 was preferable for environments with 10 dimensions or more. This is consistent with the fact that QSO with a high quantum radius performs well for chaotically changing environments of high dimensionality. It also indicates that a small quantum radius is sufficient to track peaks for E-PATTERN environments with two and five dimensions but that a larger quantum radius is necessary to track peaks in E-PATTERN environments of higher dimensionality where the change severity is greater. The APSO ranked in third place for most environment. This, and the superior ranking of QSO indicate that diversity maintenance was the most effective approach for the E-PATTERN environments. SAMS ranked second last for all environments, only ahead of the standard PSO. This highlights once again the limitation of the sub-swarm generation mechanism. However, multi-swarm obtained a good ranking on average for the multimodal environments. The reinitialising PSO ranked third last for most environments. This confirms that a detection and response strategy is not suitable for environments with a high change frequency. None of the modified PSO algorithms have shown to take advantage of the pattern in the change.

Table 8.30 shows that QSO and multi-swarm performed best for most environments. QSO uses diversity maintenance. Multi-swarm uses swarm sub-division and parallel tracking of optima. Either of these approaches have shown to be most effective depending on the nature of the environment and the intent of the practitioner. The detection and

Table 8.28: Ranking for E-CHAOS environments

DIMENSIONS		2	5	10	30	50
E-CHAOS (CME)	1 p	Multi 5 QSO - 30%, R: 5 RiPSO - 70% APSO - 10% RePSO SAMS StPSO	QSO - 90%, R: 50 APSO - 10% RePSO RiPSO - 10% Multi 5 SAMS StPSO	QSO - 90%, R: 50 RePSO APSO - 70% Multi 5 RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 30% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 50% RePSO RiPSO - 10% SAMS StPSO
	5 p	Multi 5 QSO - 30%, R: 5 RiPSO - 90% APSO - 30% RePSO SAMS StPSO	QSO - 70%, R: 50 APSO - 50% RePSO RiPSO - 10% Multi 5 SAMS StPSO	QSO - 70%, R: 50 APSO - 50% RePSO Multi 5 RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 RiPSO - 10% APSO - 50% RePSO SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 70% RePSO RiPSO - 10% SAMS StPSO
	15 p	Multi 5 QSO - 30%, R: 5 RiPSO - 90% APSO - 10% RePSO SAMS StPSO	QSO - 70%, R: 50 RiPSO - 30% RePSO APSO - 50% Multi 5 SAMS StPSO	QSO - 50%, R: 50 Multi 5 APSO - 30% RiPSO - 10% RePSO SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 30% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 30% RePSO RiPSO - 10% SAMS StPSO
E-CHAOS (ABEBC)	1 p	QSO - 50%, R: 5 Multi 5 RiPSO - 70% APSO - 10% RePSO SAMS StPSO	QSO - 30%, R: 5 Multi 5 RePSO APSO - 10% RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 70% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 30% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 50% RePSO RiPSO - 10% SAMS StPSO
	5 p	Multi 5 QSO - 10%, R: 5 RiPSO - 90% RePSO APSO - 10% SAMS StPSO	QSO - 30%, R: 5 Multi 5 APSO - 90% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 50% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 RiPSO - 10% APSO - 50% RePSO SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 70% RePSO RiPSO - 10% SAMS StPSO
	15 p	Multi 5 QSO - 10%, R: 5 RiPSO - 90% APSO - 10% RePSO SAMS StPSO	QSO - 30%, R: 5 Multi 5 RiPSO - 30% RePSO APSO - 70% SAMS StPSO	QSO - 70%, R: 50 Multi 5 APSO - 50% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 30% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 Multi 5 APSO - 30% RePSO RiPSO - 10% SAMS StPSO
E-CHAOS (ITEL)	1 p	QSO - 90%, R: 50 RiPSO - 90% Multi 5 APSO - 90% RePSO SAMS StPSO	QSO - 30%, R: 5 Multi 5 APSO - 50% RePSO SAMS RiPSO - 10% StPSO	QSO - 30%, R: 5 — — — — — —	— — — — — — —	— — — — — — —
	5 p	RiPSO - 90% Multi 5 QSO - 10%, R: 5 APSO - 90% RePSO SAMS StPSO	QSO - 30%, R: 5 Multi 5 APSO - 90% RePSO RiPSO - 10% SAMS StPSO	QSO - 90%, R: 50 — — — — — —	— — — — — — —	— — — — — — —
	15 p	RiPSO - 70% Multi 5 QSO - 10%, R: 5 APSO - 10% RePSO SAMS StPSO	QSO - 30%, R: 5 Multi 5 APSO - 10% RePSO RiPSO - 30% SAMS StPSO	QSO - 70%, R: 5 — — — — — —	— — — — — — —	— — — — — — —

Table 8.29: Ranking for E-PATTERN environments

DIMENSIONS		2	5	10	30	50
E-PATTERN (CME)	1 p	QSO - 10%, R: 5	QSO - 30%, R: 5	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
		RePSO	RePSO	RiPSO - 10%	Multi 10	Multi 10
		APSO - 10%	APSO - 30%	APSO - 30%	APSO - 70%	RePSO
		RiPSO - 10%	Multi 5	RePSO	RePSO	APSO - 10%
		Multi 5	RiPSO - 10%	Multi 5	RiPSO - 10%	RiPSO - 10%
		SAMS	SAMS	SAMS	SAMS	SAMS
		StPSO	StPSO	StPSO	StPSO	StPSO
	5 p	QSO - 10%, R: 5	QSO - 10%, R: 5	QSO - 70%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
		RePSO	Multi 5	RiPSO - 10%	Multi 10	Multi 10
		APSO - 10%	RePSO	APSO - 30%	APSO - 10%	APSO - 90%
		Multi 5	APSO - 10%	RePSO	RePSO	RePSO
		RiPSO - 10%	RiPSO - 10%	Multi 5	RiPSO - 10%	RiPSO - 10%
		SAMS	SAMS	SAMS	SAMS	SAMS
		StPSO	StPSO	StPSO	StPSO	StPSO
	15 p	Multi 5	QSO - 10%, R: 5	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
		RiPSO - 10%	Multi 5	APSO - 10%	Multi 10	Multi 10
		APSO - 70%	RePSO	Multi 5	RiPSO - 30%	APSO - 70%
		QSO - 50%, R: 50	APSO - 10%	RePSO	APSO - 70%	RePSO
		RePSO	RiPSO - 10%	RiPSO - 10%	RePSO	RiPSO - 30%
		SAMS	SAMS	SAMS	SAMS	SAMS
		StPSO	StPSO	StPSO	StPSO	StPSO

response approach used by the re-initialising PSO has also shown to be effective for specific problems. The results for the 50 dimensional environments indicate that, among the algorithms evaluated, QSO with a large radius scaled best with regards to average error (CME).

8.8 Summary

This chapter provided the results of applying a selected panel of algorithms designed to optimise dynamic problems to a representative range of DE test cases. These results were analysed and the experimental procedure used allowed to highlight the strengths and weaknesses of the various algorithms with regards to the various test environments. Using several performance measures for the abruptly and chaotically changing environments provided more insight on the behaviour and performance of the algorithms. Strengths and weaknesses were identified for each of the algorithms and a general comparison between the algorithms was also conducted. This comparison indicated that QSO, multi-swarm and (to a lesser degree) the reinitialising PSO were generally the most effective

Table 8.30: Lowest error per environment

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	QSO - 70%, R: 5	QSO - 50%, R: 5	QSO - 50%, R: 5	QSO - 50%, R: 50	QSO - 50%, R: 50
	5 p	QSO - 90%, R: 5	Multi 5	Multi 5	QSO - 50%, R: 5	QSO - 50%, R: 50
	15 p	Multi 5	Multi 5	Multi 10	Multi 5	QSO - 30%, R: 50
E-STATIC (Error)	1 p	StPSO	StPSO	StPSO	QSO - 10%, R: 5	QSO - 10%, R: 5
	5 p	StPSO	StPSO	StPSO	QSO - 10%, R: 5	QSO - 10%, R: 5
	15 p	StPSO	StPSO	RiPSO - 10%	QSO - 30%, R: 5	QSO - 30%, R: 50
E-PROGRESS (CME)	1 p	QSO - 30%, R: 5	QSO - 30%, R: 5	QSO - 50%, R: 5	QSO - 50%, R: 5	QSO - 50%, R: 50
	5 p	Multi 5	Multi 5	Multi 5	Multi 5	QSO - 50%, R: 50
	15 p	Multi 10	Multi 10	Multi 10	Multi 5	QSO - 30%, R: 50
E-ABRUPT (CME)	1 p	Multi 5	Multi 5	QSO - 50%, R: 5	QSO - 50%, R: 5	QSO - 50%, R: 5
	5 p	Multi 5	Multi 10	Multi 5	QSO - 50%, R: 5	QSO - 30%, R: 5
	15 p	RiPSO - 90%	Multi 10	Multi 5	Multi 5	QSO - 30%, R: 5
E-ABRUPT (ABEBC)	1 p	RiPSO - 50%	QSO - 10%, R: 5	QSO - 30%, R: 5	QSO - 30%, R: 5	QSO - 50%, R: 5
	5 p	RiPSO - 90%	RiPSO - 30%	Multi 5	Multi 5	Multi 5
	15 p	RiPSO - 50%	Multi 10	Multi 10	Multi 10	Multi 5
E-ABRUPT (ITEL)	1 p	QSO - 90%, R: 50	Multi 5	QSO - 50%, R: 5	QSO - 70%, R: 5	QSO - 70%, R: 5
	5 p	RiPSO - 70%	Multi 5	Multi 5	Multi 5	Multi 5
	15 p	RiPSO - 90%	Multi 5	Multi 10	Multi 5	QSO - 90%, R: 5
E-CHAOS (CME)	1 p	Multi 5	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
	5 p	Multi 5	QSO - 70%, R: 50	QSO - 70%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
	15 p	Multi 5	QSO - 70%, R: 50	QSO - 50%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
E-CHAOS (ABEBC)	1 p	QSO - 50%, R: 5	QSO - 30%, R: 5	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
	5 p	Multi 5	QSO - 30%, R: 5	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
	15 p	Multi 5	QSO - 30%, R: 5	QSO - 70%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
E-CHAOS (ITEL)	1 p	QSO - 90%, R: 50	QSO - 30%, R: 5	QSO - 30%, R: 5	—	—
	5 p	RiPSO - 90%	QSO - 30%, R: 5	QSO - 90%, R: 50	—	—
	15 p	RiPSO - 70%	QSO - 30%, R: 5	QSO - 70%, R: 5	—	—
E-PATTERN (CME)	1 p	QSO - 10%, R: 5	QSO - 30%, R: 5	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
	5 p	QSO - 10%, R: 5	QSO - 10%, R: 5	QSO - 70%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50
	15 p	Multi 5	QSO - 10%, R: 5	QSO - 90%, R: 50	QSO - 90%, R: 50	QSO - 90%, R: 50

algorithms. The experiments showed that for a given DE, the best approach to take to overcome diversity loss depends on the dynamic behaviour, the dimensionality, and the modality of the environment as well as on the intent of the practitioner. Through the comparison of the performance of the algorithms, the suggestion of the most suitable approach(es) for each environment type was made possible.

Chapter 9

Conclusions and Future Work

"A conclusion is the place where you got tired thinking."

– Martin Henry Fischer

This chapter summarises the major contributions of this work and the findings presented in the experimental part. Related topics of research that can be investigated in the future are also suggested.

9.1 Conclusions

This study aimed at providing a better understanding of dynamically changing environments, and of the effectiveness of the various approaches that can be taken to adapt the original PSO algorithm for DEs. In reaching this goal, the contributions summarised in this section were made.

The research work started with the theoretical treatment of dynamic optimisation problems. After describing the characteristics of DEs and discussing the various existing

DE classification systems, a characterisation system based on a combination of Eberhart *et al.*'s classification, Angeline's classification and novel behavioural classes was proposed.

In spite of showing a natural capacity to adapt to minor modifications of the environment, the standard PSO algorithm is limited by outdated memory and diversity loss when applied to DEs. It was shown that parameter selection can only partially overcome these limitations and that the original PSO algorithm must be modified in order to perform effectively in DEs. Various approaches to modify the standard PSO algorithm were discussed, namely, re-evaluation of the *pbest*, detection and response, diversity maintenance, and swarm sub-division and parallel tracking of optima. Then, a representative panel of algorithms that make use of these approaches were described in more detail, namely, the reinitialising PSO, the CPSO, the APSO, QSO, multi-swarm and SAMS.

To evaluate the effectiveness of swarm algorithms applied to DEs, appropriate performance measures had to be used. Therefore, existing performance measures were critically examined and additional measures were proposed.

An experimental procedure was designed to evaluate the performance of the selected algorithms on a range of dynamically changing environments. The environment test cases included a range of static environments, progressively changing environments, abruptly changing environments, chaotically changing environments and circular environments. Each test case was instantiated for a range of modality and dimensionality.

To have a performance benchmark against which the results of the algorithms designed for DEs could be measured, the standard PSO and a PSO that re-evaluates the *pbest* of the particles were evaluated. The standard PSO showed to perform well for static environments but struggled for changing environments. However, a certain capacity to take advantage of a pattern in the change was observed. The re-evaluating PSO outperformed the standard PSO for all DE test cases. The re-evaluating PSO was able to track peaks and the diversity of the swarm was observed to increase after a change. However, the algorithm struggled to detect the appearance of new peaks in progressively changing environments, and was prone to be trapped in local optima in abruptly changing environments.

An evaluation of the modified PSO algorithms was then conducted. Strengths and weaknesses could be identified for each algorithm and numerous observations were made regarding the behaviour of the algorithms for the various environments. The reinitialising PSO showed to be particularly effective for abruptly changing environments of low dimensionality but to scale badly and to perform poorly for environments with a high change frequency. The APSO could detect and track peaks in progressively and abruptly changing environments but the charged particles had little effect on the performance for chaotically changing environments. QSO showed to be generally very effective. With a small quantum radius, QSO generally provided the best results for unimodal problems. QSO with a large radius generally provided the best results for environments with high dimensionality. Multi-swarm also proved to be very effective, particularly for multimodal problems. SAMS's performance were worst than that of multi-swarm but SAMS was able to outperform the re-evaluating PSO for some abruptly changing environments while using a smaller number of particles. SAMS performed poorly for chaotically changing environments as it struggled to introduce new sub-swarms.

A general comparison of the algorithms showed that the most effective approach to take to overcome diversity loss depends on the dimensionality, modality and type of the DEs, as well as on the intend of the practitioner. Detection and response is most effective in abruptly changing environments but does not scale well. Diversity maintenance scales best and is particularly effective for unimodal environments. Swarm sub-division and parallel tracking of optima generally provides the best results for multimodal environments. However, frequent and severe changes may interfere with the mechanism that manages the formation of sub-swarms.

9.2 Future Work

Narrowing the scope and selecting algorithms, algorithm configurations, parameter values, problem types and problem configurations out of a larger set has been a recurrent exercise during the making of this work. This thesis therefore leaves a large number of paths to be explored. This section suggests further research that could be conducted on the topic of PSO algorithms in DEs.

9.2.1 Alternative Types of Problems

All problems used in this study were continuous, nonlinear, uni-objective, unconstrained problems. Integer and discrete problems, linear and quadratic problems, multi-objective problems, and constrained problems can be the object of separate research. Additionally, even though various types of DEs have been used in the experiments, more DE test cases can be designed and used to refine the evaluation the PSO algorithms. In particular, the effect of a gradual increase in temporal and/or spatial severity could be studied further.

9.2.2 Alternative Approaches and Algorithms

This study has considered the various approaches used to overcome diversity loss by evaluating a representative panel of algorithms. However, a growing number of algorithms are being developed to optimise dynamic problems and only a fraction of these were included in this study. The evaluation of additional algorithms on the different types of DEs can only provide a greater understanding of PSO in DEs.

9.2.3 Algorithms Designed to Find Stable Solutions

The algorithms evaluated in this work only focus on finding accurate solutions. Algorithms that look for solutions that remain good after a change could be the object of a separate study.

9.2.4 Algorithms That Predict Changes

From the results, only the standard PSO showed a (limited) capacity to take advantage of the presence of a pattern in the changes to improve performance. By keeping track of the previous location of the optimum, algorithms could predict the future position of the optimum. Such algorithms could be studied and/or designed.

9.2.5 Alternative Parameter Settings

Only one configuration for the standard PSO was used in the experiments and it is likely that using a different configuration would influence the performance and the behaviour of the algorithms designed for DEs. The effect that the various parameters of the standard PSO have on the modified PSOs evaluated could therefore be investigated.

9.2.6 Optimise Algorithms per Environment Type

The parameter selection for the algorithms evaluated in the experiments could be optimised independently for each type of DE. Such optimisation would provide additional information about the strengths and weaknesses of the various algorithms for the various environment types.

9.2.7 Variance of QSO

QSO has showed to be one of the most effective algorithms for most environment test cases, sometimes outperforming multi-swarm in the experiments. The size of the quantum radius showed to influence the behaviour of the swarm and additional sizes for the quantum radius could be investigated. It could also be interesting to investigate variations of the QSO algorithm. For instance, a QSO with two types of quantum particles could be designed. The first type of quantum particles would remain within a small quantum radius to exploit around the *gbest* and the second type of quantum particles would be dispersed within a large quantum radius to be able to detect new optima. Alternatively, for abruptly changing environments, the size of the quantum radius could vary. After a change, the size of the quantum radius would be set to a high value to allow peak detection. This value could be then progressively reduced to promote exploitation around the *gbest*.

9.2.8 Effect of Change Severity on Diversity

The experiments have showed that the frequency and severity of the changes influence the diversity of the swarms where the particles have their *pbest* re-evaluated. Frequent changes have shown to maintain a certain level of diversity in the swarm. Also, after a change, the diversity level seems to increase more if the peak where the particles had converged has moved to a distant location than if a peak can be found close by. Further research could investigate exactly how spatial and temporal change severity influence the diversity of the various algorithms.

Bibliography

[1] P. Angeline. Tracking extrema in dynamic environments. In Peter Angeline *et al.*, editor, *Evolutionary Programming VI*, volume 1213 of *Lecture Notes in Computer Science*, pages 335–345. Springer Berlin / Heidelberg, 1997.

[2] T. Bäck. On the behavior of evolutionary algorithms in dynamic environments. In *Proceedings of the IEEE World Congress on Computational Intelligence*, pages 446–451, May 1998.

[3] S. Bird and X. Li. Adaptively choosing niching parameters in a PSO. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 3–10, New York, NY, USA, 2006. ACM.

[4] S. Bird and X. Li. Enhancing the robustness of a speciation-based PSO. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 843–850, 2006.

[5] S. Bird and X. Li. Informative performance metrics for dynamic optimisation problems. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pages 18–25, New York, NY, USA, 2007. ACM.

[6] T.M. Blackwell. Swarms in dynamic environments. In Erick Cant-Paz *et al.*, editor, *Genetic and Evolutionary Computation*, volume 2723 of *Lecture Notes in Computer Science*, pages 200–212. Springer Berlin / Heidelberg, 2003.

[7] T.M. Blackwell. Particle swarms and population diversity. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 9:793–802, 2005.

[8] T.M. Blackwell. Particle swarm optimization in dynamic environments. In Shengxiang Yang *et al.*, editor, *Evolutionary Computation in Dynamic and Uncertain Environments*, volume 51 of *Studies in Computational Intelligence*, pages 29–49. Springer Berlin / Heidelberg, 2007.

[9] T.M. Blackwell and P.J. Bentley. Don't push me! Collision-avoiding swarms. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1691–1696, 2002.

[10] T.M. Blackwell and P.J. Bentley. Dynamic search with charged swarms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 19–26, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[11] T.M. Blackwell and J. Branke. Multi-swarm optimization in dynamic environments. In Günther Raidl *et al.*, editor, *Applications of Evolutionary Computing*, volume 3005 of *Lecture Notes in Computer Science*, pages 489–500. Springer Berlin / Heidelberg, 2004.

[12] T.M Blackwell and J. Branke. Multiswarms, exclusion, and anti-convergence in dynamic environments. *IEEE Transactions on Evolutionary Computation*, 10(4):459–472, August 2006.

[13] J. Branke. Evolutionary approaches to dynamic optimization problems - a survey. In *Proceedings of the Genetic and Evolutionary Computation Conference Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pages 134–137, 1999.

[14] J. Branke. Memory enhanced evolutionary algorithms for changing optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1875–1882, 1999.

[15] J. Branke. *Evolutionary Optimization in Dynamic Environments.* Kluwer Academic Publishers, Norwell, MA, USA, 2001.

[16] J. Branke, E. Salihoğlu, and Ş. Uyar. Towards an analysis of dynamic environments. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1433–1440, New York, NY, USA, 2005. ACM.

[17] R. Brits. Niching Strategies for Particle Swarm Optimization. Master's thesis, University of Pretoria, Pretoria, South Africa, 2002.

[18] R. Brits, A. P. Engelbrecht, and F. Van den Bergh. A niching particle swarm optimizer. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning*, pages 692–696, 2002.

[19] A.J. Carlisle. *Applying the Particle Swarm Optimizer to Non-stationary Enviornments*. PhD thesis, Auburn University, Auburn, AL, USA, 2002.

[20] A.J. Carlisle and G. Dozier. Adapting particle swarm optimization to dynamic environments. *International Conference on Artificial Intelligence*, 1:429–434, 2000.

[21] A.J. Carlisle and G. Dozler. Tracking changing extrema with adaptive particle swarm optimizer. In *Proceedings of the 5th Biannual World Automation Congress*, volume 13, pages 265–270, 2002.

[22] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1951–1957, 1999.

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

[24] H.G. Cobb. An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments. *Naval Research Lab Memorandum Report*, 6760:523–529, 1990.

[25] K. De Jong. Evolving in a changing world. In Zbigniew Ras and Andrzej Skowron, editors, *Foundations of Intelligent Systems*, volume 1609 of *Lecture Notes in Computer Science*, pages 512–519. Springer Berlin / Heidelberg, 1999.

[26] R.C. Eberhart and Y. Shi. Comparing inertia weights and constriction factors in particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 1, pages 84–88, 2000.

[27] R.C. Eberhart and Y. Shi. Tracking and optimizing dynamic systems with particle swarms. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 1, pages 94–100, 2001.

[28] R.C. Eberhart, P. Simpson, and R. Dobbins. *Computational Intelligence PC Tools*. Academic Press Professional, Inc. San Diego, CA, USA, 1996.

[29] A.P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. Wiley Publishing, 2005.

[30] A.P. Engelbrecht. *Computational Intelligence: An Introduction*. Wiley Publishing, 2nd edition, 2007.

[31] A.P. Engelbrecht, B.S. Masiye, and G. Pampard. Niching ability of basic particle swarm optimization algorithms. In *Proceedings the IEEE Symposium on Swarm Intelligence*, pages 397–400, June 2005.

[32] J.L. Fernandez-Marquez and J.L. Arcos. Keeping diversity when exploring dynamic environments. In *Proceedings of the ACM Symposium on Applied Computing*, pages 1192–1196, New York, NY, USA, 2009. ACM.

[33] T.C. Fogarty, F. Vavak, and P. Cheng. Use of the genetic algorithm for load balancing of sugar beet presses. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 617–624, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[34] A. Gasper and P. Collard. From GAs to artificial immune systems: improving adaptation in time dependent optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1866–1872, 1999.

[35] J.J. Grefenstette. Genetic algorithms for changing environments. *Parallel Problem Solving from Nature*, 2:137–144, 1992.

[36] J.J. Grefenstette. Evolvability in dynamic fitness landscapes: a genetic algorithm approach. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, 1999.

[37] M. Guntsch, M. Middendorf, and H. Schmeck. An ant colony optimization approach to dynamic TSP. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 860–867. Citeseer, 2001.

[38] A. Hashemi and M. Meybodi. Cellular PSO: a PSO for dynamic environments. In Zhihua Cai *et al.*, editor, *Advances in Computation and Intelligence*, volume 5821 of *Lecture Notes in Computer Science*, pages 422–433. Springer Berlin / Heidelberg, 2009.

[39] R. J. W. Hodgson. Particle swarm optimization applied to the atomic cluster optimization problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 68–73, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[40] X. Hu and R.C. Eberhart. Tracking dynamic systems with PSO: where's the cheese. In *Proceedings of the Workshop on Particle Swarm Optimization*, pages 80–83, Indianapolis, IN, USA, 2001.

[41] X. Hu and R.C. Eberhart. Adaptive particle swarm optimization: detection and response to dynamic systems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1666–1670, 2002.

[42] A. Ismail and A.P. Engelbrecht. Global optimization algorithms for training product unit neural networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 1, page 1132, Los Alamitos, CA, USA, 2000. IEEE, INNS, ENNS, IEEE Computer Society.

[43] S. Janson and M. Middendorf. A hierarchical particle swarm optimizer for dynamic optimization problems. In Günther Raidl *et al.*, editor, *Applications of Evolutionary Computing*, volume 3005 of *Lecture Notes in Computer Science*, pages 513–524. Springer Berlin / Heidelberg, 2004.

[44] C.L. Karr. Genetic algorithms and fuzzy logic for adaptive process control. *Intelligent Hybrid Systems*, pages 63–83, 1995.

[45] C.L. Karr. Adaptive process control using biological paradigms. *Journal of Network and Computer Applications*, 19(1):21–44, 1996.

[46] J. Kennedy. The particle swarm: social adaptation of knowledge. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 303–308, April 1997.

[47] J. Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1931–1938, 1999.

[48] J. Kennedy. Stereotyping: improving particle swarm performance with cluster analysis. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1507–1512, 2000.

[49] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

[50] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1671–1676, 2002.

[51] T. Krink, J.S. Vesterstrøm, and J. Riget. Particle swarm optimisation with spatial particle extension. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1474–1479, 2002.

[52] C. Li and S. Yang. Fast multi-swarm optimization for dynamic optimization problems. In *Proceedings of the 4th International Conference on Natural Computation*, volume 7, pages 624–628, Los Alamitos, CA, USA, October 2008. IEEE Computer Society.

[53] C. Li and S. Yang. A generalized approach to construct benchmark problems for dynamic optimization. *Simulated Evolution and Learning*, pages 391–400, 2008.

[54] C. Li and S. Yang. A clustering particle swarm optimizer for dynamic optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 439–446, May 2009.

[55] C. Li, S. Yang, T.T. Nguyen, E.L. Yu, X. Yao, Y. Jin, H.G. Beyer, and P.N. Suganthan. Benchmark generator for CEC2009 competition on dynamic optimization. *University of Leicester, University of Birmingham, Nanyang Technological University, Tech. Rep*, 2008.

[56] X. Li. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In Kalyanmoy Deb, editor, *Genetic and Evolutionary Computation*, volume 3102 of *Lecture Notes in Computer Science*, pages 105–116. Springer Berlin / Heidelberg, 2004.

[57] X. Li, J. Branke, and T.M. Blackwell. Particle swarm with speciation and adaptation in a dynamic environment. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 51–58, New York, NY, USA, 2006. ACM.

[58] X. Li and K.H. Dam. Comparing particle swarms for tracking extrema in dynamic environments. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1772–1779, December 2003.

[59] R.I. Lung and D. Dumitrescu. A collaborative model for tracking optima in dynamic environments. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 564–567, September 2007.

[60] H.B. Mann and D.R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.

[61] M. Matsumoto and T. Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, 1998.

[62] R.W. Morrison. Performance measurement in dynamic environments. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 99–102, 2003.

[63] R.W. Morrison. *Designing Evolutionary Algorithms for Dynamic Environments.* Springer-Verlag New York Inc, 2004.

[64] L. Mussi, S. Cagnoni, and F. Daolio. Empirical assessment of the effects of update synchronization in particle swarm optimization. In *Proceedings the International Conference of the Italian Association for Artificial Intelligence Workshop on Complexity, Evolution and Emergent Intelligence*, December 2009.

[65] S. Naka, T. Genji, T. Yura, and Y. Fukuyama. Practical distribution state estimation using hybrid particle swarm optimization. In *Proceedings of the IEEE Power Engineering Society Transmission and Distribution Conference*, volume 2, pages 815–820, 2001.

[66] S. Naka, T. Genji, T. Yura, and Y. Fukuyama. A hybrid particle swarm optimization for distribution state estimation. *IEEE Transactions on Power Systems*, 18(1):60–68, February 2003.

[67] O. Olorunda and A.P. Engelbrecht. Measuring exploration/exploitation in particle swarms using swarm diversity. In *Proceedings of the IEEE World Congress on Computational Intelligence Evolutionary Computation*, pages 1128–1134, 2008.

[68] E. Ozcan and C.K. Mohan. Analysis of a simple particle swarm optimization system. *Intelligent Engineering Systems Through Artificial Neural Networks*, 8:253–258, 1998.

[69] E. Ozcan and C.K. Mohan. Particle swarm optimization: surfing the waves. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1939–1944, 1999.

[70] D. Parrott and X. Li. A particle swarm model for tracking multiple peaks in a dynamic environment using speciation. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 1, pages 98–103, June 2004.

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

[72] K.E. Parsopoulos and M.N. Vrahatis. Modification of the particle swarm optimizer for locating all the global minima. In *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms*, pages 324–327, Prague, Czech Republic, 2001. Springer Verlag Wien.

[73] K.E. Parsopoulos and M.N. Vrahatis. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 1(2):235–306, 2002.

[74] E.S. Peer. A serendipitous software framework for facilitating collaboration in computational intelligence. Master's thesis, University of Pretoria, Pretoria, South Africa, 2004.

[75] J. Peng, Y. Chen, and R.C. Eberhart. Battery pack state of charge estimator design using computational intelligence approaches. In *Proceedings of the 15th Annual Battery Conference on Applications and Advances*, pages 173–177, Long Beach, CA, USA, 2000.

[76] T. Peram, K. Veeramachaneni, and C.K. Mohan. Fitness-distance-ratio based particle swarm optimization. In *Proceedings of the IEEE Symposium on Swarm Intelligence*, pages 174–181, April 2003.

[77] C.L. Ramsey and J.J. Grefenstette. Case-based initialization of genetic algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 84–91, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[78] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice hall, 3rd edition, 2010.

[79] R. Salomon. 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, 1996.

[80] I.L. Schoeman and A. P. Engelbrecht. Containing particles inside niches when optimizing multimodal functions. In *Proceedings of the Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, pages 78–85, South Africa, 2005. South African Institute for Computer Scientists and Information Technologists.

[81] I.L. Schoeman and A.P. Engelbrecht. Using vector operations to identify niches for particle swarm optimization. In *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, volume 1, pages 361–366, December 2004.

[82] I.L. Schoeman and A.P. Engelbrecht. A parallel vector-based particle swarm optimizer. In Bernardete Ribeiro *et al.*, editor, *Adaptive and Natural Computing Algorithms*, pages 268–271. Springer Vienna, 2005.

[83] I.L. Schoeman and A.P. Engelbrecht. Niching for dynamic environments using particle swarm optimization. In Tzai-Der Wang *et al.*, editor, *Simulated Evolution and Learning*, volume 4247 of *Lecture Notes in Computer Science*, pages 134–141. Springer Berlin / Heidelberg, 2006.

[84] L. Schoofs and B. Naudts. Swarm intelligence on the binary constraint satisfaction problem. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1444–1449, 2002.

[85] E. Schrödinger. An undulatory theory of the mechanics of atoms and molecules. *Physical Review*, 28:1049–1070, 1926.

[86] Y. Shi and R.C. Eberhart. A modified particle swarm optimizer. In *Proceedings of the IEEE World Congress on Computational Intelligence Evolutionary Computation*, pages 69–73, May 1998.

[87] Y. Shi and R.C. Eberhart. Parameter selection in particle swarm optimization. In Porto *et al.* V., editor, *Evolutionary Programming VII*, volume 1447 of *Lecture Notes in Computer Science*, pages 591–600. Springer Berlin / Heidelberg, 1998.

[88] Y. Shi and R.C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1875–1882, 1999.

[89] Y. Shi and R.C. Eberhart. Fuzzy adaptive particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 1, pages 101–106, 2001.

[90] I.C. Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317–325, 2003.

[91] K. Trojanowski and Z. Michalewicz. Searching for optima in non-stationary environments. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 1843–1850, 1999.

[92] F. Van den Bergh. *An Analysis of Particle Swarm Optimizers.* PhD thesis, University of Pretoria, Pretoria, South Africa, 2001.

[93] F. Van den Bergh and A.P. Engelbrecht. Effects of swarm size on cooperative particle swarm optimisers. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 892–899, San Francisco, CA, USA, July 2001.

[94] F. Van den Bergh and A.P. Engelbrecht. A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8):937–971, 2006.

[95] F. Vavak, T. Fogarty, and K. Jukes. A genetic algorithm with variable range of local search for tracking changing environments. In Hans-Michael Voigt *et al.*, editor, *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, volume 1141, pages 376–385. Springer Berlin / Heidelberg, 1996.

[96] F. Vavak, K. Jukes, and T.C. Fogarty. Adaptive balancing of bank of sugar-beet presses using a genetic algorithm with variable local search range, 3rd intl.

In *Proceedings of the Mendel Conference on Genetic Algorithms*, pages 164–169, 1997.

[97] F. Vavak, K. Jukes, and T.C. Fogarty. Adaptive combustion balancing in multiple burner boiler using a genetic algorithm with variable range of local search. In *Proceedings of the 7th International Conference on Genetic Algorithms*, volume 1, pages 719–726, East Lansing, MI, USA, 1997.

[98] G. Venter and J. Sobieszczanski-Sobieski. Particle swarm optimization. *AIAA Journal*, 41(8):1583–1589, 2003.

[99] G. Venter and J. Sobieszczanski-Sobieski. Multidisciplinary optimization of a transport aircraft wing using particle swarm optimization. *Structural and Multidisciplinary Optimization*, 26(1):121–131, 2004.

[100] J.S. Vesterstrøm, J. Riget, and T. Krink. Division of labor in particle swarm optimisation. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 2, pages 1570–1575, 2002.

[101] K. Weicker. An analysis of dynamic severity and population size. In Marc Schoenauer *et al.*, editor, *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, volume 1917, pages 159–168. Springer Berlin / Heidelberg, 2000.

[102] K. Weicker. Performance measures for dynamic environments. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, pages 64–76, London, UK, UK, 2002. Springer-Verlag.

[103] K. Weicker and N. Weicker. On evolution strategy optimization in dynamic environments. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 3, pages 2046–2052, 1999.

[104] Y. Zheng, L. Ma, L. Zhang, and J. Qian. Empirical study of particle swarm optimizer with an increasing inertia weight. In *Proceedings of the IEEE Congress on Evolutionary Computation*, volume 1, pages 221–226, December 2003.

[105] Y.L. Zheng, L.H. Ma, L.Y. Zhang, and J.X. Qian. On the convergence analysis and parameter selection in particle swarm optimization. In *Proceedings of the International Conference on Machine Learning and Cybernetics*, volume 3, pages 1802–1807, November 2003.

Appendix A

Function Rotation

This appendix provides the pseudocode for creating a rotation matrix, MResult. Any point in the search space can be multiplied by MResult to obtain the fitness at that point after the base function has been rotated. Algorithm A.1 is adapted from [79] to provide a function rotation by a constant angle of *alpha* radians.

```
Variables of type Matrix [N,N] of type double
ME, /* diagonal matrix, i.e. ME[i,i]=1, ME[i,j]=0 */
MTurn, /* misc matrix */
MResult, /* result */
alpha; /* angle of rotation in radians */
Functions
MulMatrix( M1, M2 )
    IS
        M1 := M1 * M2;
    END MulMatrix:
MRot( MTurn, i, j ) /* create a single rotation matrix */
    IS
        MTurn:=ME; /* set MTurn to the matrix ME */
        MTurn[ i, i ] := cos( alpha );
        MTurn[ j, j ] := cos( alpha );
        MTurn[ i, j ] := sin( alpha );
        MTurn[ j, i ] := -sin( alpha );
    ENDMRot;
CreateMatrix( MResult, N ) /* main routine for matrix creation */
    IS
        MResult := ME; /* initialize MResult */
        FOR i:=2 STEP 1 UNTIL N
        DO
            MRot( MTurn, 1, i );
            MulMatrix( MResult, MTurn )
        END FOR;
        FOR i:=2 STEP 1 UNTIL N-1
        DO
            MRot( MTurn, i, N );
            MulMatrix( MResult, MTurn )
        END FOR;
    END CreateMatrix;
```

Algorithm A.1: Pseudocode for rotation matrix MResult

Appendix B

CIlib XML Configurations

This appendix provides an XML configuration for CIlib version 0.7.4 which illustrates how simulations used in chapters 7 and 8 were generated. Because of space constraints, only a representative sample of the simulations have been included.

A CIlib XML file is divided into four sections:

- Algorithm definitions, where each algorithm configuration is defined separately. The XML sample shows the standard PSO ("stdpso" ID tag), the re-evaluating PSO ("reevpso" ID tag), the reinitialising PSO with reinitialisation of 10% ("reinitpso_10" ID tag), the APSO with 10% of charged particles ("charged_10"), the QSO with a quantum radius of five and quantum ratio of 10% ("quatum_r5_10" ID tag), the multi-swarm with five sub-swarms of 20 particles ("multiswarm_5" ID tag), and SAMS ("selfmultiswarm" ID tag).
- Problem definitions, where each problem configuration is defined separately. The XML sample shows the test environment with one peak and two dimensions, namely, E-STATIC ("static_p1_d2" ID tag), E-PROGRESS ("progress_p1_d2" ID tag), E-ABRUPT ("abrupt_p1_d2" ID tag), E-CHAOS ("chaos_p1_d2" ID tag), and E-PATTERN ("pattern_p1_d2" ID tag).

- Measurement definitions, where the measurement suites are defined. The XML sample shows the error measurement (measure_error ID tag) that was used to calculate the CME and final error in E-STATIC. A separate suite of measurements was used for E-ABRUPT (measure_abrupt ID tag) and E-CHAOS (measure_chaos ID tag) environments in order to calculate the ABEBC and ITEL.

- Simulation definitions, where a separate simulation is defined for each combination of algorithm and environment test case. The XML sample shows a simulation where the re-evaluating PSO is applied to the unimodal E-PROGRESS with two dimensions.

The sample of XML configuration follows.

```
<?xml version=“1.0”?>
<!DOCTYPE simulator [
<!ATTLIST algorithm id ID #IMPLIED>
<!ATTLIST problem id ID #IMPLIED>
<!ATTLIST measurements id ID #IMPLIED>
]>
<simulator>
<algorithms>
    <algorithm id=“stdpso” class=“pso.PSO”¿
        <initialisationStrategy class=“algorithm.initialisation.ClonedPopulationInitialisationStrategy” entityNumber=“100”¿
            <entityType class=“pso.particle.StandardParticle” /¿
        </initialisationStrategy¿
        <addStoppingCondition class=“stoppingcondition.MaximumIterations” maximumIterations=“1000” /¿
        <topology class=“entity.topologies.VonNeumannTopology” /¿
    </algorithm¿
    <algorithm id=“reevpso” class=“pso.PSO”¿
        <initialisationStrategy class=“algorithm.initialisation.ClonedPopulationInitialisationStrategy” entityNumber=“100”¿
            <entityType class=“pso.dynamic.DynamicParticle”/¿
        </initialisationStrategy¿
        <iterationStrategy class=“pso.dynamic.DynamicIterationStrategy”¿
            <detectionStrategy class=“pso.dynamic.detectionstrategies.AlwaysTrueDetectionStrategy” /¿
            <responseStrategy class=“pso.dynamic.responsestrategies.ParticleReevaluationResponseStrategy” /¿
        </iterationStrategy¿
        <addStoppingCondition class=“stoppingcondition.MaximumIterations” maximumIterations=“1000” /¿
        <topology class=“entity.topologies.VonNeumannTopology” /¿
    </algorithm¿
    <algorithm id=“reinitpso_10” class=“pso.PSO”>
        <initialisationStrategy class=“algorithm.initialisation.ClonedPopulationInitialisationStrategy” entityNumber=“100”>
            <entityType class=“pso.dynamic.DynamicParticle”/>
        </initialisationStrategy>
        <iterationStrategy class=“pso.dynamic.DynamicIterationStrategy”>
            <detectionStrategy class=“pso.dynamic.detectionstrategies.PeriodicChangeDetectionStrategy” period=“5” />
            <responseStrategy class=“pso.dynamic.responsestrategies.PartialReinitialisationResponseStrategy”>
                <reinitialisationRatio value=“0.1” />
            </responseStrategy>
        </iterationStrategy>
        <addStoppingCondition class=“stoppingcondition.MaximumIterations” maximumIterations=“1000” />
        <topology class=“entity.topologies.VonNeumannTopology” />
    </algorithm>
```

```
<algorithm id="charged_10" class="pso.PSO">
    <topology class="entity.topologies.VonNeumannTopology"/>
    <addStoppingCondition class="stoppingcondition.MaximumIterations" maximumIterations="1000" />
    <initialisationStrategy class="algorithm.initialisation.ChargedPopulationInitialisationStrategy" entityNumber="100">
        <chargeMagnitude value="16" />
        <chargedRatio value="0.1" />
        <entityType class="pso.dynamic.ChargedParticle">
            <velocityUpdateStrategy class="pso.dynamic.ChargedVelocityUpdateStrategy">
                <pCore value="1" />
                <p value="30" />
            </velocityUpdateStrategy>
        </entityType>
    </initialisationStrategy>
    <iterationStrategy class="pso.dynamic.DynamicIterationStrategy">
        <detectionStrategy class="pso.dynamic.detectionstrategies.AlwaysTrueDetectionStrategy" />
        <responseStrategy class="pso.dynamic.responsestrategies.ParticleReevaluationResponseStrategy" />
    </iterationStrategy>
</algorithm>
<algorithm id="quantum_r5_10" class="pso.PSO">
    <topology class="entity.topologies.VonNeumannTopology"/>
    <addStoppingCondition class="stoppingcondition.MaximumIterations" maximumIterations="1000" />
    <initialisationStrategy class="algorithm.initialisation.ChargedPopulationInitialisationStrategy" entityNumber="100">
        <chargeMagnitude value="1" />
        <chargedRatio value="0.1" />
        <entityType class="pso.dynamic.ChargedParticle">
            <velocityUpdateStrategy class="pso.dynamic.QuantumVelocityUpdateStrategy" />
            <positionUpdateStrategy class="pso.dynamic.QuantumPositionUpdateStrategy">
                <radius value="5" />
            </positionUpdateStrategy>
        </entityType>
    </initialisationStrategy>
    <iterationStrategy class="pso.dynamic.DynamicIterationStrategy">
        <detectionStrategy class="pso.dynamic.detectionstrategies.AlwaysTrueDetectionStrategy" />
        <responseStrategy class="pso.dynamic.responsestrategies.ParticleReevaluationResponseStrategy"/>
    </iterationStrategy>
</algorithm>
<algorithm id="multi_quantum_20" class="pso.PSO">
    <topology class="entity.topologies.VonNeumannTopology"/>
    <addStoppingCondition class="stoppingcondition.MaximumIterations" maximumIterations="1000" />
    <initialisationStrategy class="algorithm.initialisation.ChargedPopulationInitialisationStrategy" entityNumber="20">
        <chargeMagnitude value="1" />
        <chargedRatio value="0.5" />
        <entityType class="pso.dynamic.ChargedParticle">
            <velocityUpdateStrategy class="pso.dynamic.QuantumVelocityUpdateStrategy" />
            <positionUpdateStrategy class="pso.dynamic.QuantumPositionUpdateStrategy" radius="5"/>
        </entityType>
    </initialisationStrategy>
    <iterationStrategy class="pso.dynamic.DynamicIterationStrategy">
        <detectionStrategy class="pso.dynamic.detectionstrategies.AlwaysTrueDetectionStrategy" />
        <responseStrategy class="pso.dynamic.responsestrategies.ParticleReevaluationResponseStrategy"/>
    </iterationStrategy>
</algorithm>
<algorithm id="multiswarm_5" class="pso.multiswarm.MultiSwarm">
    <addStoppingCondition class="stoppingcondition.MaximumIterations" maximumIterations="1000"/>
        <multiSwarmIterationStrategy class="pso.multiswarm.MultiSwarmIterationStrategy" exclusionRadius="5.0"
convergenceRadius="5.0">
    </multiSwarmIterationStrategy>
    <algorithm idref="multi_quantum_20"/>
    <algorithm idref="multi_quantum_20"/>
    <algorithm idref="multi_quantum_20"/>
    <algorithm idref="multi_quantum_20"/>
    <algorithm idref="multi_quantum_20"/>
</algorithm>
```

```
    <algorithm id=“selfmultiswarm” class=“pso.multiswarm.MultiSwarm”>
        <addStoppingCondition class=“stoppingcondition.MaximumIterations” maximumIterations=“1000”/>
        <multiSwarmIterationStrategy class=“pso.multiswarm.SelfAdaptingMultiSwarmIterationStrategy” />
        <algorithm idref=“multi_quantum_20”/>
    </algorithm¿
</algorithms>
<problems>
    <problem id=“static_p1_d2” class=“problem.FunctionMaximisationProblem”>
        <function class=“functions.continuous.dynamic.MovingPeaks” domain=“R(0, 100)ˆ2”>
            <nextSeed value=“1”/>
            <changeFrequency value=“1” />
            <numberOfPeaks value=“1” />
            <vlength value=“0” />
            <lambda value=“0” />
            <heightSeverity value=“0.0” />
            <widthSeverity value=“0.0” />
        </function>
    </problem>
    <problem id=“progress_p1_d2” class=“problem.FunctionMaximisationProblem”>
        <function class=“functions.continuous.dynamic.MovingPeaks” domain=“R(0, 100)ˆ2”>
            <nextSeed value=“1”/>
            <changeFrequency value=“1” />
            <numberOfPeaks value=“1” />
            <vlength value=“1” />
            <lambda value=“0” />
            <heightSeverity value=“1.0” />
            <widthSeverity value=“0.05” />
        </function>
    </problem>
    <problem id=“abrupt_p1_d2” class=“problem.FunctionMaximisationProblem”>
        <function class=“functions.continuous.dynamic.MovingPeaks” domain=“R(0, 100)ˆ2”>
            <nextSeed value=“1”/>
            <changeFrequency value=“200” />
            <numberOfPeaks value=“1” />
            <vlength value=“50” />
            <lambda value=“0” />
            <heightSeverity value=“10.0” />
            <widthSeverity value=“0.05” />
        </function>
    </problem>
    <problem id=“chaos_p1_d2” class=“problem.FunctionMaximisationProblem”>
        <function class=“functions.continuous.dynamic.MovingPeaks” domain=“R(0, 100)ˆ2”>
            <nextSeed value=“1”/>
            <changeFrequency value=“5” />
            <numberOfPeaks value=“1” />
            <vlength value=“50” />
            <lambda value=“0” />
            <heightSeverity value=“10.0” />
            <widthSeverity value=“0.05” />
        </function>
    </problem>
    <problem id=“pattern_p1_d2” class=“problem.FunctionMaximisationProblem”>
        <function class=“functions.continuous.decorators.RotatingFunctionDecorator”>
            <cycleLength value=“100” />
            <rotatingFrequency value=“1” />
            <function class=“functions.continuous.dynamic.MovingPeaks” domain=“R(0, 100)ˆ2”>
                <nextSeed value=“1”/>
                <changeFrequency value=“1” />
                <numberOfPeaks value=“1” />
                <vlength value=“0” />
                <lambda value=“0” />
                <heightSeverity value=“1.0” />
                <widthSeverity value=“0.05” />
            </function>
        </function>
    </problem>
</problems>
```

```
<measurements>
    <measurements id=“measure_error” class=“simulator.MeasurementSuite” resolution=“1”>
        <addMeasurement class=“measurement.single.dynamic.ErrorMeasurement” />
    </measurements>
    <measurements id=“measure_abrupt” class=“simulator.MeasurementSuite” resolution=“1”>
        <addMeasurement class=“measurement.single.dynamic.ErrorMeasurement” />
        <addMeasurement class=“measurement.single.dynamic.AverageBestErrorBeforeChange” cycleSize=“200” />
        <addMeasurement class=“measurement.single.dynamic.AverageIterationsToErrorLimit” cycleSize=“200” limit=“10” />
    </measurements>
    <measurements id=“measure_chaos” class=“simulator.MeasurementSuite” resolution=“1”>
        <addMeasurement class=“measurement.single.dynamic.ErrorMeasurement” />
        <addMeasurement class=“measurement.single.dynamic.AverageBestErrorBeforeChange” cycleSize=“5” />
        <addMeasurement class=“measurement.single.dynamic.AverageIterationsToErrorLimit” cycleSize=“5” limit=“10” />
    </measurements>
</measurements>
<simulation samples=“30”>
    <algorithm idref=“reevpso”/>
    <problem idref=“progress_p1_d2”/>
    <measurements idref=“measure_error” />
    <output format=“TXT” file=“data/experiments/output/charged_10_progress_p1_d2.txt” />
</simulation>
</simulator>
```

Appendix C

Statistical Significance

This appendix lists the p-values obtained by the two-tailed Mann-Whitney U tests performed. The Mann-Whitney U test [60] is used to compare the results of two sets of simulations. The difference between two sets of results is considered statistically significant only if the p-value obtained is below 0.05. If the difference is significant, the U-values of the two sets are compared to determine which set contains greater values. The p-values are listed in separate tables. Each table compares the two algorithms specified in the caption of the table. Next to each p-value, a symbol notifies whether the difference was not significant ($\cong$), whether the first algorithm mentioned in the caption obtained a significantly larger error than the second one ($>$), or whether the first algorithm mentioned in the caption obtained a significantly lower error ($<$). The first column of a table contains the name of the environment with the measure used in brackets, and the number of peaks in the environment is indicated in the second column. The tables containing the p-values follow.

Table C.1: Comparison between standard PSO and re-evaluating PSO

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	7.034e−2 (≅)	5.440e−2 (≅)	8.205e−1 (≅)	8.504e−2 (≅)	9.476e−1 (≅)
	5 p	9.585e−2 (≅)	3.136e−1 (≅)	4.317e−1 (≅)	8.236e−2 (≅)	6.231e−1 (≅)
	15 p	6.653e−1 (≅)	5.325e−2 (≅)	2.187e−1 (≅)	2.860e−1 (≅)	2.478e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	5.521e−1 (≅)	9.826e−1 (≅)
	5 p	2.000e+0 (≅)	4.995e−1 (≅)	1.630e−1 (≅)	2.300e−1 (≅)	7.752e−1 (≅)
	15 p	2.000e+0 (≅)	2.000e+0 (≅)	4.409e−1 (≅)	3.207e−1 (≅)	9.942e−1 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)

Table C.2: Comparison between re-evaluating PSO and reinitialising PSO - 10%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	5.124e−2 (≅)	7.375e−2 (≅)	2.187e−1 (≅)	4.404e−1 (≅)	7.229e−2 (≅)
	5 p	5.228e−2 (≅)	8.779e−2 (≅)	9.062e−2 (≅)	4.492e−1 (≅)	3.428e−1 (≅)
	15 p	2.601e−1 (≅)	6.946e−2 (≅)	3.207e−1 (≅)	1.822e−1 (≅)	9.926e−2 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	3.504e−1 (≅)	4.946e−1 (≅)
	5 p	2.000e+0 (≅)	7.177e−1 (≅)	3.065e−1 (≅)	2.996e−1 (≅)	5.326e−1 (≅)
	15 p	2.000e+0 (≅)	3.233e−1 (≅)	2.193e−1 (≅)	9.351e−2 (≅)	1.381e−1 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	6.760e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	5.040e−1 (≅)	5.423e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.342e−3 (>)	6.546e−1 (≅)	4.946e−1 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	5.230e−1 (≅)	2.243e−1 (≅)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	3.195e−2 (>)	5.820e−1 (≅)	6.440e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.194e−1 (≅)	2.539e−1 (≅)	8.434e−1 (≅)
	15 p	1.000e−4 (>)	3.280e−1 (≅)	9.351e−2 (≅)	5.620e−1 (≅)	3.207e−1 (≅)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.776e−3 (<)	1.321e−2 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	8.549e−1 (≅)	2.414e−1 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.740e−4 (>)	2.104e−2 (>)	4.654e−1 (≅)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.340e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.340e−4 (>)	9.127e−1 (≅)	1.000e−4 (<)	1.524e−3 (>)	1.000e−4 (<)
	15 p	1.252e−2 (>)	2.664e−1 (≅)	5.326e−1 (≅)	1.000e−4 (<)	2.468e−3 (<)
E-CHAOS (ABEBC)	1 p	8.779e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	2.025e−1 (≅)	1.000e−4 (<)	6.047e−3 (>)	1.000e−4 (<)
	15 p	4.003e−4 (>)	8.434e−1 (≅)	3.195e−2 (<)	1.000e−4 (<)	1.366e−3 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.593e−3 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	3.099e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	9.985e−4 (>)	3.541e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	7.300e−4 (>)	1.000e−4 (<)

Table C.3: Comparison between re-evaluating PSO and reinitialising PSO - 30%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	7.334e−2 (≅)	8.811e−2 (≅)	1.006e+0 (≅)	7.752e−1 (≅)	5.134e−1 (≅)
	5 p	7.109e−2 (≅)	1.421e−1 (≅)	7.528e−1 (≅)	5.228e−2 (≅)	5.922e−1 (≅)
	15 p	2.243e−1 (≅)	1.922e−1 (≅)	1.058e−1 (≅)	7.305e−1 (≅)	5.134e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	8.091e−1 (≅)	2.025e−1 (≅)
	5 p	2.000e+0 (≅)	5.140e−1 (≅)	7.177e−1 (≅)	7.974e−2 (≅)	5.326e−1 (≅)
	15 p	2.000e+0 (≅)	2.000e+0 (≅)	7.000e−1 (≅)	5.423e−1 (≅)	7.639e−1 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.904e−3 (<)	6.047e−3 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.833e−2 (<)	7.639e−1 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	6.335e−1 (≅)	1.635e−1 (≅)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.093e−4 (>)	5.708e−2 (≅)	7.974e−2 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	8.664e−1 (≅)	9.244e−1 (≅)	2.478e−1 (≅)
	15 p	1.000e−4 (>)	1.486e−2 (>)	3.713e−2 (>)	8.664e−1 (≅)	5.423e−1 (≅)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	2.037e−3 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.694e−1 (≅)	1.523e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	3.000e−4 (>)	6.903e−1 (≅)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.547e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	2.434e−2 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ABEBC)	1 p	3.136e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	5.239e−3 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.872e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.229e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	5.325e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (<)

Table C.4: Comparison between re-evaluating PSO and reinitialising PSO - 50%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	5.120e−2 (≅)	9.054e−2 (≅)	1.304e−1 (≅)	6.868e−1 (≅)	3.207e−1 (≅)
	5 p	5.120e−2 (≅)	8.320e−1 (≅)	3.280e−1 (≅)	4.492e−1 (≅)	6.747e−2 (≅)
	15 p	5.720e−1 (≅)	7.015e−2 (≅)	3.580e−1 (≅)	4.946e−1 (≅)	5.326e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	1.381e−1 (≅)	4.404e−1 (≅)
	5 p	2.000e+0 (≅)	5.914e−1 (≅)	7.177e−1 (≅)	1.590e−1 (≅)	9.676e−2 (≅)
	15 p	2.000e+0 (≅)	3.233e−1 (≅)	7.901e−1 (≅)	9.826e−1 (≅)	8.780e−1 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.524e−3 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.421e−1 (≅)	2.750e−4 (<)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	8.379e−3 (>)	1.000e−4 (<)	8.003e−3 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	4.492e−1 (≅)	2.532e−2 (<)	1.504e−1 (≅)
	15 p	1.000e−4 (>)	4.963e−2 (>)	1.252e−2 (>)	1.006e+0 (≅)	7.471e−2 (≅)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	4.852e−1 (≅)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	2.474e−2 (>)	6.413e−1 (≅)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	5.109e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.194e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ABEBC)	1 p	1.635e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	4.671e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	2.947e−3 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.845e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	2.728e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	8.202e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	2.158e−2 (>)	1.000e−4 (<)	1.000e−4 (<)	1.050e−2 (>)	1.000e−4 (<)

Table C.5: Comparison between re-evaluating PSO and reinitialising PSO - 70%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	5.120e−2 (≅)	6.636e−2 (≅)	6.541e−2 (≅)	7.195e−1 (≅)	1.822e−1 (≅)
	5 p	5.072e−2 (≅)	3.980e−1 (≅)	1.680e−1 (≅)	9.127e−1 (≅)	4.581e−1 (≅)
	15 p	4.762e−1 (≅)	5.040e−1 (≅)	1.421e−1 (≅)	4.146e−1 (≅)	1.266e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	8.780e−1 (≅)	9.476e−1 (≅)
	5 p	2.000e+0 (≅)	6.726e−1 (≅)	1.904e−1 (≅)	6.024e−1 (≅)	8.664e−1 (≅)
	15 p	2.000e+0 (≅)	2.000e+0 (≅)	1.685e−1 (≅)	6.546e−1 (≅)	3.817e−1 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.794e−3 (<)	1.000e−4 (<)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.664e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	3.737e−1 (≅)	2.269e−4 (<)	4.528e−3 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.973e−1 (≅)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	3.670e−3 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	3.736e−1 (≅)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.169e−1 (≅)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.365e−2 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	5.708e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	5.093e−3 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	2.750e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.443e−3 (<)	1.000e−4 (<)

Table C.6: Comparison between re-evaluating PSO and reinitialising PSO - 90%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	6.603e−2 (≅)	8.795e−2 (≅)	6.760e−1 (≅)	8.779e−2 (≅)	5.922e−1 (≅)
	5 p	9.344e−2 (≅)	3.065e−1 (≅)	6.653e−1 (≅)	4.317e−1 (≅)	6.868e−1 (≅)
	15 p	2.132e−1 (≅)	3.504e−1 (≅)	4.853e−1 (≅)	9.942e−1 (≅)	3.898e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.078e−1 (≅)	7.865e−1 (≅)
	5 p	2.000e+0 (≅)	7.637e−1 (≅)	9.979e−1 (≅)	9.127e−1 (≅)	8.896e−1 (≅)
	15 p	2.000e+0 (≅)	3.233e−1 (≅)	3.971e−1 (≅)	6.976e−1 (≅)	7.639e−1 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.687e−2 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.424e−2 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	5.040e−1 (≅)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	6.205e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.409e−1 (≅)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.086e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)

Table C.7: Comparison between re-evaluating PSO and APSO - 10%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	7.639e−1 (≅)	8.091e−1 (≅)	5.620e−1 (≅)	2.996e−1 (≅)	7.865e−1 (≅)
	5 p	8.779e−2 (≅)	8.320e−1 (≅)	7.471e−2 (≅)	2.539e−1 (≅)	2.025e−1 (≅)
	15 p	9.826e−1 (≅)	8.549e−1 (≅)	8.205e−1 (≅)	4.492e−1 (≅)	9.476e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	9.709e−1 (≅)	1.774e−1 (≅)
	5 p	2.000e+0 (≅)	7.637e−1 (≅)	8.362e−1 (≅)	5.708e−2 (≅)	8.091e−1 (≅)
	15 p	2.000e+0 (≅)	3.233e−1 (≅)	3.038e−1 (≅)	4.492e−1 (≅)	8.091e−1 (≅)
E-PROGRESS (CME)	1 p	2.078e−1 (≅)	3.280e−1 (≅)	7.416e−1 (≅)	6.976e−1 (≅)	9.244e−1 (≅)
	5 p	1.000e−4 (>)	1.421e−1 (≅)	8.549e−1 (≅)	2.132e−1 (≅)	3.580e−1 (≅)
	15 p	1.000e−4 (>)	6.024e−1 (≅)	4.671e−1 (≅)	9.360e−1 (≅)	2.996e−1 (≅)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.428e−1 (≅)	5.423e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.026e−1 (≅)	5.620e−1 (≅)
	15 p	2.269e−4 (>)	1.998e−3 (>)	1.000e−4 (>)	1.304e−1 (≅)	8.549e−1 (≅)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.504e−1 (≅)	2.601e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	9.244e−1 (≅)	6.760e−1 (≅)	7.416e−1 (≅)
	15 p	6.116e−4 (>)	1.266e−1 (≅)	2.078e−1 (≅)	7.719e−2 (≅)	3.658e−1 (≅)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	2.785e−2 (<)	1.754e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.452e−2 (>)	8.766e−2 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.126e−4 (>)	7.611e−3 (>)	1.664e−1 (≅)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	5.514e−2 (≅)	5.922e−1 (≅)	5.326e−1 (≅)	4.671e−1 (≅)	9.244e−1 (≅)
	5 p	7.752e−1 (≅)	7.752e−1 (≅)	7.639e−1 (≅)	9.826e−1 (≅)	8.896e−1 (≅)
	15 p	4.146e−1 (≅)	7.195e−1 (≅)	7.195e−1 (≅)	9.244e−1 (≅)	6.440e−1 (≅)
E-CHAOS (ABEBC)	1 p	6.440e−1 (≅)	9.709e−1 (≅)	7.195e−1 (≅)	4.317e−1 (≅)	9.127e−1 (≅)
	5 p	4.317e−1 (≅)	9.011e−1 (≅)	3.207e−1 (≅)	9.942e−1 (≅)	8.205e−1 (≅)
	15 p	5.230e−1 (≅)	3.980e−1 (≅)	7.085e−1 (≅)	9.011e−1 (≅)	6.335e−1 (≅)
E-CHAOS (ITEL)	1 p	2.154e−2 (>)	7.694e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	3.735e−1 (≅)	4.403e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.566e−1 (≅)	3.895e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	8.896e−1 (≅)	6.024e−1 (≅)	9.942e−1 (≅)	9.826e−1 (≅)	8.434e−1 (≅)
	5 p	2.847e−2 (<)	7.085e−1 (≅)	3.280e−1 (≅)	2.633e−2 (>)	1.922e−1 (≅)
	15 p	3.763e−4 (>)	1.266e−1 (≅)	9.127e−1 (≅)	6.231e−1 (≅)	5.820e−1 (≅)

Table C.8: Comparison between re-evaluating PSO and APSO - 30%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	2.132e−1 (≅)	3.980e−1 (≅)	1.304e−1 (≅)	7.865e−1 (≅)	7.978e−1 (≅)
	5 p	6.231e−1 (≅)	9.942e−1 (≅)	7.752e−1 (≅)	1.727e−1 (≅)	6.546e−1 (≅)
	15 p	9.476e−1 (≅)	2.078e−1 (≅)	4.459e−2 (<)	4.963e−2 (<)	3.207e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	9.557e−1 (≅)	2.347e−2 (<)	2.269e−4 (<)	4.146e−1 (≅)
	15 p	2.000e+0 (≅)	3.233e−1 (≅)	3.295e−2 (<)	3.904e−3 (<)	4.062e−1 (≅)
E-PROGRESS (CME)	1 p	4.581e−1 (≅)	5.766e−3 (<)	2.222e−3 (<)	2.132e−1 (≅)	7.978e−1 (≅)
	5 p	1.000e−4 (>)	1.687e−2 (<)	1.635e−1 (≅)	1.617e−2 (<)	1.421e−1 (≅)
	15 p	1.000e−4 (>)	1.727e−1 (≅)	2.187e−1 (≅)	2.927e−1 (≅)	7.305e−1 (≅)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.207e−1 (≅)	5.423e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	4.317e−1 (≅)	5.326e−1 (≅)
	15 p	1.000e−4 (>)	1.222e−3 (>)	1.000e−4 (>)	2.132e−1 (≅)	3.658e−1 (≅)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	3.580e−1 (≅)	1.000e−4 (<)	4.790e−2 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	8.379e−3 (<)	3.318e−2 (<)	5.326e−1 (≅)
	15 p	1.222e−3 (>)	1.230e−1 (≅)	8.434e−1 (≅)	1.342e−1 (≅)	3.504e−1 (≅)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	6.347e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	8.226e−2 (≅)	1.321e−3 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	3.308e−4 (>)	1.000e−4 (>)	4.285e−1 (≅)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	2.078e−1 (≅)	8.780e−1 (≅)	4.317e−1 (≅)	6.868e−1 (≅)	6.653e−1 (≅)
	5 p	7.085e−1 (≅)	7.639e−1 (≅)	3.353e−1 (≅)	5.230e−1 (≅)	5.423e−1 (≅)
	15 p	8.434e−1 (≅)	3.980e−1 (≅)	1.973e−1 (≅)	5.040e−1 (≅)	8.896e−1 (≅)
E-CHAOS (ABEBC)	1 p	2.532e−2 (<)	4.671e−1 (≅)	5.134e−1 (≅)	6.335e−1 (≅)	6.760e−1 (≅)
	5 p	2.959e−2 (<)	6.653e−1 (≅)	8.549e−1 (≅)	5.134e−1 (≅)	6.024e−1 (≅)
	15 p	9.826e−1 (≅)	2.860e−1 (≅)	6.231e−1 (≅)	4.581e−1 (≅)	8.549e−1 (≅)
E-CHAOS (ITEL)	1 p	3.313e−2 (>)	4.532e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	9.011e−1 (≅)	9.651e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	4.315e−1 (≅)	3.735e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.194e−1 (≅)	7.528e−1 (≅)	4.946e−1 (≅)	8.779e−2 (≅)	2.860e−1 (≅)
	5 p	1.252e−2 (<)	1.680e−1 (≅)	3.817e−1 (≅)	1.872e−1 (≅)	6.440e−1 (≅)
	15 p	1.000e−4 (>)	1.026e−1 (≅)	4.231e−1 (≅)	1.230e−1 (≅)	6.760e−1 (≅)

Table C.9: Comparison between re-evaluating PSO and APSO - 50%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	2.737e−3 (<)	1.421e−1 (≅)	8.320e−1 (≅)	5.326e−1 (≅)	2.860e−1 (≅)
	5 p	2.532e−2 (<)	4.317e−1 (≅)	1.922e−1 (≅)	5.514e−2 (≅)	7.639e−1 (≅)
	15 p	5.720e−1 (≅)	4.492e−1 (≅)	7.974e−2 (≅)	5.521e−1 (≅)	1.124e−1 (≅)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.979e−1 (≅)	1.000e−4 (<)	5.109e−4 (<)	9.176e−3 (<)
	15 p	2.000e+0 (≅)	2.000e+0 (≅)	5.178e−3 (<)	6.965e−3 (<)	3.713e−2 (<)
E-PROGRESS (CME)	1 p	9.351e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	2.107e−3 (<)	5.239e−3 (<)
	5 p	1.000e−4 (>)	5.239e−3 (<)	3.207e−1 (≅)	1.000e−4 (<)	1.307e−2 (<)
	15 p	1.000e−4 (>)	4.622e−2 (>)	2.025e−1 (≅)	2.478e−1 (≅)	5.134e−1 (≅)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.713e−2 (<)	1.774e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	2.025e−1 (≅)	6.760e−1 (≅)
	15 p	1.000e−4 (>)	3.904e−3 (>)	1.000e−4 (>)	2.847e−2 (<)	5.134e−1 (≅)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.867e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	5.040e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	6.113e−2 (≅)
	15 p	1.524e−3 (>)	8.769e−3 (<)	1.000e−4 (<)	1.617e−2 (<)	1.004e−2 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.895e−1 (≅)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	6.280e−4 (>)	1.000e−4 (>)	7.838e−3 (<)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	4.671e−1 (≅)	4.062e−1 (≅)	4.062e−1 (≅)	5.922e−1 (≅)	2.132e−1 (≅)
	5 p	7.978e−1 (≅)	4.317e−1 (≅)	4.853e−1 (≅)	3.065e−1 (≅)	5.134e−1 (≅)
	15 p	9.244e−1 (≅)	8.434e−1 (≅)	3.065e−1 (≅)	7.752e−1 (≅)	8.205e−1 (≅)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	4.762e−1 (≅)	9.942e−1 (≅)	6.868e−1 (≅)	2.025e−1 (≅)
	5 p	3.737e−1 (≅)	9.593e−1 (≅)	1.091e−1 (≅)	3.353e−1 (≅)	4.946e−1 (≅)
	15 p	4.317e−1 (≅)	3.065e−1 (≅)	4.231e−1 (≅)	8.896e−1 (≅)	8.434e−1 (≅)
E-CHAOS (ITEL)	1 p	1.725e−1 (≅)	2.102e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	9.344e−2 (≅)	9.360e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	2.269e−1 (≅)	4.534e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	6.335e−1 (≅)	8.205e−1 (≅)	6.231e−1 (≅)	6.546e−1 (≅)
	5 p	1.000e−4 (<)	8.664e−1 (≅)	5.820e−1 (≅)	9.062e−2 (≅)	1.922e−1 (≅)
	15 p	1.000e−4 (>)	4.622e−2 (<)	6.976e−1 (≅)	4.946e−1 (≅)	5.922e−1 (≅)

Table C.10: Comparison between re-evaluating PSO and APSO - 70%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.486e−2 (<)	1.547e−1 (≅)	2.601e−1 (≅)	2.580e−4 (<)	7.297e−3 (<)
	5 p	1.230e−1 (≅)	7.229e−2 (≅)	3.577e−2 (<)	2.420e−4 (<)	5.134e−1 (≅)
	15 p	9.709e−1 (≅)	4.528e−3 (<)	9.351e−2 (≅)	6.047e−3 (<)	2.930e−4 (<)
E-STATIC (Error)	1 p	2.000e+0 (≅)	3.233e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.250e−2 (<)	1.000e−4 (<)	1.000e−4 (<)	2.339e−2 (<)
	15 p	2.000e+0 (≅)	1.073e−3 (<)	4.893e−3 (<)	1.700e−3 (<)	1.000e−4 (<)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	3.714e−3 (<)
	5 p	1.000e−4 (>)	5.109e−4 (<)	2.107e−3 (<)	1.000e−4 (<)	9.648e−2 (≅)
	15 p	1.000e−4 (>)	1.159e−1 (≅)	5.708e−2 (≅)	9.648e−2 (≅)	1.973e−1 (≅)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.155e−3 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	4.103e−3 (<)	6.113e−2 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.207e−1 (≅)	6.765e−2 (≅)
E-ABRUPT (ABEBC)	1 p	8.504e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	9.942e−1 (≅)	2.580e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	5.766e−3 (>)	1.147e−2 (<)	8.690e−4 (<)	1.366e−3 (<)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	9.127e−1 (≅)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.043e−3 (<)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	4.146e−1 (≅)	3.737e−1 (≅)	9.476e−1 (≅)	4.317e−1 (≅)	9.127e−1 (≅)
	5 p	9.709e−1 (≅)	3.898e−1 (≅)	3.353e−1 (≅)	8.434e−1 (≅)	4.853e−1 (≅)
	15 p	5.720e−1 (≅)	5.134e−1 (≅)	3.580e−1 (≅)	4.492e−1 (≅)	2.478e−1 (≅)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	3.713e−2 (<)	5.521e−1 (≅)	2.358e−1 (≅)	8.320e−1 (≅)
	5 p	3.075e−2 (<)	8.434e−1 (≅)	3.980e−1 (≅)	8.434e−1 (≅)	4.231e−1 (≅)
	15 p	3.075e−2 (<)	3.817e−1 (≅)	9.709e−1 (≅)	3.980e−1 (≅)	2.664e−1 (≅)
E-CHAOS (ITEL)	1 p	8.166e−4 (>)	1.518e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	3.207e−4 (>)	6.545e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	4.989e−1 (≅)	6.177e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	4.963e−2 (<)	6.335e−1 (≅)	6.127e−1 (≅)	9.476e−1 (≅)
	5 p	1.000e−4 (<)	7.305e−1 (≅)	6.231e−1 (≅)	1.973e−1 (≅)	5.720e−1 (≅)
	15 p	1.000e−4 (>)	3.577e−2 (<)	1.304e−1 (≅)	9.360e−1 (≅)	5.820e−1 (≅)

Table C.11: Comparison between re-evaluating PSO and APSO - 90%

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	7.297e−3 (<)	1.748e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	3.763e−4 (<)
	15 p	1.894e−3 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-STATIC (Error)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	4.003e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	4.003e−4 (<)	2.750e−4 (<)	1.000e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	3.034e−3 (<)
	15 p	1.000e−4 (>)	2.996e−1 (≅)	1.000e−4 (>)	1.774e−1 (≅)	1.155e−3 (<)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	3.207e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	9.176e−3 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	3.897e−1 (≅)	8.332e−3 (>)	1.000e−4 (<)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.504e−1 (≅)	3.737e−1 (≅)	7.229e−2 (≅)	2.601e−1 (≅)	6.760e−1 (≅)
	5 p	8.091e−1 (≅)	7.978e−1 (≅)	8.091e−1 (≅)	8.091e−1 (≅)	8.320e−1 (≅)
	15 p	2.847e−2 (<)	2.996e−1 (≅)	3.065e−1 (≅)	8.664e−1 (≅)	7.865e−1 (≅)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	4.301e−2 (<)	6.765e−2 (≅)	1.973e−1 (≅)	7.416e−1 (≅)
	5 p	1.531e−4 (<)	4.762e−1 (≅)	5.521e−1 (≅)	8.205e−1 (≅)	8.320e−1 (≅)
	15 p	1.000e−4 (<)	1.026e−1 (≅)	7.195e−1 (≅)	8.320e−1 (≅)	7.752e−1 (≅)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	3.372e−2 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.090e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	2.960e−1 (≅)	7.140e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	8.379e−3 (<)	9.244e−1 (≅)	6.976e−1 (≅)	3.280e−1 (≅)
	5 p	1.000e−4 (<)	8.549e−1 (≅)	6.127e−1 (≅)	8.779e−2 (≅)	9.011e−1 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (<)	4.853e−1 (≅)	3.353e−1 (≅)	2.927e−1 (≅)

Table C.12: Comparison between re-evaluating PSO and QSO - 10%, radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.894e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	2.860e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.292e−3 (>)	7.195e−1 (≅)	1.198e−2 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (>)	1.000e−4 (>)
	5 p	2.000e+0 (≅)	4.187e−2 (<)	9.191e−1 (≅)	9.176e−3 (>)	2.222e−3 (>)
	15 p	2.000e+0 (≅)	7.937e−2 (≅)	8.726e−4 (<)	2.025e−1 (≅)	2.927e−1 (≅)
E-PROGRESS (CME)	1 p	6.047e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	7.195e−1 (≅)	5.908e−2 (≅)	5.325e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	4.762e−1 (≅)	7.528e−1 (≅)	1.443e−3 (<)	2.738e−2 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.252e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	9.826e−1 (≅)	6.113e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.774e−1 (≅)	5.141e−2 (≅)	2.417e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	2.601e−1 (≅)	1.000e−4 (<)	3.532e−3 (>)	6.868e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	6.922e−1 (≅)	1.000e−4 (>)	2.728e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	8.779e−1 (≅)	1.229e−1 (≅)	1.000e−4 (>)	3.602e−1 (≅)	1.000e−4 (>)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	2.996e−1 (≅)	1.531e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	7.643e−3 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	2.532e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	9.127e−1 (≅)	2.078e−1 (≅)
	5 p	1.381e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	6.653e−1 (≅)	2.339e−2 (<)
	15 p	2.927e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	4.528e−3 (<)	6.341e−3 (<)

Table C.13: Comparison between re-evaluating PSO and QSO - 30%, radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	5.134e−1 (≅)	5.766e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	8.091e−1 (≅)	1.680e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.570e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	5 p	2.000e+0 (≅)	1.190e−3 (<)	2.302e−4 (<)	2.728e−1 (≅)	4.317e−1 (≅)
	15 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	8.780e−1 (≅)	6.113e−2 (≅)
E-PROGRESS (CME)	1 p	1.292e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	6.760e−1 (≅)	1.000e−4 (<)	5.766e−3 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	2.078e−1 (≅)	3.537e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	9.593e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	2.187e−1 (≅)	1.610e−3 (<)	1.032e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	4.755e−3 (<)	4.853e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (<)	1.000e−4 (<)	7.085e−1 (≅)	6.231e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	5.391e−4 (<)	1.000e−4 (>)	9.526e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.227e−1 (≅)	1.290e−3 (<)	7.706e−2 (≅)	7.682e−2 (≅)	1.000e−4 (>)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	7.978e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	2.750e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.797e−1 (≅)	1.000e−4 (>)	3.233e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	3.519e−3 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	2.994e−1 (≅)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.050e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	5.040e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)

Table C.14: Comparison between re-evaluating PSO and QSO - 50%, radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (<)	8.549e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	3.898e−1 (≅)	3.737e−1 (≅)	3.577e−2 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)
	5 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	7.297e−3 (<)	8.205e−1 (≅)
	15 p	2.000e+0 (≅)	1.000e−4 (<)	2.038e−4 (<)	1.000e−4 (<)	1.424e−2 (<)
E-PROGRESS (CME)	1 p	5.497e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	9.244e−1 (≅)	1.000e−4 (<)	9.176e−3 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.462e−1 (≅)	1.000e−4 (<)	1.610e−3 (<)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	5.423e−1 (≅)	1.000e−4 (>)	1.252e−2 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	3.998e−2 (<)	4.257e−4 (<)	1.867e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	8.003e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.990e−2 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.774e−1 (≅)	4.062e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (>)	3.056e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	7.014e−2 (≅)	1.000e−4 (<)	1.437e−3 (>)	1.696e−1 (≅)	1.000e−4 (>)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	9.244e−1 (≅)	9.476e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	6.868e−1 (≅)	6.765e−2 (≅)	1.433e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (<)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (<)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (<)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	3.817e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.300e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	7.195e−1 (≅)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)

Table C.15: Comparison between re-evaluating PSO and QSO - 70%, radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	4.103e−3 (<)	6.324e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	9.952e−2 (≅)	4.003e−4 (<)	1.774e−1 (≅)	6.047e−3 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	3.233e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.366e−3 (<)
	15 p	3.233e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	5.427e−4 (<)
E-PROGRESS (CME)	1 p	3.532e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	6.868e−1 (≅)	1.000e−4 (<)	3.120e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	5.820e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.504e−1 (≅)	1.000e−4 (>)	3.446e−2 (>)	1.366e−3 (>)	1.000e−4 (>)
	15 p	2.882e−3 (<)	1.340e−4 (<)	6.646e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	7.195e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	7.297e−3 (>)	1.000e−4 (>)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.358e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (>)	3.053e−1 (≅)	7.191e−3 (>)	1.000e−4 (>)
	15 p	1.426e−1 (≅)	1.000e−4 (<)	2.241e−2 (>)	7.387e−1 (≅)	1.000e−4 (>)
E-CHAOS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	4.963e−2 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.822e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	3.065e−1 (≅)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	5.521e−1 (≅)	1.000e−4 (>)	1.004e−2 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.424e−2 (>)	1.000e−4 (>)	8.896e−1 (≅)	2.025e−1 (≅)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (<)	1.000e−4 (>)	3.233e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (<)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (<)	1.000e−4 (>)	7.925e−2 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	3.065e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	5.040e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)

Table C.16: Comparison between re-evaluating PSO and QSO - 90%, radius: 5

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (>)	1.304e−1 (≅)	9.011e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	5.497e−3 (<)	1.524e−3 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.381e−1 (≅)	2.247e−2 (<)	3.537e−4 (<)	8.779e−2 (≅)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.058e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-PROGRESS (CME)	1 p	1.421e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	5.427e−4 (<)	1.000e−4 (<)	3.714e−3 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	2.860e−1 (≅)	1.340e−4 (<)	1.000e−4 (<)	2.420e−4 (>)	8.379e−3 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	5.908e−2 (≅)	1.000e−4 (>)	1.894e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	3.532e−3 (<)	1.000e−4 (<)	2.358e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	4.946e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (>)	6.865e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	9.611e−2 (≅)	1.000e−4 (<)	3.205e−1 (≅)	1.169e−1 (≅)	1.000e−4 (>)
E-CHAOS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	4.963e−2 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.633e−2 (>)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	3.358e−3 (<)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.078e−1 (≅)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.050e−2 (>)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	5.497e−3 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (<)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (<)	6.023e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (<)	5.373e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	3.898e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	4.231e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)

Table C.17: Comparison between re-evaluating PSO and QSO - 10%, radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	7.229e−2 (≅)	5.514e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	6.024e−1 (≅)	5.325e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	3.580e−1 (≅)	3.136e−1 (≅)	6.024e−1 (≅)	2.127e−4 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	1.610e−3 (>)	1.000e−4 (>)
	5 p	2.000e+0 (≅)	5.914e−1 (≅)	1.109e−1 (≅)	1.727e−1 (≅)	3.353e−1 (≅)
	15 p	2.000e+0 (≅)	1.570e−1 (≅)	1.734e−2 (<)	8.091e−1 (≅)	8.236e−2 (≅)
E-PROGRESS (CME)	1 p	1.922e−1 (≅)	1.194e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	4.147e−2 (<)	1.421e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	4.622e−2 (>)	3.322e−4 (<)	3.280e−1 (≅)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.252e−4 (>)	2.750e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	5.708e−2 (≅)	1.000e−4 (>)	1.194e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.990e−2 (>)	1.159e−1 (≅)	6.646e−3 (>)	1.000e−4 (>)
	15 p	2.580e−4 (>)	2.738e−2 (<)	9.593e−1 (≅)	8.504e−2 (≅)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	2.992e−1 (≅)	2.419e−3 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.327e−1 (≅)	9.534e−1 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	6.210e−2 (≅)	4.533e−2 (>)	1.084e−1 (≅)	3.233e−1 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.998e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	5.134e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.680e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	4.991e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	5.514e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.173e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.049e−2 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.589e−1 (≅)	8.169e−3 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	6.546e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	2.434e−2 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.252e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)

Table C.18: Comparison between re-evaluating PSO and QSO - 30%, radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	3.136e−1 (≅)	6.541e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	2.132e−1 (≅)	5.720e−1 (≅)	3.428e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	5.423e−1 (≅)	1.910e−2 (<)	8.780e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	3.658e−1 (≅)
	5 p	2.000e+0 (≅)	1.694e−2 (<)	1.000e−4 (<)	1.000e−4 (<)	9.476e−1 (≅)
	15 p	2.000e+0 (≅)	1.000e−4 (<)	9.480e−4 (<)	7.229e−2 (≅)	4.853e−1 (≅)
E-PROGRESS (CME)	1 p	8.091e−1 (≅)	7.229e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.292e−3 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.687e−2 (>)	1.867e−4 (<)	1.700e−3 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	6.324e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.680e−1 (≅)	1.000e−4 (>)	1.124e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	9.205e−4 (>)	1.610e−3 (<)	6.324e−2 (≅)	1.000e−4 (>)
	5 p	1.000e−4 (>)	8.091e−1 (≅)	1.000e−4 (<)	2.601e−1 (≅)	1.000e−4 (>)
	15 p	7.739e−4 (>)	2.269e−4 (<)	1.307e−2 (<)	1.617e−2 (<)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	9.942e−1 (≅)	1.050e−3 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.124e−1 (≅)	6.659e−4 (>)	4.111e−2 (<)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.155e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	3.195e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	2.269e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	6.965e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.687e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.047e−2 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)

Table C.19: Comparison between re-evaluating PSO and QSO - 50%, radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	2.434e−2 (<)	4.528e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.973e−1 (≅)	1.252e−2 (<)	3.504e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	3.737e−1 (≅)	1.000e−4 (<)	3.998e−2 (<)	3.192e−3 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.959e−2 (<)
	15 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	3.120e−4 (<)	2.434e−2 (<)
E-PROGRESS (CME)	1 p	2.633e−2 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	8.780e−1 (≅)	1.000e−4 (<)	2.737e−3 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	6.231e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	2.996e−1 (≅)	2.737e−3 (>)	3.817e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.822e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	4.528e−3 (>)
	15 p	8.769e−3 (>)	1.000e−4 (<)	1.000e−4 (<)	1.748e−4 (<)	2.269e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	5.243e−3 (<)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	4.146e−1 (≅)	1.025e−1 (≅)	1.151e−3 (<)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	4.671e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	3.065e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	7.085e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	9.787e−2 (≅)	1.000e−4 (>)	3.233e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	2.073e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)

Table C.20: Comparison between re-evaluating PSO and QSO - 70%, radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (<)	4.946e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	9.648e−2 (≅)	2.247e−2 (<)	9.011e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)
	15 p	4.404e−1 (≅)	1.155e−3 (<)	8.780e−1 (≅)	1.547e−1 (≅)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	9.826e−1 (≅)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.266e−1 (≅)	1.000e−4 (<)	5.720e−1 (≅)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.737e−3 (>)	6.116e−4 (>)	1.000e−4 (>)
	15 p	1.636e−4 (>)	9.599e−3 (<)	5.109e−4 (>)	8.664e−1 (≅)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.292e−3 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.794e−3 (<)	1.000e−4 (<)	1.000e−4 (<)	3.577e−2 (<)
	15 p	2.339e−2 (>)	1.000e−4 (<)	1.093e−4 (<)	1.000e−4 (<)	6.646e−3 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	9.981e−3 (>)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	7.290e−3 (<)	3.706e−2 (>)	1.000e−4 (<)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	3.353e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	7.639e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.305e−2 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	4.229e−1 (≅)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	3.120e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	4.809e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)

Table C.21: Comparison between re-evaluating PSO and QSO - 90%, radius: 50

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	9.011e−1 (≅)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.860e−1 (≅)
	15 p	2.468e−3 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.728e−1 (≅)
E-STATIC (Error)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.794e−3 (>)
	15 p	1.000e−4 (>)	7.719e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	3.980e−1 (≅)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	6.994e−2 (≅)	1.000e−4 (<)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)
	15 p	1.524e−3 (>)	1.000e−4 (<)	1.000e−4 (<)	8.769e−3 (<)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	3.904e−3 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	4.790e−2 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	3.206e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	4.809e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (<)	1.000e−4 (>)	3.233e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (<)	1.000e−4 (>)	3.233e−1 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.486e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.774e−1 (≅)	2.468e−3 (<)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)

Table C.22: Comparison between re-evaluating PSO and multi-swarm with five sub-swarms

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	4.790e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	3.532e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	3.835e−2 (>)	1.000e−4 (<)	4.459e−2 (>)	3.280e−1 (≅)
	15 p	2.000e+0 (≅)	3.233e−1 (≅)	3.196e−1 (≅)	3.713e−2 (>)	4.622e−2 (>)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	2.127e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.524e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	8.664e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	2.882e−3 (>)	2.158e−2 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.759e−2 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (<)	2.599e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (>)	7.471e−2 (≅)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.340e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	6.868e−1 (≅)	1.000e−4 (<)
	5 p	1.000e−4 (<)	8.003e−3 (>)	2.127e−4 (<)	1.292e−3 (>)	2.107e−3 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	5.230e−1 (≅)	6.440e−1 (≅)	1.748e−4 (<)

Table C.23: Comparison between re-evaluating PSO and multi-swarm with 10 sub-swarms

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.366e−3 (<)	2.132e−1 (≅)	4.317e−1 (≅)	7.639e−1 (≅)	6.047e−3 (<)
	5 p	6.231e−1 (≅)	1.000e−4 (>)	4.755e−3 (>)	1.617e−2 (>)	1.000e−4 (>)
	15 p	7.300e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.724e−1 (≅)	1.000e−4 (<)	1.381e−1 (≅)	3.817e−1 (≅)
	15 p	2.000e+0 (≅)	1.000e−4 (<)	6.646e−1 (≅)	9.176e−3 (>)	5.708e−2 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.300e−1 (≅)	7.865e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.006e+0 (≅)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	3.904e−3 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	4.622e−2 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	6.127e−1 (≅)	5.109e−4 (>)
E-CHAOS (ABEBC)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.304e−1 (≅)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	4.317e−1 (≅)	8.690e−4 (>)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.246e−2 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	5.415e−2 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	2.882e−3 (>)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.867e−4 (>)
	15 p	1.000e−4 (>)	1.021e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)

Table C.24: Comparison between QSO - 50%, radius: 5 and multi-swarm with five sub-swarms

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (>)	9.244e−1 (≅)	1.000e−4 (<)	6.047e−3 (<)
	15 p	1.000e−4 (<)	2.132e−1 (≅)	1.922e−1 (≅)	2.468e−3 (>)	1.170e−4 (<)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.000e−4 (>)	8.374e−1 (≅)	1.000e−4 (>)	7.229e−2 (≅)
	15 p	2.000e+0 (≅)	1.000e−4 (>)	2.139e−3 (>)	1.000e−4 (>)	1.000e−4 (>)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	5.134e−1 (≅)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.342e−1 (≅)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.050e−2 (>)	1.000e−4 (<)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	9.205e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	4.459e−2 (>)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	8.087e−1 (≅)	1.000e−4 (>)	1.000e−4 (>)	7.195e−1 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.856e−1 (≅)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	5.720e−1 (≅)	8.504e−2 (≅)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	6.868e−1 (≅)	2.358e−1 (≅)	4.762e−1 (≅)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	2.300e−1 (≅)	1.124e−1 (≅)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	8.236e−2 (≅)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)

Table C.25: Comparison between QSO - 50%, radius: 5 and multi-swarm with 10 sub-swarms

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (>)	1.006e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	3.428e−1 (≅)	1.872e−1 (≅)	2.996e−1 (≅)	1.000e−4 (<)
E-STATIC (Error)	1 p	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	1.000e−4 (>)	1.006e+0 (≅)	1.748e−4 (>)	2.633e−2 (>)
	15 p	2.000e+0 (≅)	7.207e−3 (>)	2.707e−4 (>)	1.000e−4 (>)	1.000e−4 (>)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.894e−3 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	5.908e−2 (≅)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	9.011e−1 (≅)	1.000e−4 (<)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	7.471e−2 (≅)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	8.772e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	2.972e−2 (<)	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (<)
E-CHAOS (CME)	1 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	9.599e−3 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (>)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)
	15 p	1.000e−4 (>)	2.342e−3 (<)	1.000e−4 (<)	1.000e−4 (>)	1.000e−4 (>)

Table C.26: Comparison between Re-evaluating PSO and SAMS

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	4.003e−4 (<)	6.868e−1 (≅)	8.205e−1 (≅)	1.222e−3 (>)	6.541e−2 (≅)
	5 p	7.739e−4 (<)	8.779e−2 (≅)	1.421e−1 (≅)	1.366e−3 (>)	1.000e−4 (>)
	15 p	1.093e−4 (<)	1.000e−4 (<)	7.974e−2 (≅)	1.833e−2 (>)	1.000e−4 (>)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	2.000e+0 (≅)	2.023e−1 (≅)	1.000e−4 (<)	6.324e−2 (≅)	6.341e−3 (<)
	15 p	3.233e−1 (≅)	1.000e−4 (<)	7.747e−1 (≅)	3.446e−2 (>)	2.794e−1 (≅)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	7.305e−1 (≅)	1.000e−4 (<)	4.103e−3 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (>)	1.000e−4 (>)	8.504e−2 (≅)	1.006e+0 (≅)	1.170e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	4.404e−1 (≅)	4.003e−4 (<)
	5 p	1.000e−4 (>)	3.737e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	2.222e−3 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.927e−1 (≅)	1.973e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.759e−2 (>)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	4.064e−1 (≅)
	5 p	1.000e−4 (>)	1.000e−4 (>)	1.000e−4 (>)	3.503e−1 (≅)	2.034e−2 (>)
	15 p	1.000e−4 (>)	8.428e−4 (>)	1.000e−4 (>)	7.506e−4 (>)	2.329e−3 (>)
E-CHAOS (CME)	1 p	1.610e−3 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ABEBC)	1 p	7.719e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	4.062e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	7.974e−2 (≅)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (<)	6.629e−2 (≅)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (<)	4.614e−2 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)

Table C.27: Comparison between multi-swarm with five sub-swarms and SAMS

DIMENSIONS		2	5	10	30	50
E-STATIC (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	2.127e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-STATIC (Error)	1 p	2.000e+0 (≅)	2.000e+0 (≅)	2.104e−1 (≅)	3.065e−1 (≅)	1.617e−2 (<)
	5 p	2.000e+0 (≅)	2.870e−1 (≅)	5.374e−1 (≅)	2.860e−1 (≅)	1.000e−4 (<)
	15 p	3.233e−1 (≅)	1.000e−4 (<)	1.774e−1 (≅)	2.728e−1 (≅)	1.998e−3 (<)
E-PROGRESS (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (ABEBC)	1 p	5.720e−1 (≅)	2.728e−1 (≅)	2.417e−1 (≅)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	6.884e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-ABRUPT (ITEL)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	3.376e−2 (<)	1.000e−4 (<)	1.090e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (CME)	1 p	1.610e−3 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ABEBC)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
E-CHAOS (ITEL)	1 p	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	5 p	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
	15 p	1.000e−4 (<)	1.000e−4 (<)	2.000e+0 (≅)	2.000e+0 (≅)	2.000e+0 (≅)
E-PATTERN (CME)	1 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	5 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)
	15 p	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)	1.000e−4 (<)

Appendix D

Acronyms

This appendix provides brief descriptions for acronyms and commonly used terms in this thesis. Acronyms are listed alphabetically and typeset in bold, with the meaning of the acronym alongside.

gbest	global best
nbest	neighbourhood best
nbest	personal best
ABEAC	Average Best Error After Change
ABEBC	Average Best Error Before Change
ABFAC	Average Best Fitness After Change
ABFBC	Average Best Fitness Before Change
ADAC	Average Diversity After Change
AI	Artificial Intelligence
APSO	Atomic Particle Swarm Optimization

BKPE	Best Known Peak Error
CI	Computational Intelligence
CME	Collective Mean Error
CMF	Collective Mean Fitness
CPSO	Charged Particle Swarm Optimization
DE	Dynamic Environment
GA	Genetic Algorithm
ITEL	Iteration To Error Limit
LHC	Local Hill Climbing
MPB	Moving Peak Benchmark
PSO	Particle Swarm Optimization
QSO	Quantum Swarm Optimization
SAMS	Self-Adapting Multi-Swarm
SEPSO	PSO model with Spatial Particle Extension

Appendix E

Symbols

This appendix defines all the symbols used throughout the thesis. The symbols are re-defined in the chapters in which they are introduced. A separate list of symbols, ordered alphabetically, is given for the different chapters containing symbols. For each chapter, only newly introduced symbols are defined, or those symbols whose meaning is overloaded.

E.1 Chapter 2: Dynamic Environments

b	peak index
$B(\mathbf{x})$	basis landscape
$dom(x_j)$	the domain of x in dimension j
f	objective function
$\mathcal{F}$	feasible space
$F(\mathbf{x}, t)$	moving peak function
g_m	m^{th} inequality constraint
h_m	m^{th} equality constraint

$\mathcal{I}$	infeasible space
j	dimension index for solution vector
λ	variable determining peak direction
n_g	number of inequality constraints
n_h	number of equality constraints
n_p	number of peaks
n_x	number of dimensions
$\mathcal{N}$	set of feasible points in a neighborhood
$N(0,1)$	Gaussian random variable with 0 mean and a deviation of 1
p_h	peak height
$\mathbf{p_p}$	peak location
$\mathbf{p_r}$	random vector
$\mathbf{p_{v_{\mathit{b}}}}(t)$	shift vector for peak b at time step t
p_w	peak width
P	function defining peak shape
s	moving distance for peak
S	spatial severity
$\mathcal{S}$	search space
σ	random variable $N(0,1)$
t	time step index
T	trajectory circumference
$w(t)$	vector of time dependent control parameters
w_1, w_2	control parameters for shape of function
$\mathbf{x}$	candidate solution
$\mathbf{x}^*(\mathbf{t})$	optimum found at time step t

E.2 Chapter 3: Particle Swarm Optimisation

c_1, c_2	acceleration coefficients
χ	constriction coefficient
i	particle index
κ	constant in constriction model
n_s	number of particles
$\mathbf{r_1}, \mathbf{r_2}$	random vectors
$S.n_s$	number of particles of swarm S
t	iteration index
$\mathbf{v}_i(t)$	velocity of particle i at iteration t
$V_{max,j}$	maximum velocity for dimension j
w	inertia weight
$\mathbf{x}_i(t)$	position of particle i at iteration t
$\mathbf{y}_i$	personal best of particle i
$\hat{\mathbf{y}}_i(t)$	neighbourhood best

E.3 Chapter 4: Particle Swarm Optimisation and Dynamic Environments

$\mathbf{a}_i(t)$	acceleration vector for particle i at iteration t
B_n	quantum cloud
C	current number of sub-swarms
C_{free}	number of free sub-swarms
k, l	swarm indices
M	number of sub-swarms
n_{excess}	maximum number of free sub-swarms

o	particle index
$\mathbf{p}$	swarm attractor
Q_i	charge magnitude of particle i
$r(t)$	dynamic convergence radius
r_{cloud}	quantum radius
r_{excl}	exclusion radius
R_c	core radius
R_p	perception limit
X	domain length

E.4 Chapter 5: Performance Measures for Dynamic Environments

c	iteration during which the last change occurred
e_v	v^{th} evaluation
$err_{c,t}$	best error at iteration t after change c
$f(gbest(t,m))$	fitness of the best particle at iteration t of simulation m
$gbest(t)$	best particle in the swarm at iteration t
$gbest_j(t,m)$	position the $gbest$ in dimension j at iteration t in simulation m
$gworst(t)$	worst particle in the swarm at iteration t
m	simulation index
$max_j(t,m)$	position of the $gbest$ in dimension j at iteration t in simulation m
n_e	number of evaluations
n_k	number of environmental changes
n_m	number of simulations
n_t	number of iterations
r	number of iterations between two changes

v	evaluation index
W	window size
τ	target error

E.5 Chapter 6: Experimental Procedure

$\bar{x}_j(t)$	average positions of the particles in dimension j at time t

E.6 Chapter 8: Evaluation of Swarm algorithms Designed for Dynamic Environments

C_m	maximum number of sub-swarms for simulation m
$C_{m,t}$	number of sub-swarms at iteration t of simulation m

Index