OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INF

FACULTY OF
COMPUTER SCIENCE

CHAIR OF COMPUTATIONAL INTELLIGENCE
AND INTELLIGENT SYSTEMS

Bachelor Thesis in Computer Science

Decision Making Swarms in Dynamic Environments

Examiner:	Prof. Dr. Sanaz Mostaghim
Advisor:	Palina Bartashevich (M.Sc.)
Author:	Doreen Körte
Student-ID:	202924
Submitted:	29th September 2017

Abstract

This work presents a new Navigation Wind PSO (NW-PSO) algorithm as an energy efficient search mechanism in unknown dynamic environments. The environment is simulated by vector fields and represents natural environments as for example wind flows. Inspired by the nature, the vector field influences the particles movement and causes higher energy usage if particles move in the contrary direction of the flow. The goal of NW-PSO is to find an optimal solution and simultaneously save energy during the search process. The concept of NW-PSO is based on Particle Swarm Optimization (PSO) but takes the energy consumption into consideration. In contrast to PSO, the new approach pursues two goals, energy efficiency and qualifying search results. Therefore, a multi-criteria decision making process is used for the movement calculation in order to generate a compromise for both goals. The new approach is compared with two other search mechanisms in different environments and search problems. The first one is Compensative PSO (C-PSO) which works like standard PSO but includes energy computations. The other approach Compensative Wind PSO (CW-PSO) is a modified version of Vector Field Map PSO (VFM-PSO) [BGM17]. Nevertheless, both of them do not consider the energy consumption. The results show that NW-PSO is able to reduce the energy consumption compared with C-PSO and CW-PSO for the considered fitness landscapes and vector fields. The quality of the search and energy results depends on the fitness landscape and the vector field. The obtained results for the proposed model prove that NW-PSO is a suitable search mechanism for energy reduction in dynamic environments.

The proposed approach NW-PSO works as follows: an optimizer swarm and an explorer swarm work together as a multi-swarm. The optimizer swarm searches for optimal solutions in the fitness landscape, whereas the explorer swarm collects the information about the vector field and stores its values into the Information Map. The optimizers needs to make energy efficient movements in order to reduce the energy usage. Therefore, it uses the collected data from the Explorers saved in the Information Map to generate a set of target points. Subsequent, the particle chooses one of the target points and moves there. The decision is based on the energy costs for the movement towards the target point and the improvement of the search. As both objectives need to be optimized, the multi-criteria decision making method Weighted Sum Method is applied.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

C-PSO	Compensative PSO
CW-PSO	Compensative Wind PSO
MCDM	Multi-Criteria-Decision Making
MOO	Multi-Objective Optimization
MPSO	Modified PSO
NW-PSO	Navigation Wind PSO
PSO	Particle Swarm Optimization
SR	Swarm Robotics
UAV	Unmanned aerial vehicle
VFM-PSO	Vector Field Map PSO
WSM	Weighted Sum Method

1. Introduction

Swarm Robotics (SR) is "a new approach to the coordination of multi-robot systems" [SW08]. The main idea is that swarms of robots are able to solve complex tasks by local interaction [BL08]. Typical applications are unmanned aerial vehicle (UAV), cooperative transportation and geological survey. The advantages of swarm robotics compared to single robotics are parallelization, scalability, stability, economical and energy-efficiency [TZ13]. A swarm is able to work in parallel on multiple targets which is time saving. Additionally, it is scalable because the approach is adaptable to different sizes of swarms. Moreover, it guarantees stability because the swarm still executes its task if one robot quits performing. In contrast to a complex single robot, a multi swarm is also cheaper because it consists of multiple simple and small robots. As a consequence, they also consume less energy which is an important factor for mobile robotic systems [TZ13]. Often the robotic systems consist of small and simple mobile robots which cost less in order to The energy results for Rosenbrock function and Ackley function are approximately the same as for Sphere function. Thus, they will not be explained again make large populations affordable[TZ13]. Depending on the type of problem, these robots can either be grounded or aerial. Though, mobile robots have the great disadvantage of a limited battery capacity. Therefore, a good energy management is essential [RZF08]. Another challenge for aerial robots are environmental influences like wind gust disturbances. Therefore, this work focuses on an energy saving search mechanism based on Particle Swarm Optimization (PSO) for unknown dynamic environments. The main idea is that robots reduce their energy usage during the search process by adjusting their movement to the environment.

1.1. State of the Art

Environmental influence on robotic systems is an important topic in recent literature. Especially navigation for UAVs gains in importance as described by Roberge et al. [RTL13]. The authors introduce an automatic path planning algorithm which applies PSO and a genetic algorithm to avoid obstacles and determine optimal trajectories. Another algorithm is presented by Sadiq and Hasan [SH17] which uses the D* algorithm instead of genetic algorithms. Raja and Pugazhenthi [RP09] explain an approach which is able to deal with obstacles with different shapes and varying velocities of the robots. Besides, autonomous underwater vehicles in unknown, whirling environments have been surveyed by Zeng et al. [Zen+12] and Garau et al. [GAO06]. In the year 2005, Zheng et al. [Zhe+05] developed an evolutionary path planner for unmanned air vehicles to deal with unforeseeable environmental changes. Another fully autonomic approach for aerial systems is presented by Sanchez et al. [SPC17]. It includes the A* search mechanism and generates a collision-free path in challenging environments with dynamic obstacles. In 2007, Jatmiko et al. [JSF07] introduced a modified PSO (MPSO) which allows a swarm of robots to detect environmental changes and trace an odour. The aim is to find the source of the odour. One year later, MPSO is extended so that the robots are able to determine the direction of the odours flow and navigate in the opposite direction of the stream to locate the source [Jat+08]. In 2017, Shen et al. [She+17] introduced a bio-inspired method for the control of small unmanned air vehicles. These are able to cope with wind gust disturbances [She+17]. In addition to environments, energy awareness for robots has been brought into focus in latest literature. Moreover, Mei et al. [Mei+04] considered the motion planning of mobile robots. To save energy, different routes and their velocities were compared. Another energy-efficient path planning method was introduced by Ganganath et al. [GCT15]. In contrast, the path planning was made for uneven terrain with the help of a heuristic search mechanism. The energy management of indoor hovering robots which search a predefined target was developed by Roberts et al. [RZF08]. Compared to ground-based robots, the flying robots need more energy. A ceiling attachment was used to be able to use the birds-eye view to improve the search results. Additionally, in 2010 another indoor search algorithm for flying robots was introduced by Stirling et al. [SWF10]. The energy consumption is reduced by local sensing and low-bandwidth communication. Moreover, the total swarm flight time was reduced and energy saved by launching one robot at a time. However, the solving of complex environments or problems can become difficult. Furthermore, Zhou and Kinny [ZK13] offered a bio-inspired energy management for a swarm of robots. Standard PSO is extended so that energy is taken into account in the task selection and motion planning. The main idea is division of labour so that a task which needs a high amount of energy is performed by a robot which also has a high energy level. As a consequence, the energy homogeneity across the swarm can be improved. Although, the energy usage as well as the environmental influences have been important topics in robotics there is no research about energy reduction in relation to the environment.

1.2. Research Goals and Specific Objectives

The goal of this work is to develop a preferably energy efficient search mechanism for dynamic environments. Therefore, vector fields represent the dynamic environments. The proposed model is a modified version of PSO including multi criteria decision making methods in order to choose energy and search efficient movements. The approaches applicability is judged by comparison to other approaches which do not take the energy effort into consideration.

Goal
To develop an energy efficient search mechanism for dynamic environments.

For reaching this goal, the main structure of this thesis can be summarized as the following. First, a model for the particles movement needs to be implemented. This includes a dynamic environment which affects the particles movement. The dynamic environment is represented by a vector field.

Objective 1
To implement a model for the general particles movement in dynamic environments.

Afterwards, an energy consumption model is developed. Therefore, a selection of parameters is estimated which shall be measured. This represents the basis for comparison with existing models. However, a model is only a simplification and does not take into account all parameters which would be important for a real world scenario.

Objective 2
To generate a simplified energy consumption model.

The third objective includes the modification of the standard PSO equation in order to generate an energy saving algorithm. Compared to standard PSO the new approach needs to fulfil two goals. The first one is gaining satisfactory search results. The second goal is the reduction of the energy consumption during the search process. As a result, the approach needs to include a decision making method in order to distinguish the best solution which is a compromise between both goals.

Objective 3
To create a new approach based on PSO which reduces the energy consumption without negligence of the search results quality.

Subsequently, the fourth objective regards the implementation of the different approaches with regard to the created energy model. This includes standard PSO, Vector Field Map PSO (VFM-PSO) and the newly developed approach Navigation Wind PSO (NW-PSO). Furthermore, the approaches are tested in different environments and for different objective functions. The results offer feedback about the type of problem for which each approach is appropriate or not.

Objective 4
To implement the new energy model on other approaches based on PSO and run simulations for all approaches in varying environments.

The last objective considers the comparison of the approaches results. Therefore, the energy consumption and quality of the search results are observed. A conclusion about the applicability of the new designed approach is drawn predicated on the evaluation.

Objective 5

To evaluate the new approach by comparing the simulation results between the different approaches.

1.3. Structural Overview

The following section gives a review of the thesis structure. First, a general background about the main topics of the thesis and later used approaches is introduced in Chapter 2. Afterwards, Chapter 3 presents the proposed model which includes the composition for the following experiments and detailed information about the different approaches that will be compared. Accordingly, the implementation of the fundamental functions is provided by Chapter 4. The experimental setup and parameter settings are described in Chapter 5. This chapter also includes the evaluation of the experimental results. In the end, a conclusion is drawn in Chapter 6. Furthermore, suggestions for future works are given.

2. Background

This chapter presents the background information for the later proposed model and the following experiments. Fundamental information about Particle Swarm Optimization will be provided in Section 2.1. Additionally, Vector Field Map PSO will be explained in Section 2.2 which will be part of the proposed model and the following experiments. Besides, Multi-Criteria Decision Making will be introduced in Section 2.3. Afterwards, the Weighted Sum Method will be explained in Section 2.4. At the end, different boundary handling techniques are introduced in Section 2.5.

2.1. Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm was first introduced by Kennedy and Eberhart [EK95], [KE95] 1995. They developed an optimization process which is inspired by biological structures like ant colonies or swarms of birds. In these populations the individuals interact with each other and share information. As a result, a collective intelligence and a self-organized swarm develops which is able to search for the optima for example in consideration to the food quality. The PSO approach exactly executes this swarm behaviour and optimizes the search results in regard to an objective function [Eng07]. Therefore, a swarm consists of N particles. The basic PSO iteration provides that each particle i is allocated to a position $\vec{x}_i(t+1)$ at time step $t+1$. The particle movement is calculated by the velocity vector $\vec{v}_i(t+1)$ which is added to the current position:

$$\vec{v}_i(t+1) = w\vec{v}_i(t) + C_1\vec{\sigma}_1\left(\vec{P}_{best} - \vec{x}_i(t)\right) + C_2\vec{\sigma}_2\left(\vec{x}_g(t) - \vec{x}_i(t)\right) \tag{2.1}$$

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{2.2}$$

The velocity vector is the main factor of the optimization process. It calculates the next movement depending on the cognitive and social component. The cognitive component, denoted as $\vec{P}_{best}$, is the best found solution of the particle. Whereas the social component represents the best found solution of the whole swarm and is described as $\vec{x}_g(t)$. The influence of each component is determined by the positive constants C_1 and C_2 which are called acceleration coefficients. The vectors $\vec{\sigma}_1$ and $\vec{\sigma}_2$ are two random vectors $\epsilon[0,1]^n$. The old velocity vector $\vec{v}_i(t)$ is weighted by the inertia factor $w > 0$.

2.2. Vector Field Map PSO

The Vector Field Map PSO approach (VFM-PSO) is a modified version of standard PSO (Section 2.1). It is developed for aerial micro-robots in order to perform in unknown environments with uncertain dynamics. The environment is modelled by a two-dimensional vector field which affects the particles movement. As a result, the standard PSO search process is disturbed. VFM-PSO presents a solution to deal with the vector field dynamics based on a multi-swarm [BB04]. This includes an optimizer swarm and an explorer swarm. While the optimizer swarm tries to find the global optimum, the explorer swarm discovers the search space and saves information about the vector field. As the environment is based on a Cartesian Grid, a vector $\overrightarrow{VF}(x_1, x_2)$ for each each point $p = (x_1, x_2)$ can be calculated by interpolation as described in the following:

$$\overrightarrow{VF}(x_1, x_2) = (1 - u_2)(1 - u_1)\overrightarrow{VF}_{0,0} + (1 - u_2)u_1\overrightarrow{VF}_{1,0} + u_2(1 - u_1)\overrightarrow{VF}_{0,1} + u_1u_2\overrightarrow{VF}_{1,1} \tag{2.3}$$

The collected information can be used by the optimizer swarm to calculate movements which counteract the vector field. This movements are generated by a modified PSO equation given below:

$$\vec{x}_i(t + 1) = \vec{x}_i(t) + \vec{v}_i(t + 1) + \sum_{k=0}^{K} \overrightarrow{VF}(p^k) \tag{2.4}$$

As a result, VFM-PSO gains better search results than VF-PSO which applies standard PSO in dynamic environments without corrections [BGM17].

2.3. Multi-Criteria Decision Making

Multi-Criteria Decision Making (MCDM) is a general class for models which are designed for decision problems including multiple criteria. In many cases multiple objective functions need to be optimized. This kind of problem is called Multi-Objective Optimization (MOO) Problem. The main difficulties of their solving is based in the conflicts between the different functions. For example an objective function for the parameter costs and an objective function for benefit are often in conflict with each other. This is due to the trade between profit and costs. The profit should be maximized while the costs shall be minimized at the same time. As a result, there is no unique optimal solution as in mono objective decision making. Instead a set of equally important optima called Pareto Front exists. A point is called optimum "if no criterion can improve the solution without worsening at least one criterion" [LS17]. As shown in Figure 2.1 the Pareto Front consists of a set of non-dominated solutions. They are called non-dominated because they are better than all other solutions and therefore dominate them. Consequently, the other solutions form the dominated set.

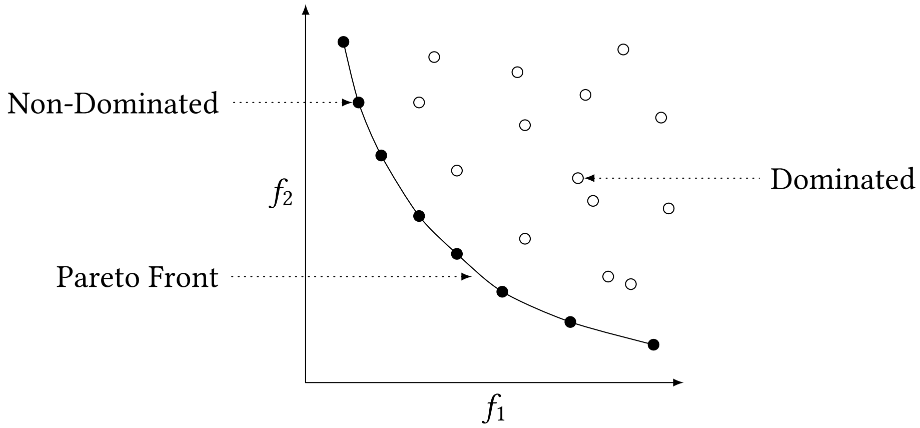


Figure 2.1.: Pareto Front

The objective functions which needs to be optimized are defined as the following.

$$f_1(\vec{x}) \text{ for } i = 1, ...m \text{ and } \vec{x} \epsilon S \tag{2.5}$$

In a minimization problem, a vector $\vec{x}$ is said to be Pareto Optimal if its value is lower than the value of all other vectors $\vec{x} \epsilon S$ for all objective functions f_i for $i = 1, ..., m$. The set of Pareto Optimal solutions is called Pareto Front [Pur+14].

A solving technique to determine a solution out of the Pareto Front is called Aggregation Function Method. Thereby, the multi-objective optimization problem is transformed into a single-objective optimization problem. These problems can be easily solved by estimation of a minimum. One of the used methods is the ϵ-constraint method. However, this method has high computational costs and therefore is not suitable for this thesis. Additionally, the estimation of a good ϵ is difficult because its possible that no solution is found if the selection was unsuitable. Another solving method is goal programming which defines a target value for each objective function and measures the distance of each solution towards this value. Nevertheless, the estimation of a target point is difficult. The most commonly method is the Weighted Sum Method which will be explained in the following Section 2.4 [LS17].

2.4. Weighted Sum Method

The Weighted Sum Method (WSM) is a widely spread method for transforming a multi-dimensional problem into a single-dimensional problem. Thereby, the decision maker is able to display preferences for specific objective functions by a weight vector w_i which is applied in the calculation. WSM is defined as followed:

$$\min f(x) = \sum_{i=1}^{m} w_i f_i(x), \quad w_i >= 0, \quad \sum_{i=1}^{m} w_i = 1 \tag{2.6}$$

where m is the number of objective functions and $f_i(x)$ the corresponding function value of x [LS17]. Usually, WSM is used for single-dimensional problems. This is due to the fact, that different dimensions of the objective functions cannot be summed up. However, multi-dimensional problems can be solved by normalizing the objective functions value. Therefore, the min-max method can be used. The main advantages of WSM are the simplicity and fast computation. Nevertheless, it is not suitable for concave Pareto Fronts [GR06].

2.5. Reflected Bound Handling

Bound handling is an important topic in PSO because it has great influence on the search results especially if the optimum is near the bound. In general, optimization problems have a bounded search space. However, the particles are able to leaf the feasible space. Therefore, it is important to apply bound handling techniques for guaranteeing that the particles stay inside the bound. Otherwise they would not be able to find the optimal solution. One of the techniques is Reflected Bound Handling "where the infeasible solution is reflected in the feasible space" [HBM13] as shown in Figure 2.2.

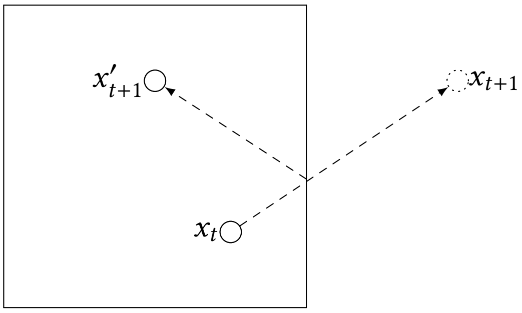


Figure 2.2.: Reflected Bound Handling

The calculation of the particles new position is given below. The boundaries of the search space are denoted as x_{min} and x_{max}.

$$if\ x_{t+1} > x_{max},\ then\ x'_{t+1} = x_{max} - (x_{t+1} - x_{max}) \tag{2.7}$$

$$if\ x_{t+1} < x_{min},\ then\ x'_{t+1} = x_{min} + (x_{min} - x_{t+1}) \tag{2.8}$$

3. Proposed Model

This chapter describes the model for the following experiment. It includes three different optimization swarms using modified versions of PSO which will be explained in Section 3.3. Apart from the optimization swarms a concept of a swarm called Explorers (Section 3.3.1) is introduced. Its main task is the improvement of the search results of the Compensative Wind PSO (CW-PSO) (Section 3.3.3) and Navigation Wind PSO (NW-PSO) (Section 3.3.4) by collecting information about the vector field. Furthermore assumptions and constraints are settled in Section 3.1. Afterwards the energy calculation is presented in Section 3.2.

3.1. Assumptions and Constraints

First, the assumption is made that the individuals have an alignment and can only move forward. If they want to move left they first have to change their arrangement by twisting and secondly move forwards. Additionally they can only move by a limited velocity $vMax$. If their velocity vector gets larger than this threshold it is set to $vMax$. The focus of the energy calculation is laid on the individuals movement actions. Besides the model provides that each individual uses a battery which is set to a predefined start value at the beginning and discharges by moving and turning. If the battery is entirely empty, the individual cannot move or turn any more and is defined as dead. Moreover, particles still use energy if they do not show movements. This is due to the fact that they need to compensate the vector field in order to stay at one position. If the particles would not apply any energy they could not stay at this position and would be blown away like the Explorers. Furthermore, all weights which are used for the decision making process in Section 3.3.4 are constant for the whole model and derived by experiments. They can be found in Table A.1. To sum up, the assumption is made that the particles of all swarms except the CW-PSO swarm, can estimate the vector of the vector field at its local position.

3.2. Energy Calculation

The energy consumption is measured for each individual i inside a population p and for the population itself. It is based on two parameters: moving and turning. The parameters are illustrated by the following Figures.

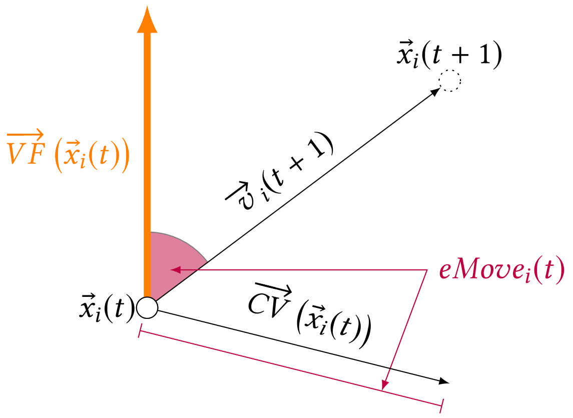

$\vec{x}_i(t+1)$
$\overrightarrow{VF}\left(\vec{x}_i(t)\right)$
$\vec{v}_i(t+1)$
$eMove_i(t)$
$\vec{x}_i(t)$
$\overrightarrow{CV}\left(\vec{x}_i(t)\right)$

Figure 3.1.: Energy Calculation $eMove_i(t)$

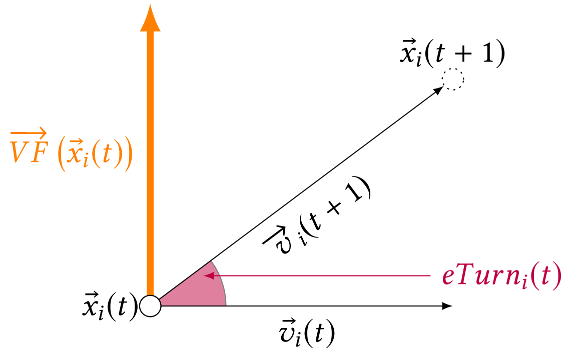


Figure 3.2.: Energy Calculation $eTurn_i(t)$

At the beginning, each individual i has an equal amount of energy which is defined by an initial value $e_{init} > 0$. The battery for each individual is set to this initial value. Afterwards, the battery value $battery_i(t+1)$ for the time step $t+1$ is defined by the battery value $battery_i(t)$ and the spend energy $eTotal_i(t)$ at the current time step t:

$$battery_i(t+1) = battery_i(t) - eTotal_i(t) \tag{3.1}$$

The total energy $eTotal_i(t)$ represents the energy consumed by one individual during one time step t for moving and turning. It is calculated as the following:

$$eTotal_i(t) = eMove_i(t) + eTurn_i(t) \tag{3.2}$$

$$eMove_i(t) = |\overrightarrow{CV}\left(\vec{x}_i(t)\right)| + angle_i(t) \cdot |\overrightarrow{CV}\left(\vec{x}_i(t)\right)|, \quad -2 < angle_i(t) < 2 \tag{3.3}$$

$$angle_i(t) = \frac{\left(\arccos\left(\frac{\vec{VF}\left(\vec{x}_i(t)\right)\cdot\vec{v}_i(t+1)}{|\vec{VF}\left(\vec{x}_i(t)\right)\cdot\vec{v}_i(t+1)|}\right)\right) - 90}{45} \tag{3.4}$$

$$eTurn_i(t) = \arccos\left(\frac{\vec{v}_i(t) \cdot \vec{v}_i(t+1)}{|\vec{v}_i(t)| \cdot |\vec{v}_i(t+1)|}\right) \tag{3.5}$$

The first summand $eMove_i(t)$ is the energy for moving, which is defined by the correction vector $\overrightarrow{CV}(x_i(t))$ which will be explained in the following Section and the angle between the velocity vector $\vec{v}_i(t)$ and the vector of the vector field $\overrightarrow{VF}(\vec{x}_i(t))$. The calculation of $angle_i(t)$ provides the possibility of energy reduction if the individual moves in the same direction as the vector of the vector field. In contrast, the individual needs more energy if it moves in the contrary direction. Thus, the compensation of the vector field and therefore spent energy is estimated. The second summand $eTurn_i(t)$ is the energy effort for turning which is set by the angle between the old velocity vector $\vec{v}_i(t)$ and the new velocity vector $\vec{v}_i(t+1)$. Therefore, the individuals rotation during one time step is calculated. At the end of each time step t, the energy for the whole population p is calculated as the following:

$$e_p(t) = \sum_{i=0}^{N} battery_i(t) \tag{3.6}$$

Where N is the number of individuals inside the population p and $battery_i$ is the battery value of the respective individual i. Admittedly, energy calculations are only made for the optimization swarms because the Explorers do not spend any energy for moving. Due to the fact, that they are driven by the vectors of the vector field and therefore do only passive movements.

3.3. Swarm Operation Methods

The model contains four kinds of swarms. These are Explorers, Compensative PSO (C-PSO), Compensative Wind PSO (CW-PSO) and Navigation Wind PSO (NW-PSO) which will be explained in the following sections. The particle velocity of each population is influenced by the vector field which is explained in Section 2.2. The Explorer swarm is the only one which is not doing any optimization. Its main task is to improve the search results of the CW-PSO (Section 3.3.3) and NW-PSO (Section 3.3.4) by collecting information about the vector field. The other swarms apply PSO (Section 2.1) in order to discover the search space and find the optimal solution. As the vector field influences the particles position and movement in each swarm, the introduced PSO velocity equation (Equation 2.1) is modified as given below:

$$\vec{v}_i(t+1) = \overrightarrow{DV}\left(\vec{x}_i(t)\right) + \overrightarrow{VF}\left(\vec{x}_i(t)\right) \tag{3.7}$$

$$\overrightarrow{DV}\left(\vec{x}_i(t)\right) = \overrightarrow{PSO}\left(\vec{x}_i(t)\right) + \overrightarrow{CV}\left(\vec{x}_i(t)\right) \tag{3.8}$$

$$\overrightarrow{PSO}\left(\vec{x}_i(t)\right) = w\vec{v}_i(t) + C_1\sigma_1\left(\vec{P}_{best} - \vec{x}_i(t)\right) + c_2\sigma_2\left(\vec{x}_g(t) - \vec{x}_i(t)\right) \tag{3.9}$$

The example of calculation is illustrated for the Compensative PSO in Figure 3.3. The velocity is computed by the summing up of a direction vector $\overrightarrow{DV}\left(\vec{x}_i(t)\right)$ which determines the particles intended movement and the vector of the vector field $\overrightarrow{VF}\left(\vec{x}_i(t)\right)$ at the particles position. The value for $\overrightarrow{VF}\left(\vec{x}_i(t)\right)$ is determined by interpolation explained in Section 2.2. The direction vector $\overrightarrow{DV}\left(\vec{x}_i(t)\right)$ is estimated by the sum of $\overrightarrow{PSO}\left(\vec{x}_i(t)\right)$ known as the old PSO velocity equation (Equation 2.1) and a correction vector $\overrightarrow{CV}\left(\vec{x}_i(t)\right)$. This correction vector is generated differently for each optimization swarm and will be explained below. It is the main difference between the swarms algorithms and the divergent performances.

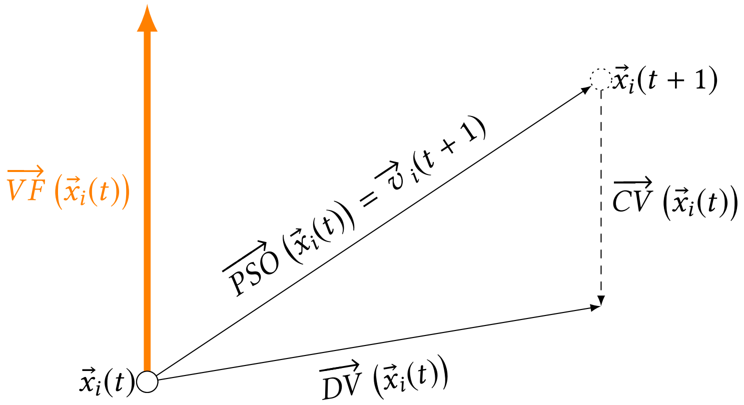

$\vec{x}_i(t+1)$
$\overrightarrow{VF}\left(\vec{x}_i(t)\right)$
$\overrightarrow{PSO}\left(\vec{x}_i(t)\right) = \vec{v}_i(t+1)$
$\overrightarrow{CV}\left(\vec{x}_i(t)\right)$
$\vec{x}_i(t)$
$\overrightarrow{DV}\left(\vec{x}_i(t)\right)$

Figure 3.3.: C-PSO Correction Vector Estimation

3.3.1. Explorers

Since the simulation takes place in an unknown environment, the swarms have no information about the vector field. Therefore, the task of the Explorer Swarm is to investigate the surrounding area. For this purpose, the Explorers move by the vectors of the vector field described by the following equation.

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \frac{\overrightarrow{VF}\left(\vec{x}_i(t)\right)}{|\overrightarrow{VF}\left(\vec{x}_i(t)\right)|} \tag{3.10}$$

Where $\overrightarrow{VF}\left(\vec{x}_i(t)\right)$ is the vector of the vector field at position $\vec{x}_i(t)$. The vector fo the vector field is normalized by its magnitude $|\overrightarrow{VF}\left(\vec{x}_i(t)\right)|$ so that the particles cannot move further than one step at each iteration. This generates a continuous movement and a better exploration at the end. Given that each particle of the Explorer Swarm can measure the vector at its own position, the swarm simply saves this information in a matrix called Information Map. The collected data is used for the CW-PSO (Section 3.3.3) and NW-PSO (Section 3.3.4) for better performances.

3.3.2. C-PSO

The idea of the Compensative PSO (C-PSO) population is, that particles move directly towards the current optimum without any distraction by the vector field (Figure 3.3). This results in a good convergence, however the energy consumption is neglected. The movement is realized by a correction vector which deals with the interference of the vector field:

$$\overrightarrow{CV}\left(\vec{x}_i(t)\right) = -\overrightarrow{VF}\left(\vec{x}_i(t)\right) \tag{3.11}$$

3.3.3. CW-PSO

The main concept of the Compensative Wind PSO (CW-PSO) is reaching the global optimum by taking advantage of the Information Map. The approach is explained in Section 2.2. Since the particles of the CW-PSO swarm cannot generate the value of the vector field at their local position, they are to a greater or lesser extend depending on the Information Map. Therefore, the CW-PSO algorithm distinguishes two cases for calculating the correction vector. The first case occurs if the vector at the current position is saved in the Information Map. In this case, the swarm behaves the same as the C-PSO swarm (Section 3.3.2). The correction vector is calculated by the following equation.

$$Case1 : \overrightarrow{CV}\left(\vec{x}_i(t)\right) = -\overrightarrow{VF}\left(\vec{x}_i(t)\right) \tag{3.12}$$

In the other case, the vector at the particles position is not stored in the Information Map. As a result the correction vector cannot be calculated by the negative value of the vector field so that the correction vector is set to zero:

$$Case2 : \overrightarrow{CV}\left(\vec{x}_i(t)\right) = 0 \tag{3.13}$$

As a consequence, the particles velocity is highly influenced by the unknown value of the vector field. The particle will probably be redirected (Figure 3.4).

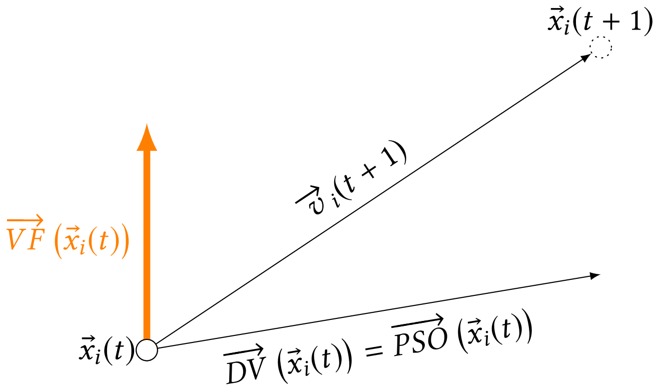

$\vec{x}_i(t+1)$
$\vec{v}_i(t+1)$
$\overrightarrow{VF}\left(\vec{x}_i(t)\right)$
$\vec{x}_i(t)$
$\overrightarrow{DV}\left(\vec{x}_i(t)\right) = \overrightarrow{PSO}\left(\vec{x}_i(t)\right)$

Figure 3.4.: CWPSO Correction Vector Estimation Case 2

3.3.4. NW-PSO

This section is about Navigation Wind PSO (NW-PSO) which is as well a search mechanism based on PSO (Section 2.1). Its focus lies on the energy consumption during the search in unknown vector fields. The goal is to spend as little energy as possible for finding the optimal solution. This is realized by forming a multi-swarm [BB04] with an Explorer Population (Section 3.3.1). The NW-PSO swarm can use the information collected by the Explorer swarm to improve the navigation in the vector field and reduce the energy usage. The main aspects of the algorithm are listed in Figure 3.5 and will be explained exemplary for one particle of the NW-PSO population.

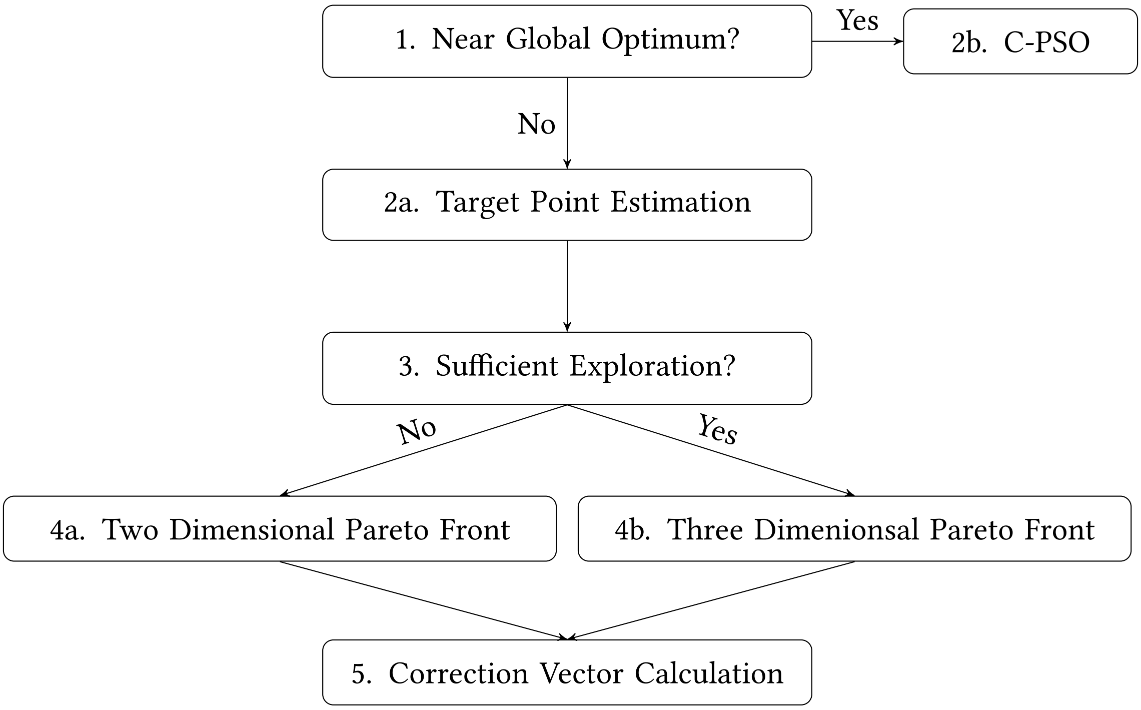
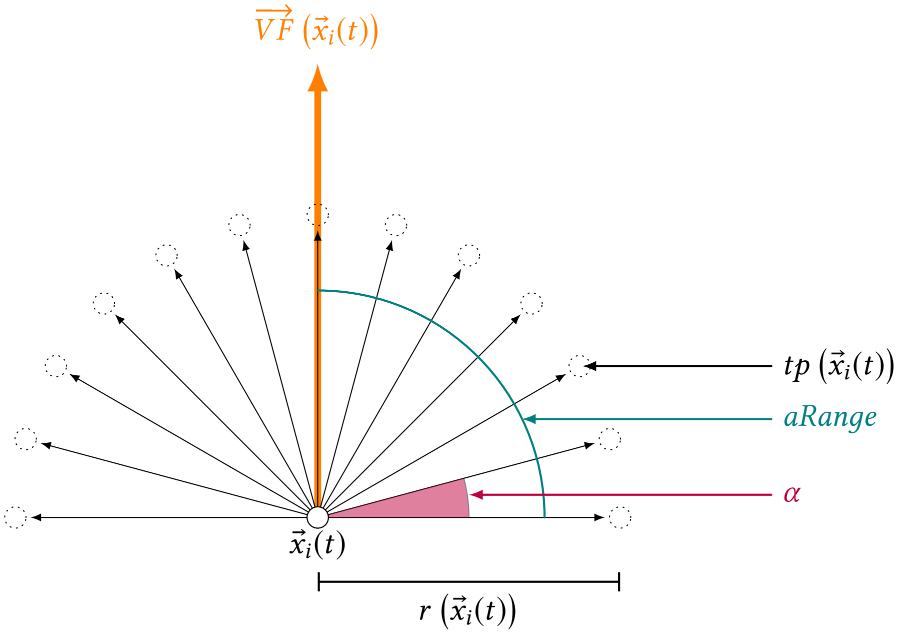

1. Near Global Optimum?
Yes
2b. C-PSO
No
2a. Target Point Estimation
3. Sufficient Exploration?
No
Yes
4a. Two Dimensional Pareto Front
4b. Three Dimenionsal Pareto Front
5. Correction Vector Calculation

Figure 3.5.: NW-PSO Algorithm

First, the distance towards the global optimum is calculated. If it is smaller than the predefined search radius, the particle moves directly towards this optimum by calculating a correction vector explained in Section 3.3.2. As a result, the particles are able to converge at the optimum. In the other case, the particles need to navigate towards the global optimum. Therefore the basic concept of the mechanism is to find an energy efficient target point $\overrightarrow{tp}\left(\vec{x}_i(t)\right)$ at each time step t. Energy efficient means the movement towards this point consumes only small energy. Additionally, it is expected that the next movement starting at this target point will also be energy efficient. Another important factor for the selection of the target point is the quality in reference to the search. The target point must also lead to satisfying search results.

At first a collection of possible target points must be extracted and evaluated. The selection of the target points is illustrated in Figure 3.6.



$\overrightarrow{VF}\left(\vec{x}_i(t)\right)$
$tp\left(\vec{x}_i(t)\right)$
$aRange$
α
$\vec{x}_i(t)$
$r\left(\vec{x}_i(t)\right)$

Figure 3.6.: NW-PSO Target Points Estimation

Given that each particle can only move with a limited velocity (Section 3.1) the target points should not be further than this restriction. Therefore the search radius $r\left(\vec{x}_i(t)\right)$ (Figure 3.6) is set to the maximum velocity $vMax$. Following the definition of the search radius, a vector similar to the vector $\overrightarrow{VF}\left(\vec{x}_i(t)\right)$ of the vector field at the particles position $\vec{x}_i(t)$ is generated. Its length is set to the predefined search radius. This vector is stepwise rotated anticlockwise and clockwise by a given angle α. This angle is calculated by the division of a pre-set angle range $aRange$ and precision $prec$. The angle range determines in which range the vector will be rotated. Besides, it defines how similar the later movement $\vec{v}_i(t+1)$ will be to the vector $\overrightarrow{VF}\left(\vec{x}_i(t)\right)$. This similarity is a crucial factor for the energy consume (Section 3.2). A smaller range means a higher affinity and less energy usage. The precision defines how many vectors are calculated within the angle range. The smaller its value, the more vectors are calculated and the more precise the results will be. If the computational costs become too high, this parameter can be set higher for lowering the precision. The set $TP = \{\overrightarrow{tp_1}\left(\vec{x}_i(t)\right), \overrightarrow{tp_2}\left(\vec{x}_i(t)\right), ..., \overrightarrow{tp_n}\left(\vec{x}_i(t)\right)\}$ of n target points is placed at the end of each calculated vector.

The second task is the identification of the best target point out of this set. Therefore the target points are arranged in a Pareto Front (Section 2.3) and evaluated by appliance of WSM (Section 2.3). Certainly the parameter selection for the Pareto Front is divided into two cases depending on the Information Map (Section 3.3.1). The decision between the cases is generated with the help of the target points. As already mentioned, the Information Map includes collected data about the vector field but only the vectors of these positions are added where

the particles of the Explorer Swarm came across. As a result, the information is incomplete and not all vectors of all positions are saved in the map. The position of each target point is checked in the Information Map and whether it contains a value or not. If the vector of half of the target points or more is known, the first case occurs. This means that there are enough explored fields and saved in the map. As a result, the values of the map can improve the particles navigation. Otherwise the second case occurs which means that not enough vectors are known and the results might be misleading. Both will be explained in detail in the following. The first case offers more decision parameters due to the fact that the vector at most of the target points is known. Thus the Pareto Front will be three-dimensional.
The first parameter is the angle between the vector $\overrightarrow{VF}\left(\vec{x}_i(t)\right)$ of the vector field at the particle position $\vec{x}_i(t)$ and the vector towards the target point $\overrightarrow{tp}\left(\vec{x}_i(t)\right)$ (Figure 3.7). The other two parameters require the vector $\overrightarrow{VF}\left(\overrightarrow{tp}\left(\vec{x}_i(t)\right)\right)$ from the Information Map. The second parameter is defined by the distance between the vector $\overrightarrow{VF}\left(\overrightarrow{tp}\left(\vec{x}_i(t)\right)\right)$ of the vector field and the position of the global best $P_{best}(t)$ (Figure 3.7). The last parameter set by the angle between $\overrightarrow{VF}\left(\overrightarrow{tp}\left(\vec{x}_i(t)\right)\right)$ and the vector from the target point towards the global best (Figure 3.7). Altogether the parameters offer a good decision making and navigation for the particle, because the vector at the target position is considered. This influences the decision making at the next time step.

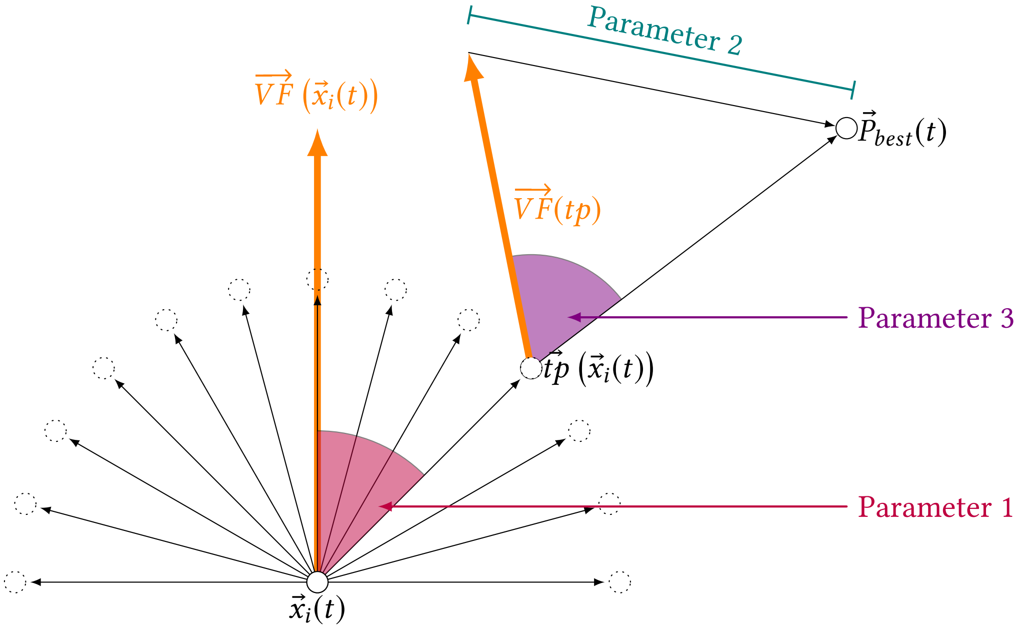

Parameter 2
$\overrightarrow{VF}\left(\vec{x}_i(t)\right)$
$\vec{P}_{best}(t)$
$\overrightarrow{VF}(tp)$
Parameter 3
$\overrightarrow{tp}\left(\vec{x}_i(t)\right)$
Parameter 1
$\vec{x}_i(t)$

Figure 3.7.: NWPSO Parameter Estimation Case 1

In the second case, there are not enough known vectors at the target points. As a result there is less information used for the decision making process. A two-dimensional Pareto Front is generated. The first parameter is generated by the same calculation as for the three-dimensional Pareto Front mentioned before. Though, the second vector is the distance between the position of the target point $\overrightarrow{tp}\left(\vec{x}_i(t)\right)$ and the position of the global best $\vec{P}_{best}(t)$ as shown in Figure

3.8. In contrast to the other case, this one results in worse navigation results, because the vector field at the target point cannot be observed. Nevertheless it is energy-saving.

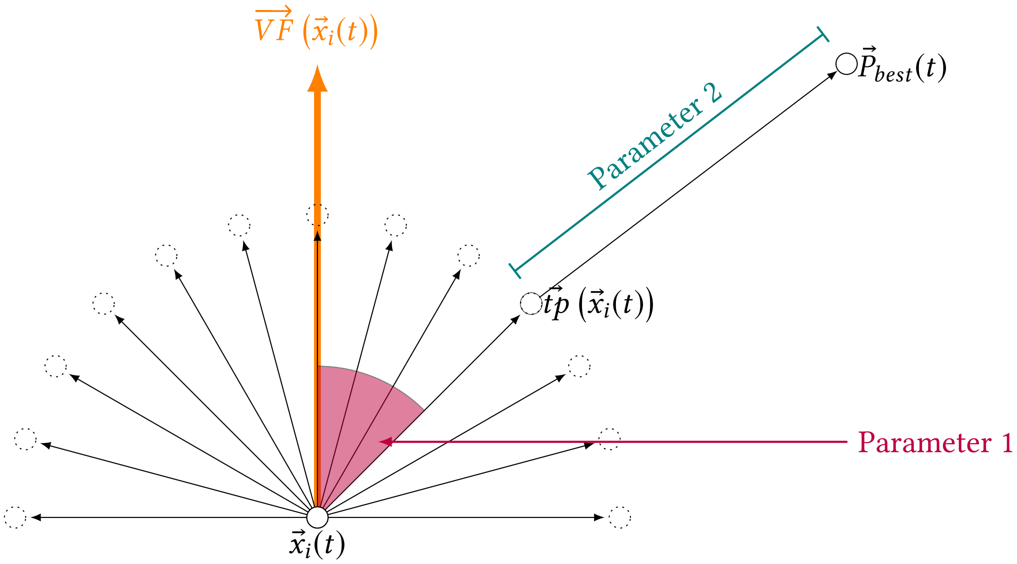

$\overrightarrow{VF}\left(\vec{x}_i(t)\right)$
$\vec{P}_{best}(t)$
Parameter 2
$\vec{tp}\left(\vec{x}_i(t)\right)$
Parameter 1
$\vec{x}_i(t)$

Figure 3.8.: NWPSO Parameter Estimation Case 2

In the end, the correction vector $\overrightarrow{CV}\,(x_i(t))$ is calculated by the following equation:

$$\vec{P}\left(\vec{x}_i(t)\right) = \vec{tp}\left(\vec{x}_i(t)\right) - \vec{x}_i(t) \tag{3.14}$$

$$\overrightarrow{CV}\left(\vec{x}_i(t)\right) = -\overrightarrow{PSO}\left(\vec{x}_i t(t)\right) - \overrightarrow{VF}\left(\vec{x}_i(t)\right) + \vec{P}\left(\vec{x}_i(t)\right)\big) \tag{3.15}$$

4. Implementation

This chapter explains the simulation of the proposed model in Chapter 3. For comparison of the three approaches, a simulation environment was compiled using MATLAB 2016a. The aim of the simulation is to run comparable experiments for different objective functions and vector fields. The environment is shown exemplary in Figure 4.1 which illustrates the simulation of Sphere function for vector field *Rotation* at iteration 30. The results of the simulation will be evaluated in Chapter 5.

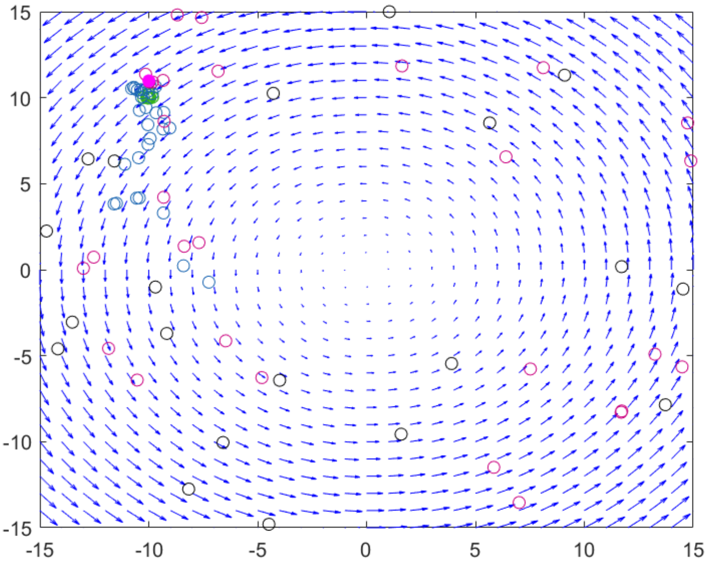


Figure 4.1.: Simulation of Sphere Function for Vector Field Rotation at iteration 30

As the simulation includes stochastical parameters, it is executed for 100 times. In the following, the four main algorithms of the simulation are proposed. First, the general simulation function will be explained. Afterwards, the Explorers calculation will be presented. Section 4.3 deals with the realisation of the optimizer swarms. At the end, the concrete implementation of the NW-PSO approach is described.

4.1. Simulation

The whole simulation is started by running Algorithm 1 which is explained below. The number of runs for the simulations and iterations can be set by the associated variables. Whereby, each simulation consists of a predefined number of iterations and presents one passage. The function *initialise* uses the pre-set parameters to generate the configuration. This includes the generation of the swarms and the search space. The particles are randomly placed whereby their positions are identical for each swarm. Additionally, the configuration includes information about the selected vector field and objective function. Subsequently, the performance of the swarms is defined. The optimizer swarm movement is calculated by the function *move* which is explained in Section 4.3. As the Explorer swarm differs from the other its movement is generated by the function *explore* which will be explained in the following section.

Algorithm 1: Run Simulation

```
Result: Calculates the simulation
for j < simulations do
    config ← initialise( parameters );
    for i < iterations do
        cpso ← move( cpso );
        cwpso ← move( cwpso );
        nwpso ← move( nwpso );
        informationMap ← explore( informationMap );
    end
end
```

4.2. Explorers

The Explorers movement and behaviour is implemented by Algorithm 2 which is illustrated at the end of the section. At each iteration each particle of the explorer swarm updates the Information Map. This is realized by the function *updateMap* which saves the vector of the vector field at the particle position in the map. Afterwards, the particle movement is generated by the scaled velocity of the vector field as explained in Section 3.3.1. At the end, the particle position is updated using Reflected Bound Handling (Section 2.5).

Algorithm 2: Explore

```
Result: Calculates the movement of the Explorers and their exploration
for i < swarmSize do
    e = explorers( i );
    $\overrightarrow{VF}$ ← getVectorField(e.position);
    informationMap ← updateMap( $\overrightarrow{VF}$);
    newVelocity ← scale( $\overrightarrow{VF}$ );
    e.position ← boundaryHandling( e.position, newVelocity );
end
```

4.3. Optimization Swarms

The performance of the optimizer swarms is realized by Algorithm 3 which is shown below. The input of the algorithm is the swarm for which it generates the particles movement and energy consumptions. First, the algorithm proofs whether the particle is dead or not. This is due to the fact that no more movement is possible if the particles have no energy left. If the particle is alive the new velocity is calculated as described in Chapter 3. Depending on the kind of swarm different correction vectors are used. If the new velocity is larger than the maximum velocity, it is reduced to the value of the maximum velocity. The functions *getMoveEnergy* and *getTurnEnergy* calculate the energy consumption for the new velocity as described in Section 3.2. Subsequent, the battery value and live status of the particle are updated. The function *boundaryHandling* calculates the new position according to Reflected Bound Handling (Section 2.5). At the end, the global best found solution is updated for PSO.

Algorithm 3: Move

```
Result: Calculates new position and energy usage for each particle and updates global best.
for i < swarmSize do
    p = swarm.particle(i);
    if p.isAlive then
        $\overrightarrow{PSO}$ ← getPso(p);
        $\overrightarrow{VF}$ ← getVectorField(p.position);
        $\overrightarrow{CV}$ ← getCorrection(p, swarm, $\overrightarrow{VF}$, $\overrightarrow{PSO}$);
        oldVelocity = p.velocity;
        newVelocity = $\overrightarrow{PSO} + \overrightarrow{CV} + \overrightarrow{VF}$;
        if newVelocity > vMax then
            newVelocity = vMax;
        end
        p.velocity = newVelocity;
        $eMove$ ← getMoveEnergy( $\overrightarrow{VF}$, newVelocity );
        $eTurn$ ← getTurnEnergy( oldVelocity, newVelocity );
        $eTotal = eMove + eTurn$;
        p.battery = p.battery - $eTotal$;
        if p.battery <= 0 then
            p.isAlive = false;
        end
        p.position ← boundaryHandling( p.position, newVelocity );
        p.value ← getFunctionValue( p.position );
        if p.value <= swarm.best then
            swarm.best ← updateBest (p.value, p.position);
        end
        swarm.particle(i) = p;
    end
end
```

4.4. NW-PSO

The following algorithm 4 at the end of the section describes the calculation of the correction vector for the NW-PSO particles. At first, the distance from the particle position towards the global optimum is determined. If it is smaller than the maximum velocity, the correction vector is set to the vector of the vector field. As a result, the particle can converge to the global optimum. In the second case, the particle is too far away from the global optimum and needs to navigate to reach it. Therefore, the best target point inside a search radius is defined as explained in Section 3.3.4. At the beginning, the search radius is generated by the normalization of the vector of the vector field. Afterwards, the target points are estimated with the help of a rotation vector. The function *decisionMakingNWPSO* generates the best target point. Its functionality is explained in Algorithm 5 given in this section. Finally, the correction vector is calculated by the selected target point.

Algorithm 4: Get Correction Vector NW-PSO

```
Result: Calculates correction vector for NW-PSO algorithm
Output: Correction vector for NW-PSO
distanceToGlobalBest ← normalize( localPosition - globalBestPosition );
if distanceToGlobalBest < $VMax$ then
    $\overrightarrow{CV} = -\overrightarrow{VF}$;
end
else
    radius ← normalize( $\overrightarrow{VF}$ );
    if radius > maxVelocity then
        radius = maxVelocity;
    end
    rotationVector = radius * ( $\overrightarrow{VF}$ / normalize( $\overrightarrow{VF}$ ) );
    targetPoints ← getTargetPoints( rotationVector );
    targetPointVectors ←getTargetVectors( targetPoints, localPosition );
    bestTargetPoint ← decisionMakingNWPSO( targetPoints );
    $\overrightarrow{CV}$ = bestTargetPoint - localPosition;
end
```

The decision making process is performed by Algorithm 5. Therefore, WSM from Section 2.4 is implemented. First, the parameters of the target points as described in Section 3.3.4 are given as input variables f1, f2, and f3. The array f1 includes the first parameter values for each target point, f2 includes the second parameter values and f3 the third parameter values. Depending on the number of known wind the decision process contains two or three variables. At the beginning, each array is normalized by using the min-max method. Subsequent, a Pareto Front (Section 2.3) is generated by estimating the non-dominated set. Afterwards, a value is generated for each target point of this set. This is made by application of the predefined weights from Table A.1. At the end, the target point with the best value is chosen.

Algorithm 5: Decision Making NW-PSO

```
Result: Selects best target point in pareto front
Output: index of best target point
if number of known wind >= minimum number of known wind then
    f1Utop, f1Nadir ← getMinimumAndMaximum( f1 ); f2Utop, f2Nadir ← getMinimumAndMaximum( f2 ); f3Utop, f3Nadir ← getMinimumAndMaximum( f3 );
    for i < size do
        f1( i ) = ( f1( i ) - f1Utop ) / ( f1Nadir - f1Utop );
        f2( i ) = ( f2( i ) - f2Utop ) / ( f2Nadir - f2Utop );
        f3( i ) = ( f3( i ) - f3Utop ) / ( f3Nadir - f3Utop );
    end
    bestValue = inf;
    bestIndex = 0; for i < size do
        isDominated = false;
        for j< size do
            if j!=i AND f1(j) < f1(i) AND f2(j) < f2(i) AND f3(j) < f3(i) then
                isDominated = true;
            end
        end
        if !isDominated then
            value = wf1 * f1(i) + wf2 * f2(i) + wf3 * f3(i);
            if value < bestValue then
                bestValue = value;
                bestIndex = i;
            end
        end
    end
else
    f1Utop, f1Nadir ← getMinimumAndMaximum( f1 ); f2Utop, f2Nadir ← getMinimumAndMaximum( f2 );
    for i < size do
        f1( i ) = ( f1( i ) - f1Utop ) / ( f1Nadir - f1Utop );
        f2( i ) = ( f2( i ) - f2Utop ) / ( f2Nadir - f2Utop );
    end
    bestValue = inf;
    bestIndex = 0; for i < size do
        isDominated = false;
        for j< size do
            if j!=i AND f1(j) < f1(i) AND f2(j) < f2(i) then
                isDominated = true;
            end
        end
        if !isDominated then
            value = wf1 * f1(i) + wf2 * f2(i);
            if value < bestValue then
                bestValue = value;
                bestIndex = i;
            end
        end
    end
end
```

5. Evaluation

The proposed model (Section 3.3) includes three different approaches for the search behaviour of swarms in an unknown environment under uncertain dynamic influences. Their applicability is evaluated in consideration of the success rate, convergence and energy consumption during the search process. Therefore, the goal is to determine which one of the approaches is suitable for the selected environment. First, the experimental setup will be introduced in Section 5.1. Subsequent, in Section 5.2 the results of the experiments will be analyzed. At the end, a summary of the results for each approach is provided.

5.1. Experiments

This section deals with the experimental environment which offers the framework for the simulations. At the beginning, all particles are randomly placed in the two-dimensional search space. The applied parameters are given in Table 5.1.

Table 5.1.: Parameter Setting

Description	Parameter	Value
Search Area		
Search space	d	2
Grid width	X	$[-15, 15]$
Grid length	Y	$[-15, 15]$
Optimum	p	0 ± 0.01
Optimum position	$\overrightarrow{opt}$	$[-10, 10]$
PSO Parameters		
Population size	N	30
Inertia weight	w	0.6
Acceleration coefficients	C_1, C_2	1.0
Iterations	I	100
Swarm Parameters		
Maximum velocity	$vMax$	2
Number of Explorers	M	20
Initial battery	$battery_{init}$	100
Angle range	$aRange$	90
Precision	$prec$	2

All approaches are tested on three different objective functions for 100 simulations, each consisting of 100 iterations. At the beginning of each simulation the particles are rearranged randomly in the search space and have no information the position of the optimum. The global optimum of each function is zero. The optimal position is set to $[-10, 10]$ for better analysis. The first function is Sphere which is the simplest of all due to the fact that there are no local minima except the global optimum. The Sphere function is defined as:

$$f(x) = \sum_{i=1}^{|X|} (x_i)^2, \quad x \in X \tag{5.1}$$

The second function is Rosenbrock, which possesses a narrow, parabolic valley in which the optimum needs to be found. The equation is as follows:

$$f(x) = \sum_{i=1}^{|X|-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2], \quad x \in X \tag{5.2}$$

The last function is Ackley, which provides a lot of local minima which might be misleading the swarms. The equation is described in the following:

$$f(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{|X|}\sum_{i=1}^{d} x_i^2} - \exp\left(\frac{1}{d}\sum_{i=1}^{d} \cos(\frac{1}{2}\pi x_i)\right)\right) + a + e, \quad x \in X \tag{5.3}$$

Since the experiment provides results in dynamic environments, each of the objective functions is tested in nine different vector fields (Appendix B.1). The corresponding functions are described in Table 5.2.

Table 5.2.: Vector Field Functions

Description	Title	Function
"Cross"	VF1	$\overrightarrow{VF}_1(x_1, x_2) = (x_2, x_1)$
"Rotation"	VF2	$\overrightarrow{VF}_2(x_1, x_2) = (-x_2, x_1)$
"Sheared"	VF3	$\overrightarrow{VF}_3(x_1, x_2) = (x_1 + x_2, x_2)$
"Wave"	VF4	$\overrightarrow{VF}_4(x_1, x_2) = (-\sin(x_2)), \cos(x_1 \cdot x_2 - x_1^2)$
"Tornado"	VF5	$\overrightarrow{VF}_5(x_1, x_2) = (-x_1 - x_2, x_1)$
"Bi-Directional"	VF6	$\overrightarrow{VF}_6(x_1, x_2) = (-0.7 \cdot x_2, 0)$
"Random"	VF7	$\overrightarrow{VF}_7(x_1, x_2) = (rand[-1, 1], rand[-1, 1])$
"Multi-Rotation"	VF8	$\overrightarrow{VF}_8(x_1, x_2) = \left(\sin\left(\frac{4}{30}\right) \cdot 2\pi x_2, \sin\left(\frac{4}{30}\right) \cdot 2\pi x_1\right)$
"Vortex"	VF9	$\overrightarrow{VF}_9(x_1, x_2) = \left(\cos\left(\frac{4}{30}\right) \cdot (x_1 + 2x_2), \sin\left(\frac{4}{30}\right)(x_1 - 2x_2)\right)$

5.2. Results

This section is about the analysis of the experimental results. For this purpose, the convergence of each swarm is analyzed. Afterwards, the energy consumption is examined. Additionally, the success rates are regarded in order to determine the chances of success for each swarm. In the end, the results of the analysis are summarized in Section 5.3.

5.2.1. Convergence Analysis

This section deals with the convergence analysis for the three optimization swarms mentioned in Chapter 3. For this purpose, the convergence plots are proposed. They illustrate the convergence rate for each optimization swarm. The convergence rate displays the value of the average best found solution of the swarm at each iteration. Additionally, the awareness rate for NW-PSO and CW-PSO is shown. The awareness rate for NW-PSO illustrates how many particles percentile inside one iteration used the Information Map and calculated a three-dimensional Pareto Front. The awareness rate for CW-PSO displays how many particles calculated a correction vector during each iteration and thereby used the Information Map. Besides, the alive plots are analyzed in order to find correlations between the convergence rate and the alive rate. As C-PSO calculates a correction vector which negates the influence of the vector field, the convergence results for C-PSO are the same for each vector field. Thus, they are not analyzed separately in each vector field. In Sphere function, C-PSO reaches the optimum at iteration eight and fast converges (Figure B.11). In Rosenbrock and Ackley function, the swarm reaches similar results. However, the optimum in Ackley function is first found at iteration 15 (Figure B.29).

Vector Field Cross

The first vector field *Cross* (B.1) leads the Explorers to move towards the bottom left-hand corner and top right-hand corner. Therefore, the exploration of the search space in the top left-hand corner and bottom right-hand corner is sparely. However, the information about the top left-hand area is important for the swarms because the optimum is arranged in this sector. CW-PSO is not able to find the optimum in most of the simulations in Sphere function (Figure B.11). The awareness rate rises linearly until the 15th iteration were it drops until the end of the simulation because parallel the particles are dying (Figure B.39). The particles do not have enough information about the vector field to reach the optimum. Consequently, they use a lot of energy because they constantly move in the opposite direction of the vectors of the field. The awareness rates for Rosenbrock and Ackley are slightly better with at maximum 50 percent (Figure B.48 and Figure B.57). But the particles still run out of energy easily. Due to the vector field and the insufficient exploration, CW-PSO is not able to converge in most cases for all functions.

The convergence rate for NW-PSO in Sphere function shows that it is not able to converge in most cases (Figure B.11). Nevertheless, the swarm uses less energy because nearly no particle dies during the simulation as shown in Figure B.39. The awareness rate of NW-PSO rises slower than for CW-PSO and reaches its maximum at iteration 8 with a value of 15 percent. This is due to the fact that most of the particles are fast blown to the borders. Consequently, they can only use the data from the Information Map in the first 20 iterations. However, they are not able to reach the optimum once they are blown to the corners because the vector fields influence is too strong. The convergence rate shows how difficult it is to find the optimum. Only those particles who start in the left-hand corner can even search for the optimum because the other particles do not reach this area due to the vector field. The Rosenbrock and Ackley function show similar results as the swarm cannot even find the optimum in the simplest fitness landscape (Figure B.20 and Figure B.29).

Vector Field Rotation

The second vector field *Rotation* (Figure B.2) causes a rotation around the centre of the search space. In this case, the Explorers are blown in a rotary movement around this centre being slightly drifted to the borders of the space. As a result, the exploration increases linearly. Since the Explorers collect more information at each iteration, the CW-PSO swarm can also trigger more information. Thus, the awareness rate increases, until the 45th iteration where it rapidly drops (Figure B.12). At this point, the CW-PSO particles run out of energy and start dying (Figure B.40). Thereby, the awareness rate starts shrinking because less particles use the data from the Information Map. The convergence rate shows an exponential decrease until the 40th iteration where it reaches the optimum. At this iteration the particles strongly try to converge at the optimum. However, the exploration is insufficient and the particles spend a lot of energy because they move in the opposite direction of the vector field. Consequently, the particles start dying as shown in Figure B.40. The alive rate in Ackley function is identical to the rate in Sphere function but the convergence rate shows that the swarm needs more time to find the optimum (Figure B.30). This is due to the fact that Ackley function provides multiple local minima which are misleading the swarm at first. In Rosenbrock function the swarm also needed more time to converge (Figure B.21). Furthermore, the alive rate illustrates that the particles run out of energy faster than in Sphere function (Figure B.49).

The NW-PSO convergence rate for Sphere function shows a fast decrease at the beginning and stagnates at the optimum at iteration 12 (Figure B.12). This early convergence is due to the fact that the NW-PSO approach is suitable in the vector field because the particles are blown towards the optimum independent of their start position. As shown in Table A.2 a high amount of particles is able to converge at the optimum. This is due to the good exploration. The awareness rises linearly until the 60th iteration and reaches at maximum 94 percent. Afterwards, the rate decrease because the particles start dying (Figure B.40). The particles run out of energy because they have to compensate the vector field in order to stay at the position of the optimum. The results for Rosenbrock (Figure B.21) and Ackley function (Figure B.30) are similar but the swarm needs more time to find the optimum. In Rosenbrock function the particles start dying earlier because they still search for the global optimum when they concentrate at the best found solution (Figure B.49).

Vector Field Sheared

The third vector field *Sheared* (Figure B.3) is a flow from the centre of the search space towards the corners. As a result, the Explorers are quickly drifted towards the borders and corners and stop exploring. This leads to worse awareness rates than in the previous vector field (Figure B.13). Nevertheless, the awareness rate for CW-PSO rises until the 30th iteration where it stabilizes until the 45th iteration. Thenceforward, the rate falls parallel to the alive rate (Figure B.41). At the end, less than 20 percent of the population is still alive. This behaviour is owed to the bad exploration. As a result, the particles cannot reach the optimum. However, they are continuously trying and spent a lot of energy on their movements. The convergence rates for the CW-PSO swarm show a fast drop for all objective functions. However, the average rate stagnates and does not reach the optimum (Figure B.13). Because of the bad exploration, the CW-PSO particles have less occasions to use the data from the Information Map and correct their movements according to the vector field. As a result, they struggle to reach the optimum because their movement is massively distributed by the vector field. The results for Rosenbrock function and Ackley function are even worse as the swarm is not even able to reach the optimum in Sphere function (Figure B.22 and Figure B.31).

Although, the awareness rate reaches less than 10 percent at its maximum, the NW-PSO convergence rate in Sphere function shows a fast stabilization at the optimum at iteration 15 (Figure B.13). Compared to the CW-PSO swarm, the NW-PSO swarm reaches the optimum because it even navigates if there is not enough information about the vector field. The particles which start in the middle or left-hand corner are able to reach the optimum because they are blown towards it. Additionally, they correct their movements in order to not be disturbed by the vector field. NW-PSO is also able to find the optimum in most of the cases in Ackley function. Nevertheless, the average convergence rate shows that the particles do not reach the optimum which illustrates the varying search results depending on the particles start position (Figure B.31). Moreover, NW-PSO is in most of the simulations in Rosenbrock function unsuccessful (Figure B.22). This is due to the fact, only a small number of particles converges at the optimum as shown in Table A.2. This complicates the search process especially for Rosenbrock function.

Vector Field Wave

The forth vector field *Wave* (Figure B.4) lets the Explorer particles slowly fly to the bottom of the search space. As a result, there is a lot of exploration because the particles which start at the top are blown through almost the whole search space vertically. The alive rates in the forth vector field display that in each swarm and each function no particle runs out of energy and dies (Figure B.42). In Sphere function, CW-PSO reaches the optimal solution as fast as C-PSO (Figure B.14). The awareness rate rises at the beginning until the 20th iteration where it stabilizes at 40 percent. The influence of this vector field is not as strong as other vector fields which is shown by the alive rate. Although, only 40 percent of the swarm use data from the Information Map it is able to reach the optimal solution as fast as C-PSO. Nevertheless, the exploration is not high enough for the swarm to concentrate at the optimum and stop moving. Instead it is constantly moving in the area of the optimum. In Ackley function, the swarm also finds the optimum in most of the simulations (Figure B.32). The particles are not mislead by local optima and do not search for better solutions because they are not able to concentrate at a specific position and therefore continue investigating the fitness landscape. The convergence rate for Rosenbrock function increases similar to Sphere function and the optimum is found in most of the simulations (Figure B.23).

The convergence rate for NW-PSO in Sphere function drops slower than the rate for CW-PSO and C-PSO. The optimum is found in the 35th iteration (Figure B.14). This is due to the flow of the vector field. Some particles move horizontally in the same direction as the flow but slowly. Therefore, they need more time to reach the area of the optimum and to improve the search results. As the flow of the vector field is going from the top towards the bottom, nearly no particle is blown towards the optimal area. Only those who start in the top area of the vector field are able to reach the optimum. The others are blown to the bottom or the borders. The awareness rate is increasing fast at the beginning because the swarm tries to follow the flow of the vector field. Therefore, they are driven to the same places as the Explorer. The awareness rate rises until 45 percent of the swarm uses the data from the Information Map but decreases at iteration 50. This is due to the fact, that a lot of particles now reached the borders and quit the search process. Given, that the number of particles starting in the upper area is low, the search for the optimal solution is difficult especially for Rosenbrock and Ackley function. As a consequence, the average convergence rate shows that the swarm is in most cases not able to find the optimum in those functions (Figure B.23 and Figure B.32).

Vector Field Tornado

The fifth vector field *Tornado* (Figure B.5) causes a strong rotational flow from the corners and borders towards the centre of the search space. Thus, the Explorers are blown proportionally fast towards the middle where they start concentrating after some iterations. As the vector field is intense, the exploration is difficult because the particles are blown towards the center of the space. Nevertheless, the awareness rate for CW-PSO in Sphere function rises until the 20th iteration to 65 percent and falls at the 50th iteration parallel to the alive rate (Figure B.43). The awareness rate drops because the particles are dying. At the end, almost no particle of the start population is still alive. The convergence rate for CW-PSO shows a fast decline in the beginning and reaches the optimum at the 30th iteration (Figure B.15). Nevertheless, the particle start dying because in most simulations the exploration is insufficient and the swarm is not able to concentrate at the optimum. Consequently, it is continuously moving and spends more energy and dies earlier than C-PSO. In Rosenbrock function, the swarm did not find the optimum in most of the simulations (Figure B.24). A reason for that is that the particles ran out of energy even faster than in Sphere function (Figure B.52). Only 11 percent of the population is left in the end. However, it is difficult to find the optimum with a small number of particles. Additionally, the particles have a higher chance of success in Rosenbrock function if the particles converge to one point. The results in Ackley function are similar to those in Sphere function (Figure B.61). However, the average convergence rate shows that the swarm did not find the optimum (Figure B.33). This happens for the same reason as in Rosenbrock function. The convergence rate for NW-PSO in Sphere function is similar to the convergence rate of C-PSO (Figure B.15). The optimum is found in the 10th iteration. Additionally, only 21 percent of the particles die in this vector field as shown in Figure B.43. The awareness rate is linearly increasing until the 40th iteration where it stabilizes at 80 percent. At this iteration it slightly falls because the particles are dying. The rate is high because a lot of particles are blown towards the center like the Explorer particle. As a consequence, the center is explored and the particles use the collected data from the Information Map. The convergence rate in Rosenbrock function shows that NW-PSO is not able to find the optimum in most of the cases (Figure B.24). As already mentioned, only a small number of particles converges at the best found solution and the others are blown towards the center. As a result, it is difficult to find better solutions if the number of searching particles is small. Furthermore, in Ackley function, the average convergence rate also shows that the optimum is not found (Figure B.33).

Vector Field Bi-Directional

The sixth vector field *Bi-Directional* (Figure B.6) is a bi-directional flow. In the upper half, the particles are blown towards the left border and in the lower half towards the right border. The Explorer particles move in a parallel line to the horizon towards one of the borders depending on their start position. Thus, only the line segments of the particles start position towards the border are explored and stored in the Information Map. In contrast to C-PSO, the alive rate for CW-PSO in Sphere function begins declining at the 60th iteration and reaches 85 percent at the end of the simulations (Figure B.44). The convergence rate is similar to C-PSO as shown in Figure B.16. The awareness rate is linearly increasing until the 30th iteration where it stagnates until the 60th iteration at which the particles start dying. The rate rests at 60 percent at the end. The CW-PSO particles are able to converge at one position because of the high awareness rate. As a consequence, the search results are better than in the previous vector field. The alive rate in Rosenbrock function falls until 75 percent of the population remains which are 10 percent more than in Sphere function (Figure B.53 and Figure B.44). As already explained, the particles need more energy to converge at the optimum because they longer move and try to find better solutions. The alive rate and awareness rate in Ackley function are the same as in Sphere function (Figure B.62 and Figure B.44). Nevertheless, the swarm is in most cases not able to find the optimum (Figure B.34). This is due to the local minima. If the swarm has not enough information about the vector field it cannot concentrate at the optimums position and slowly improve its search results as it is needed in Ackley function. Therefore, the search results are highly varying and the average value is not at the optimum. The NW-PSO swarm moves energy efficient in Sphere function which causes that none of the particles dies (Figure B.44). The awareness rate rises until the 30th iteration where it reaches its maximum of 23 percent. The convergence rate is identical to CW-PSO (Figure B.16). Approximately, 28 percent of the particles reach the optimum as the other ones are blown towards the borders (Table A.2). Those who are blown to the borders do not use the data from the Information Map because the particles are stuck at the border. NW-PSO can easily converge because it is blown towards the optimal solution by the vector field. Thus, the energy usage is also low as shown in the alive rate. Nevertheless, Rosenbrock function is slightly more challenging as five percent of the population dies and the energy consumption is higher (Figure B.53). Although, the vector field is profitable for the swarm, they did not reach the optimal solution in Rosenbrock function (Figure B.25). A reason for that might be, that all particles are blown from the right border towards the optimum and converge at the first best found solution. Consequently, they are not able to visit more places which are positioned on the left side of the best found solution. Moreover, the small number of particles which converges makes it difficult to search for better solutions. In Ackley function, the results are similar to these in Sphere function (Figure B.34 and Figure B.16). However, the swarm needs more time to find the optimal solution at iteration 20.

Vector Field Random

The seventh vector field *Random* (Figure B.7) consists of randomly positioned vectors with varying magnitudes. Thereby, the Explorers are blown in random directions. Most of them are stuck at their position and cannot further explore the search space. This circumstance effects the awareness rates of the other swarms as less data is collected about the vector field. The alive rate in Sphere function shows that each swarm still has an alive rate of 100 percent at the end of the simulation in all fitness landscapes (Figure B.45). It indicates that the energy usage is relatively low if no particle dies. The convergence rate of CW-PSO in Sphere function is identical to the convergence rate of C-PSO (Figure B.17). Although, the awareness rate is

relatively low, as it reaches its maximum of 28 percent at the 15th iteration, the swarm can easily find the optimum. This is due to the fact, that the vector field provides random vectors. Consequently, there does not arise a continuous flow which leads the particles in a specific direction. As a result, the particles are less disturbed by the vector field even if the awareness is low. This is underlined by the results in Rosenbrock function and Ackley function (Figure B.26 and Figure B.35). Although, the fitness landscape is more challenging to investigate, the swarm finds the optimum as fast as C-PSO.
In contrast to the other swarms, NW-PSO needs a lot more time to find the optimal solution at iteration 10 (Figure B.17). Although, the awareness rate rises directly at the beginning to 50 percent the swarm has more difficulties to find the optimum. A reason is that the swarm always tries to move in the same direction as the vector field in order to reduce the energy consumption. Though, the *Random* vector field does not provide a directed flow. Instead the particles are mislead by the random vectors of the vector field. Consequently, they need more time to concentrate at one position and find the optimum. This provides especially in Rosenbrock function and Ackley function a higher impact. The convergence rate in Rosenbrock function shows that it is difficult for the swarm to converge as it does not find the optimum in most cases (Figure B.26). In Ackley function, the swarm is able to find the optimum in most cases but the convergence rate shows that the search results are highly varying as the average results is not the optimum (Figure B.35).

Vector Field Multi-Rotation

The eighth vector field *Multi-Rotation* (Figure B.8) is a multi-rotational flow, by which the Explorers are drifted towards the borders. In comparison to the other vector fields, this one provides a lot of rotations and a continuous flow which makes it possible that the Explorer visit more positions and collect more information but also get stuck easily. Likewise the previous vector field, none particle of the swarms dies in any fitness landscape because the vector field is not as strong as others. In Sphere function, the awareness rate of CW-PSO is constantly rising at the beginning until the 30th iteration where it stagnates at 60 percent (Figure B.18). Moreover, the swarm finds the optimal solution as fast as C-PSO. This is due to the fact that the vector field does not have an high impact on the particles movement due to the good exploration of the vector field. As there are a lot of local tornado, the Explorer are drifted towards their center and can explore them. Consequently, the CW-PSO particles are not drifted towards these center because they can correct their movement using the data from the Information Map. However, finding the optimum in Rosenbrock function is slightly more challenging because the particles have to slowly converge to better solutions once they have concentrated at one position (Figure B.27). As the awareness rate shows, not all places are investigated by the Explorers. If the vector at the optimums position is not known, the swarm has difficulties to move to this position which worsens the search results. In Ackley function

the swarm has even more difficulties than in Rosenbrock function as it needs more time to find the optimal solution (Figure B.36).
In Sphere function, the awareness rate for NW-PSO rises rapidly at the beginning until the 10th iteration where it stagnates and starts growing at the 70th iteration (Figure B.18). The awareness rate first rises because the particles start at the same positions as the explorers and therefore move across the same points which the Explorer visited and stored information about in the Information Map. Afterwards, the awareness rate first stagnates and subsequently rises again because the particles converge try to reach the optimal solution and therefore use the data from the Information Map. In contrast to CW-PSO, the NW-PSO swarm tries to follow the flow of the vector field like the Explorer. Therefore, more particles move to the same places as the Explorers. As a consequence, a lot of particles are stuck in local tornado because the flow drifted them towards this position. The NW-PSO swarm needs more time to converge compared to the other swarms because the particles need more time to concentrate at one position. As the particles try to reduce the energy, they do not use the shortest path to reach the best found solution but move in the same direction as the flow. In Rosenbrock function, the swarm is not able to find the optimum in most of the cases which is due to the amount of particles at the position of the best found solution (Figure B.27). When all CW-PSO particles converged at the optimum, only 26 percent of the NW-PSO population concentrated at the best found solution (Table A.2). The other particles where blown towards the center of the tornado and are not able to reach the optimum. However, especially for Rosenbrock function the number of searching particles is highly affecting the search results. In Ackley function, the swarm also had more difficulties than in Sphere function which is shown by the average convergence rate (Figure B.36).

Vector Field Vortex

In the ninth vector field *Vortex* (Figure B.9), the Explorer particles are either blown to the borders or stuck in the middle of local tornadoes inside the vector field. As a result, the exploration of the vector field is insufficient. The alive rates in this vector field provide the same results as in the previous vector fields (Figure B.47). The convergence rate shows a drastic decline at the beginning until the optimum is found in the 40th iteration (Figure B.19). Though, the awareness rate only reaches 28 percent at the 15th iteration where it remains until the end of the simulation. Due to the bad exploration in the area of the optimum, the swarm is not able to converge at a specific position. Therefore, it needs more time to find the optimum compared to C-PSO. In Rosenbrock function it cannot find the optimum in most of the simulations due to the distraction by the vector field (Figure B.28). The results in Ackley function are more likewise the results in Sphere function (Figure B.37).
The awareness rate for NW-PSO in Sphere function is rising to 50 percent until the 40th iteration and stabilizes until the end of the simulation (Figure B.19). This is due to the fact,

that most of the particles get stuck in local tornado likewise the Explorers. As a consequence, they use the information collected by the Explorers as their positions are fully explored. The NW-PSO average convergence rate shows for each fitness landscape that the swarm in most simulations is not able to find the optimum. This is reasoned by the number of particles which are even searching for the optimal solution as most of the particles are blown away. These are for example the particles in the left-hand corner and sometime particles of the top right-hand corner. As they are only a few particles which start in this area, the search process becomes complicated. Even in Sphere function there are not enough searching particles to find the optimal solution. Consequently, the results for Rosenbrock function are even worse as shown in Figure B.28. Though, the results for Ackley function also show that the swarm is not able to find the optimum (Figure B.37). The results are highly depending on the start positions of the particles. If the start position of many particles is beneficial, the swarm is able to find the optimal solution.

5.2.2. Energy Analysis

This section is about the energy analysis for the three optimization swarms. Therefore, the energy plots are surveyed. The total swarm energy at each iteration is shown so that a conclusion about the energy consumption can be drawn. In addition, alive plots and convergence plots from the previous section are considered in order to explain the energy development.

Vector Field Cross

In the first vector field *Cross* (Figure B.1), the energy rate for C-PSO in Sphere function shows a fast decrease until the 15th iteration where it continues falling shallowed (Figure B.67). The energy rate reaches the value zero at iteration 80. At the beginning, the energy rate decreases faster because nearly all particles need to move in the contrary direction of the vector field to get to the optimum as it is placed in the top left-hand corner. When the particles reach the optimum, they need less energy because they only have to compensate the vector field and do not move any more. Therefore, the energy falls less at the 15th iteration. At this time all particles reach the optimum. In Rosenbrock function the energy rate presents similar results (Figure B.76). Although, in contrast to Sphere function, the particles do not instantly find the optimum. When they concentrate at the best found solution they still improve the results and use energy to move to a better solution which was found. As a consequence, the energy usage is higher from iteration 15 to iteration 80 than for Sphere function.
In Sphere function, CW-PSO generates a drastically falling energy rate which is zero at the end of the simulation (Figure B.67). As a consequence, all particles are dead. In contrast to C-PSO, CW-PSO has no information about the vector field except the data from the Information Map. Nevertheless, the data is insufficient because the particles are not able to reach the optimum and concentrate at its position. Instead, they are searching for the optimal solution

and continuously move in the contrary direction of the vector of the field. This causes high energy usages as shown in Figure B.67.
The NW-PSO swarm spends essentially less energy in Sphere function (Figure B.67). The rest energy almost stagnates and nearly no energy is used. This is due to the fact, that most of the particles fly in the same direction as the vectors of the vector field. Nevertheless, they cannot find the optimum in most cases because they are driven to the corners. Only those particles which concentrated at the best found solution use energy for compensation of the vector field. Their alive rate underlines the low energy usage because almost no particle dies (Figure B.39). Whereas, the whole population of C-PSO and CW-PSO is dead at the end of the simulation.

Vector Field Rotation

In the second vector field *Rotation* (Figure B.2), the energy rate for C-PSO in Sphere function displays a linear decrease until iteration 40 at which it flattens slightly (Figure B.68). In contrast to the previous vector field, the amount of energy at the beginning is not higher than during the whole simulation. This is caused by the rotational flow of the vector field. Therefore, some particles use less energy to converge at the optimum if their position is beneficial according to the flow. At iteration 40, the particles start dying and therefore the energy rate flattens until all particles are dead at iteration 70. The particles run out of energy because of the compensation of the vector field. The results in Ackley and Rosenbrock function are nearly the same as in Sphere function with the difference that the particles are dying earlier in Rosenbrock function (Figure B.77).
In Sphere function, the energy rate for CW-PSO displays a fall parallel to the C-PSO Swarm (Figure B.68). Whereby, the rate for CW-PSO is slightly better than for C-PSO until iteration 30. Subsequently, the C-PSO swarm offers a better energy rate. Nevertheless, the energy is zero at the end of all iterations because the particles concentrate at the best optimum and investigated energy in order to stay at their position. As a consequence, all particles die (Figure B.40). The CW-PSO swarm first uses less energy because the optimum is not found. When it is found at iteration 25 (Figure B.12), the whole swarm tries to converge at this position and uses a lot of energy. In Rosenbrock function, the CW-PSO swarm again uses less energy but at iteration 25 it uses exactly the same energy as C-PSO (Figure B.77). This is due to the fact, that both swarms converge and afterwards slowly improve their search results by little movements. The results in Ackley function are more alike the results for Sphere function, although the swarm needs more time to find the optimum and therefore shows better results than C-PSO until the 45th iteration (Figure B.86).
In contrast, the NW-PSO swarm uses less energy in Sphere function because the particles move in the same direction as the vector field and therefore need less energy (Figure B.68). Though, 92 percent of the population dies because nearly all particles converge at the optimum and spend energy on staying at this position (Figure B.40). At the beginning, the energy

rate is falling slowly until the 20th iteration. At this time, the optimum is found and particles start compensating the vector field. From iteration 60 until the end of the simulation, the energy rate flattens due to the dying of the particles. But in the end, there is still energy left compared to the other swarms. The results in Ackley function are the same as in Sphere function (Figure B.86). The development of the energy rate in Rosenbrock function relates to this energy rate (Figure B.77). The energy rate likewise declines faster when the particles converge but less particles converge and therefore less particles need to compensate the vector field and increase their energy usage.

Vector Field Sheared

In the third vector field Sheared (Figure B.3), the energy rate for C-PSO in Sphere function shows a drastic linear decrease until the end of the simulation (Figure B.69). At the beginning, the energy effort is minimal higher than in the rest of the simulation due to the intense vector field which causes a flow to the borders. The particles are concentrating at one position and therefore also move in the contrary direction of the vector field and need more energy. When the first particle reaches the optimum at iteration seven (Figure B.13), the energy is dropping less because the particles only spend energy for compensation of the vector field and do less movements. Admittedly, the swarm reaches worse results than CW-PSO. The results in Rosenbrock and Ackley function are almost the same (Figure B.78).

In Sphere function, the energy rate for CW-PSO shows a fall which ends at 20 percent remaining energy. The particles are fast blown to the borders due to the flow of the vector field. The exploration of the vector field is bad as shown in the awareness rate (Figure B.41). As a consequence, the swarm struggles to reach the optimum because it cannot make correction in order to compensate the vector field. The particles are consistently blown back to the borders if they move towards the best found solution. This means they use a lot of energy because they move in the contrary direction of the vectors of the field. The energy rate flattens at iteration 40 because the particles start dying. The CW-PSO reaches the same results in Rosenbrock and Ackley function (Figure B.78).

The NW-PSO energy rate in Sphere function is a lot better than the rates of the other approaches (Figure B.69). It shows a parallel development as the energy rate of C-PSO. However, the fall is smaller because the amount of particles which converge at the optimum is smaller. Consequently, the number of particles which need to investigate energy to stay at their position is also smaller. Less particles converge because most of the particles follow the flow of the vector field. As a consequence, only a few particles are even in the area of the optimum to reach it. At the end, 75 percent of the start energy is left. This also applies to Ackley function (Figure B.87). In Rosenbrock function, NW-PSO also uses more energy like C-PSO (Figure B.78). The reason is that the swarm still moves if the particles concentrate at the best found solution because it finds better solutions. However, this shows a high energy usage because the particles need to compensate the vector field in their movements.

Vector Field Wave

In the fourth vector field *Wave* (Figure B.4), the energy rate for C-PSO is decreasing rapidly at the beginning and flattens at the 10th iteration (Figure B.70). Due to the fast convergence, the particles early stop moving and only invest energy on compensation. Nevertheless, at the end 85 percent of the start energy is left. In relation to the other vector field, this vector field is less strong which causes less energy usage as shown in Figure B.70. In Rosenbrock and Ackley function, the energy rate decreases as in Sphere function (Figure B.79).
CW-PSO also early reaches the optimum in Sphere function (Figure B.14). Consequently, it starts compensating at the 10th iteration. As shown in Figure B.70, the energy rate drops fast until the 10th iteration. Afterwards, it flattens slightly and continues declining. Nevertheless, CW-PSO needs less energy than C-PSO because the particles start compensating earlier. The energy rate for Ackley function is the same as in Sphere function (Figure B.88). Thus, in Rosenbrock function, the swarm needs more energy towards the end of the simulation (Figure B.79). Therefore, it reaches the same energy value as C-PSO.
The NW-PSO swarm has a better energy rate in Sphere function as it stays at 97 percent left energy at the end (Figure B.70). The reason is that most particles are blown towards the borders and only a little number reaches the optimum (Table A.2). Thus, only those who concentrate at the optimum need to use energy in order to stay at this position and compensate the vector field. At the beginning, the energy consumption is a little higher because the particles first have to converge. Subsequently, they do not move any more. In Rosenbrock function, the energy rate is slightly worse which is caused by the improvement of the search result when the particles concentrate at the best found solution (Figure B.79).

Vector Field Tornado

In the fifth vector field *Tornado* (Figure B.5) the energy rate for the C-PSO swarm presents a faster decrease until the 10th iteration where it flattens (Figure B.71). When the swarm finds the optimum in the 10th iteration, all particles try to converge and therefore used a lot of energy. Afterwards, the energy usage is reduced because no more movement is needed. The only energy is spend on the compensation of the vector field. At the end, no energy is left. Whereas, the energy rate in Ackley function is the same as in Sphere function (Figure B.89), the energy rate is faster declining in Rosenbrock function (Figure B.80). The reason is already mentioned in the previous vector fields. As the particles improve their search result by the time and spend energy on little movements. As a result, all particles are already dead at iteration 87 (Figure B.52).
The energy rate for CW-PSO in Sphere function shows a linear decrease until the 60th iteration (Figure B.71). Subsequently, the rate continues falling slower because the particles start

dying (Figure B.43). There energy usage remains almost the same during the simulation because the particles are constantly moving. Due to the bad exploration, they are not able to converge at the optimum. As a consequence, the energy usage is higher than for the C-PSO swarm which is able to converge. The energy rate in Rosenbrock function falls further than in Sphere function (Figure B.80). Ackley function offers the same results as Sphere function (Figure B.89).

The energy rate for NW-PSO slightly falls at the beginning and than flattens like C-PSO (Figure B.71). Only a few particles are even able to reach the optimum because of the location of their start position. As most of the particles are blown towards the center, only a small number needs to investigate energy to converge at the optimum. The energy rate in Ackley function is slightly worse (Figure B.89). This is due to the fact that the particles find better solutions when they converge and slowly improve their search result. Therefore, the energy usage rises because often the movement is in the contrary direction of the vectors of the field. In Rosenbrock function, the swarm uses the least energy (Figure B.80).

Vector Field Bi-Directional

In the sixth vector field *Bi-Directional* (Figure B.6), the energy rate for C-PSO in Sphere functions presents a linear drop until the end of the simulation. As well as in the previous vector fields, the swarm finds the optimum after the first ten iteration. Thus, all particles try to converge and slightly increase the energy usage by their movement. When the particles reach the optimum, they do not need any more movement and the energy usage is reduced. Whereas, C-PSO saves more energy than CW-PSO in Sphere function, the swarm uses more energy than CW-PSO in the other fitness landscapes. The results in Rosenbrock function show a higher decrease because the particles need slightly more time to find the optimal solution and to converge (Figure B.81). The swarm needs even more time to find the optimum in Ackley function as shown in Figure B.34. Thus, the particle later stop moving and the energy consumption is higher (Figure B.90).

The energy rate for CW-PSO falls parallel to the energy rate of C-PSO as both swarms converge (Figure B.72). This vector fields provides a bi-directional flow by what the Explorers are drifted straight to one side of the vector field. In most cases, the Explorer visit the position of the global optimum. Therefore, the CW-PSO swarm is able to converge at its position. CW-PSO provides almost the same results for each fitness landscape (Figure B.81).

Most of the NW-PSO particles are also drifted to the borders where they are not able to reach the optimum anymore. But those who reach it use less energy than the other swarms as shown in Figure B.72. This is due to the fact that the particle movement is determined by the flow of the vector field. Only those who have a beneficial start position are blown towards the optimum and therefore are able to converge. The other particles are blown to the borders. Consequently, a small number of particles continues spending energy. The energy usage in

Rosenbrock function is smaller than in Sphere function (Figure B.81) because the particles stop moving earlier. In contrast to Sphere function, it is more challenging to find the optimum in Rosenbrock function and the particles are easily mislead by local minima. In Ackley function, the energy rate is higher than in Sphere function because the particles slowly improve their search results compared to Rosenbrock function and therefore spend more energy (Figure B.90).

Vector Field Random

In the seventh vector field *Random* (Figure B.7), the energy rate for C-PSO shows a rapid decrease at the beginning until the 10th iteration (Figure B.73). Henceforward, the rate continues falling more slowly. At the beginning, the swarm uses a lot of energy to converge at the best found solution and easily finds the optimum at the seventh iteration (Figure B.17). Afterwards, the energy consumption is reduced to the compensation of the vector field. At the end of the simulation, the remaining energy is 86 percent of the start energy. In Rosenbrock function, the swarm needs more time to converge and find the optimal solution. As a result, the energy usage at the beginning is slightly higher (Figure B.82). However, in Ackley function, the swarm finds the optimal solution faster and uses a little less energy (Figure B.91). The CW-PSO swarm is able to find the optimum even with a low awareness rate because the vector field does not provide a directed flow (Figure B.17). Thus, the CW-PSO particles are less distributed and can early reach the optimum. As a consequence, the energy usage at the beginning is higher and decreases at iteration 10 (Figure B.73). Due to the bad exploration, the swarm is not able to converge to a specific point as the particles are shallowly distributed by the vector field. Nevertheless, the energy usage is not significantly higher than for C-PSO because some particles move in the same direction as the vectors of the vector field and therefore reduce their energy consume. Additionally, the C-PSO particles all need to compensate the vector field. However, in Rosenbrock function the difference between C-PSO and CW-PSO becomes more apparent (Figure B.82). CW-PSO still has 87 percent of the energy left at the end of the simulation which is three percent more than C-PSO. In Ackley function, the swarm reaches the same results as in Sphere function (Figure B.91).

NW-PSO is able to drastically reduce the energy consumption in all fitness landscapes. In Sphere function, the energy rate is linearly falling but in contrast to the other swarms, NW-PSO does not use more energy at the beginning (Figure B.82). Even when the particles have found the optimal solution, the energy usage remains the same. Most of the particles follow the random flow of the vector field. As a result, the energy usage is relatively low. Only those particles which are in the area of the optimum correct their movements according to the vector field and converge. The results for all functions are almost the same (Figure B.82).

Vector Field Multi-Rotation

In the eight vector field *Multi-Rotation* (Figure B.8), the results for the energy rates of C-PSO in Sphere function are related to the rates in vector field *Tornado*. Though, the rates show a high energy consumption at the beginning until the 10th iteration due to the convergence at the best found solution (Figure B.74). As the swarm converges fast in each fitness landscape, the energy rates are the same for all (Figure B.83).

As well as in vector field *Tornado*, the CW-PSO swarm first offers a better energy rate than C-PSO (Figure B.74). However, down iteration 55 the energy rate becomes marginal lower than the C-PSO rate. The exploration of the vector field is as difficult as in *Tornado*. Thus, the swarm is often not able to converge which cause a higher energy usage than for C-PSO. The results are the same in Rosenbrock and Ackley function because the swarm still is not able to converge (Figure B.83 and Figure B.92).

Similar to the vector field *Tornado*, some particles of the NW-PSO swarm are not able to reach the global optimum. As the vector field provides multiple local tornado, a lot of particles are drifted towards them and get stuck. Only a few particles are able to converge at the best found solution. As most of the particles do follow the flow, the energy rate is significantly high and the swarm remains 98 percent of the start energy in Sphere function (Figure B.74). The results for Rosenbrock and Ackley function are the same because the number of particles which reach the optimum is also the same (Figure B.83).

Vector Field Vortex

In the ninth vector field *Vortex* (Figure B.9), the energy rate for C-PSO in Sphere function also falls drastically at the beginning until the optimum is found at iteration 10 (Figure B.75). Henceforward, the swarm uses energy for staying at its position and compensating the vector field. The energy rates in Rosenbrock and Ackley function are the same (Figure B.84). Though, the optimum in Ackley function is first found at iteration 15 (Figure B.37). The results for C-PSO are similar but the swarm uses more energy during the simulation (Figure B.75).

Likewise the previous vector fields, the exploration is insufficient because the Explorers easily get stuck. As a consequence, the CW-PSO particles have difficulties to converge at the optimum. Instead, they are constantly trying to move to its position but are disturbed by the vector field. As the movement costs more energy than staying at the best solution, the energy rate drops further than the one of C-PSO. But at the end, still 83 percent of the energy remains in Sphere function. The energy rate in Ackley function is identical to the one in Sphere function (Figure B.93). However, in Rosenbrock function the energy rate decreases slightly more (Figure B.84).

The energy rate for NW-PSO shows a linear decrease in all fitness landscapes (Figure B.75). At the end, more than 98 percent of the energy remains because most of the particles are blown to the border or got stuck in local tornado like the Explorers. Only a small number of particles reach the optimum and spend energy for compensation.

5.2.3. Success Rate Analysis

After the convergence and energy plots have been analyzed before, the following section surveys the success rate for each swarm. Therefore, a plot is analyzed for each objective function including the different success rates for each swarm. The plot illustrates the percentage of successful simulations. A simulation is appointed to be successful, if the swarm was able to find the optimum. The equation is given below:

$$s(x) = \begin{cases} 1 & , \min \vec{x}_g(t) \leqslant p \quad for\ t = 1, ..., I \\ 0 & , else \end{cases} \tag{5.4}$$

$$f(x) = \frac{1}{M} \cdot \sum_{i=0}^{M} s(x) \tag{5.5}$$

Where $s(x)$ indicates whether a simulation was successful or not. Therefore the global best solution $\vec{x}_g(t)$ at iteration t is compared to the optimum p. Whereby the optimum includes a threshold as defined in Table 5.1. Subsequently, $f(x)$ in Equation 5.5 determines the percentage of successful simulations where M denotes the number of simulations. First, the Sphere function will be analyzed, following the Rosenbrock function and in the end the Ackley function.

Sphere Function

Sphere function provides the best success rates in comparison to the other objective functions as shown in Figure 5.1. This is due to the fact that the function does not supply multiple minima and lightens the swarms search process. The C-PSO swarm gains 99 percent of successful simulations in each vector field. The reason is that the swarm does not allow any disturbance by the vector field. As a consequence, the results are as good as for search spaces without vector fields. Nevertheless, this negligence shows a significant drawback for the energy rates as explained in Section 5.2.2.

The results for CW-PSO are the worst in vector field *Cross*. The swarm is only successful in 40 percent of the simulations because the swarm is depending on the information collected by the Explorers. However, this vector field is due to the flow very difficult to discover. The Explorer particles are quickly blown to the borders. Consequently, the CW-PSO particles have difficulties to reach the optimum. The success is highly addicted to the Explorers start positions. NW-PSO offers a slightly better success rate in *Cross* than C-PSO because the particles can navigate even if the exploration is insufficient. However, the swarm also struggles to reach the optimum because only six percent of the particles reach the best found solution due to the flow of the vector field (Tab A.2). This means that only one or two particles of the population search for the optimum. Those particles start in the left-hand top corner and are able to reach the optimum by following the flow of the vector field.

In the second vector field *Rotation*, the success rate is maximal for both swarms. The CW-PSO swarm is able to reach the optimum because the flow of the vector field is suitable for the Explorers. These are blown to almost each place in the vector field. As a result, the exploration is high and the CW-PSO particles can use the collected information in order to get to the optimum. The NW-PSO swarm performs also good in this vector field because the particles are just like the Explorers blown to nearly every position of the search space. As a consequence, they are also blown to the optimum. Table A.2 shows, that nearly all particles converge at the optimum because each particle independent of its start position is driven towards the optimum by the flow of the vector field.

In vector field *Sheared* (Figure B.3) the success rate for CW-PSO is low. This is due to the difficult exploration. The flow of the vector field blows the Explorers particles quickly to the borders of the search space. As a consequence, the CW-PSO particles have unsatisfactory information about the vector field and are not able to correct their movement to get towards the optimum. The swarm is only successful, if the Explorers randomly reach the place of the optimum when they are blown to the borders. In contrast, the NW-PSO swarm gains a success rate of 100 percent. The particles are also blown to the borders but they are able to do corrections in their movement. As a result, they are able to navigate towards the optimum when they drift to the borders.

The results in the fourth vector field *Wave* (Figure B.4) provide the maximum success for CW-PSO. As this vector field causes a top to the bottom flow of the Explorers, nearly the whole search space is discovered. Thus, the CW-PSO particles can take advantage of the Information Map and correct their movement to get towards the optimum. Additionally, the vector field is less strong as the previous vector fields and therefore the movements of the particle are less disturbed. The results for NW-PSO are equally high. The reason is that those particles which start in the top area of the fitness landscape are able to reach the optimum.

In the fifth vector field *Tornado* (Figure B.5) CW-PSO gains a success rate of 86 percent. The swarm is successful because the Explorers are blown from the corners to the centre. In this process they collect information about the vector field which can help the CW-PSO swarm to reach the optimum. Even if the vector at the specific position of the global optimum is not known, the CW-PSO swarm is able to move around the area of the optimum as the swarm cannot converge. The results for NW-PSO are even better than for CW-PSO because the swarm is able to navigate even if the Explorers did not came across the optimum. The particles are likewise blown to the center but within this movement they navigate and are able to reach the optimum.

The results in the sixth vector field *Bi-Directional* (Figure B.6) are similar to the previous vector field. However, the results for CW-PSO are slightly better. This is due to the fact that the Explorers more often come across the global optimum and therefore store necessary information about the vector field. Consequently, the CW-PSO swarm can make corrections and is more often able to concentrate at the position of global optimum. The NW-PSO swarm gains a high success rate again. The approach is suitable for this vector fields as the particles are directly blown towards the optimum by following the flow.

The plot illustrates maximal percentages in the seventh vector field *Random* (Figure B.7) for all CW-PSO. The vector field does not disturb the particles movement of the CW-PSO swarm because there is no continuous flow in a specific direction. As a result, the particles randomly move in the same direction as the vector field and are not mislead. The NW-PSO swarm also gains little worse results than in vector field *Tornado* because the particles try to follow the flow in the vector field. However, this vector field does not provide such a directed flow. Consequently, the particles move more or less randomly. Only some particles which by accident get nearer to the best found solution are able to converge.

The results in the eighth vector field *Multi-Rotation* (Figure B.8) are analogical success rates to the previous vector field. The CW-PSO swarm is always able to reach the optimum because its movement towards the optimum is not highly disturbed. The multi-rotational vector field only provides local flows which affects the CW-PSO swarm less and the vector field is less strong than the others. However, the swarm has difficulties to land at the optimums position because of the bad exploration. As a consequence, the particles can only move around the optimum because they cannot calculate a concrete correction vector to land at the optimum. The NW-PSO swarm again uses its navigation ability to be blown by the local flows towards the optimum.

The results in the ninth vector field *Vortex* (Figure B.9) are likewise the previous ones for CW-PSO. This is due to the fact, that the influence of the vector field is not strong enough to highly disturb the particles movement. Consequently, the swarm is able to find the optimum although it cannot land at it due to the insufficient exploration. The success rate for NW-PSO is nearly 30 percent because, as well as in *Cross*, only six percent of the population reaches the best found solution and improves the search results.

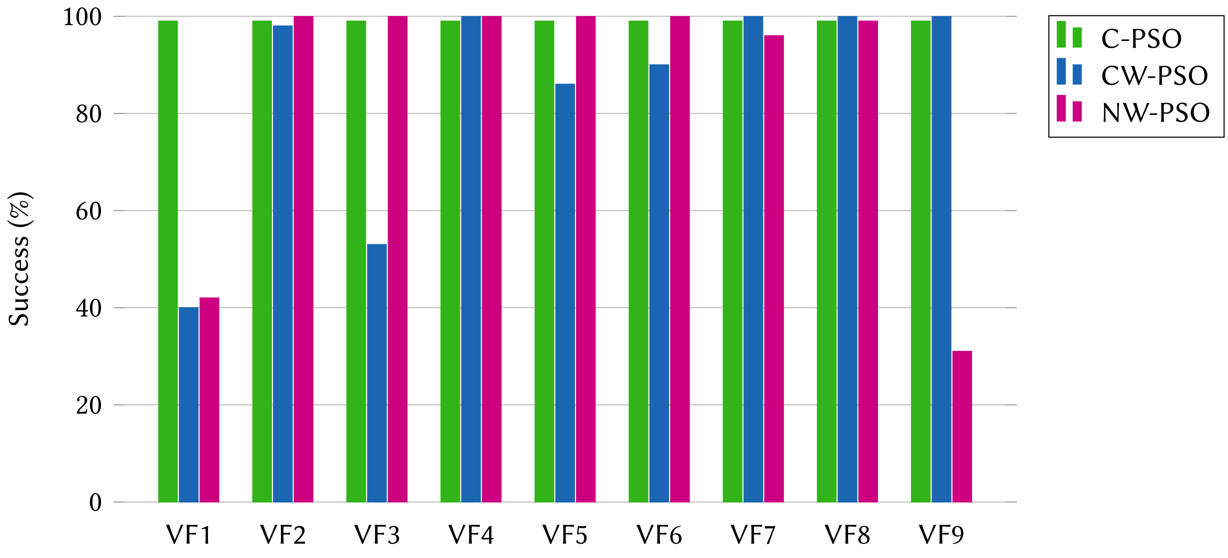


Figure 5.1.: Success Rate Sphere Function

Rosenbrock Function

The success rates for Rosenbrock function are illustrated in Figure 5.2. It offers worse results than Sphere function due to the deeply curved valley in which the global optimum is positioned. This complicates the search process. Although, the success rates for C-PSO are still high, the results in some vector fields are worse than in Sphere function. Thus, the results for C-PSO underline the difficulties for the swarms to find the optimum as C-PSO not takes into account the vector field.

The results in the first vector field *Cross* (Figure B.1) are bad for CW-PSO and NW-PSO. Both success rates do not rise above 10 percent. A reason is the complicated search. The Explorers are quickly drifted away and the discovery is inadequate. As a consequence, there is a huge lack in the Information Map. Especially for objective functions with a plateau of optimal solutions, the exploration needs to be satisfying. Otherwise, the CW-PSO is not able to converge because it is disturbed by the vector field. As a consequence, it cannot move to its pursued position and is easily moved to different places. The results for NW-PSO are even worse than for CW-PSO because in this case only a small number of particles is even able to reach the optimum. These particles are the ones starting in the upper left-hand corner. As important as the Information Map for CW-PSO is the number of particles starting at the appropriate position. Particularly for functions with multiple minima, there need to be as many search particles as possible. The more particles the higher is the chance that they not get stuck at a good solution and are able to find the global optimum. In the case of NW-PSO there are not enough particles which would be able to reach the optimum. Those particles which start at a good position, quit their search too early because they do not find a better solution. As a result, the success rate is very low.

The results in the second vector field *Rotation* (Figure B.2) are better. CW-PSO reaches a suc-

cess rate of 76 percent. As already mentioned, this vector field provides a good discovery by the Explorers. Consequently, the CW-PSO particles are able to use the information and converge at the optimum. The NW-PSO swarm is more successful with 81 percent because they can navigate towards the optimum even if there is not enough information collected by the Explorer. Compared to Sphere function, finding the optimum is more complicated because of the curved valley in which the global optimum is placed. In some simulations, the particles get stuck at the first best found solution and do not find the global optimum.
In the third vector field *Sheared* (Figure B.3) both swarms gain worse results. Whereby NW-PSO gains slightly better results than CW-PSO. As already explained for Sphere function, the CW-PSO swarm has difficulties to reach the optimum due to a bad exploration by the Explorers. The results for Rosenbrock function are even worse than for Sphere function because the particles are not able to reach better solutions inside the plateau. The NW-PSO swarm as well has a low chance of success. Due to the vector field, the particles are easily blown to the borders. Consequently, still a small number of particles reaches the best found solution. Therefore, the search process is hindered.
In the fourth vector field *Wave* (Figure B.4) CW-PSO presents more satisfying results. The success rate scores 64 percent. As already explained for Sphere function, the particles have a high chance to reach the optimum because of the good exploration and the low influence of the vector field. Considering that all particles converge at the best found solution, the swarm is able to find the global optimum in most of the simulations. In contrast, the NW-PSO swarm gains a success rate of 34 percent. The particles get stuck easily at a good solution but not at the global optimum and finish their search process too early. This is caused by the smaller number of particles which concentrate at the best found solution compared to CW-PSO.
The results in the fifth vector field *Tornado* (Figure B.5) are almost the same results for both approaches (Figure 5.2). However, the NW-PSO swarm obtains mildly better results. Nevertheless, both swarms do not find the optimal solution in most of the simulations. The finding of the minimum depends on the exploration of the search space. If the Explorer particles start at benefiting positions, the CW-PSO particles have a higher chance to converge and slowly improve the search results with good correction vectors. The NW-PSO swarm can also navigate without information of the Explorers but again only some particles converge at one point. Most of the particles are driven to the center of the tornado like the Explorers. As a result, the chance of success diminishes because less particles have a lower chance to slowly improve the search results.
In the sixth vector field *Bi-Directional* (Figure B.6), CW-PSO reaches a better success rate with a value of 80 percent. In contrast, NW-PSO only reaches 36 percent successful simulations. The reason is, that a lot particles of the NW-PSO swarm are blown to the borders. As in the previous vector field, this complicates the search and the swarm is less successful. CW-PSO is more often successful because all particles are in the area of the global optimum. Often the exploration is even suitable so that the particles are able to concentrate exactly at the opti-

mums position and improve the search results.

In the seventh vector field *Random* (Figure B.7), the results show the same proportion between CW-PSO and NW-PSO. CW-PSO has a high success rate although the particles are not able to concentrate at the target position due to the bad exploration. However, CW-PSO is suitable for this vector field because when the particles move around their current target there is a high chance to find a better solution inside the plateau of good solutions. The particles for NW-PSO are slightly less successful because of the smaller amount of particles in the area of the optimum. As the particles try to move in the same direction as the vectors of the field, they are often moving in random directions and only those particles which start in the nearer distance to the global optimum are able to search for the best solution.

The results in the eighth vector field *Multi-Rotation* (Figure B.8) are similar results as in the vector field *Random*. It shows a likewise convergence of the particles for CW-PSO. As well as in the vector field *Bi-Directional*, the Explorers are able to store the vector at the position of the global optimum. Consequently, the CW-PSO particles can move to exactly this position and can better improve their search results than without this possibility. Nevertheless, the influence of the vector field is relatively low. As a result, the CW-PSO swarm is less distributed by the vector field and able to get close to the optimum. NW-PSO reaches almost the same success rate as in *Bi-Directional* because not more than 35 percent of the particles converges at the optimum (Table A.2). Most of the particles are stuck at the center of local tornado or at the borders.

In the ninth vector field Vortex (Figure B.9), both approaches offer bad results as both of them fail in most of the cases. However, the results for NW-PSO are significantly bad. Already in Sphere function the swarm had difficulties to find the optimum. Due to the flow of the vector field, only 13 percent of the particles are able to converge at the best found solution (Tab A.2). Although the number of particles is higher than in Sphere function, the swarm still does not find the global optimum because the fitness landscape is misleading the swarm. The CW-PSO swarm reaches a higher success score which is due to the fact that the particles at least concentrate in one area. However, the exploration is in most cases too bad to converge exactly at the optimums position. Nevertheless, the higher amount of particles moving around the area of the optimum is able to find it.

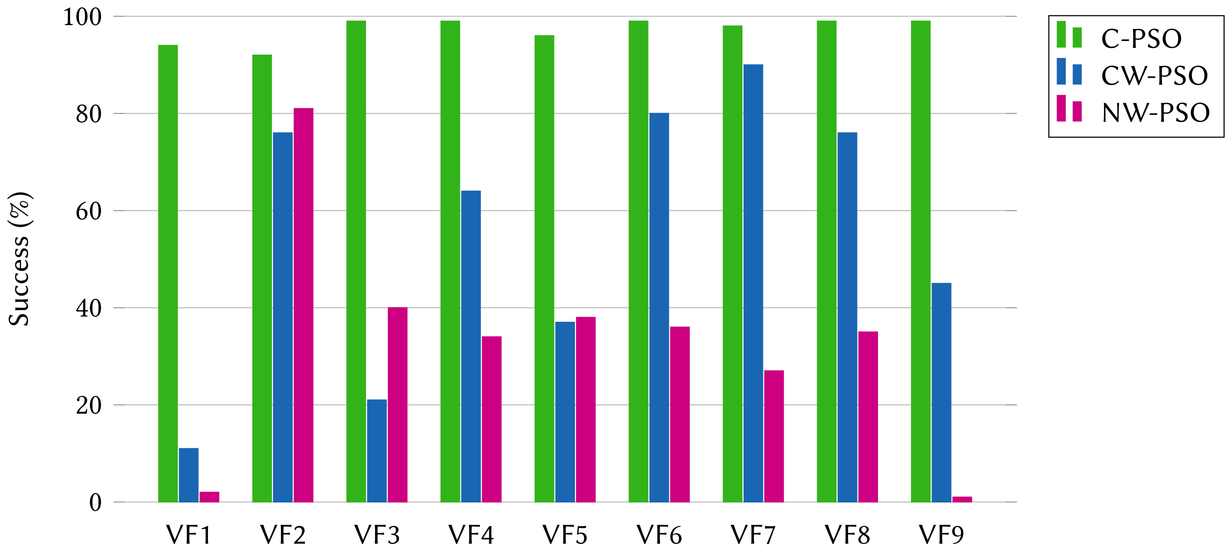


Figure 5.2.: Success Rates Rosenbrock Function

Ackley Function

The third objective function is Ackley function. In contrast to Rosenbrock function it consists of a lot of local minima. The plots are illustrated in Figure 5.3. The results in the first vector field *Cross* (Figure B.1) are similar as for Sphere function although the results are less successful. NW-PSO has a higher chance to reach the optimum because it can navigate in any case and therefore slowly improve its solution at each iteration. The CW-PSO swarm cannot slowly improve its solution if the exploration is insufficient like it is the case in this vector field. As a consequence, CW-PSO gains worse results than NW-PSO.

In the second vector field Rotation (Figure B.2), both approaches present high success rates. The good exploration by the Explorer affords a good search for CW-PSO and NW-PSO. However, NW-PSO reaches a higher success rate because the CW-PSO start dying and therefore cannot find better solutions.

The results in vector field *Sheared* (Figure B.3) are high again for NW-PSO but drastically dropped for CW-PSO. As already explained before, CW-PSO has nearly no chance to reach the optimum because the data of the Information Map is unsatisfactory especially for objective functions with multiple optima. However, NW-PSO gains a high success rate because the particles are able to navigate and can converge when they are blown in the direction of the borders as they come across the global optimum.

In vector field *Wave* (Figure B.4), CW-PSO offers worse results than in Rosenbrock function. The reason is the sometimes bad exploration because the particles are not able to converge at one single position. Consequently, it is difficult to find the optimum in Ackley function. NW-PSO gains a lot better results because it is easier to converge for Ackley function and therefore gets better results than for Rosenbrock function. Although the number of particles which converge at the optimum is the same, the particles can more easily improve the search results for Ackley function.

In the vector field Tornado (Figure B.5), NW-PSO has a high chance of success whereas CW-PSO is unsuccessful in most simulations. In contrast to the Rosenbrock function, this function offers better possibilities for NW-PSO because the particles can slowly converge if they have found a good solution. CW-PSO has more difficulties in this objective function because the particles can hardly concentrate at one position. Nevertheless, Ackley function requires the particles convergence at one point and a slowly improvement of the results by little movements.

The results in the sixth vector field *Bi-Directional* (Figure B.6) are satisfying for both approaches. However, the results for NW-PSO are a lot better than for Rosenbrock function. This is due to the fact that the particles are directly blown towards the optimum and are able to converge at the global optimum by little movements. In contrast, in Rosenbrock function the particles got more easily stuck at a good solution which was not the global optimum. The results for CW-PSO are slightly worse in Ackley function because sometimes the exploration is insufficient. As a consequence, the particles are not able to converge.

The results in the seventh vector field *Random* (Figure B.7) are good for NW-PSO but CW-PSO only reaches a success rate of 31 percent. By comparison, Rosenbrock function was more benefiting because of the layout of the fitness landscape. The particles are not able to concentrate at one position and therefore cannot do little movements. However, these are necessary for a good convergence for Ackley function. The NW-PSO is able to do so and therefore reaches good results.

In the eighth vector field *Multi-Rotation* (Figure B.8), NW-PSO gains similar results as in vector field Random due to the same reason as already mentioned. Nevertheless, the results for CW-PSO are better than before. This is caused by the Explorers as they can offer information so that the CW-PSO particles can converge at one position. In vector field Random the explorers are moved randomly but in vector field Multi-Rotation there is a general flow. As a consequence, the chance that the Explorers cross the optimal position increases. Consequently, the search results for CW-PSO improve as well.

In the ninth vector field *Vortex* (Figure B.9), the success rate for both swarms is bad. As already explained in Sphere function, only a small number of the NW-PSO population searches for the optimal solution. However, the search in this fitness landscape is even more challenging than in Sphere function. Consequently, the results are also worse. Nevertheless, CW-PSO gains minimal better results which is due the fact that the whole swarm moves around the optimum because the vector field is not strong.

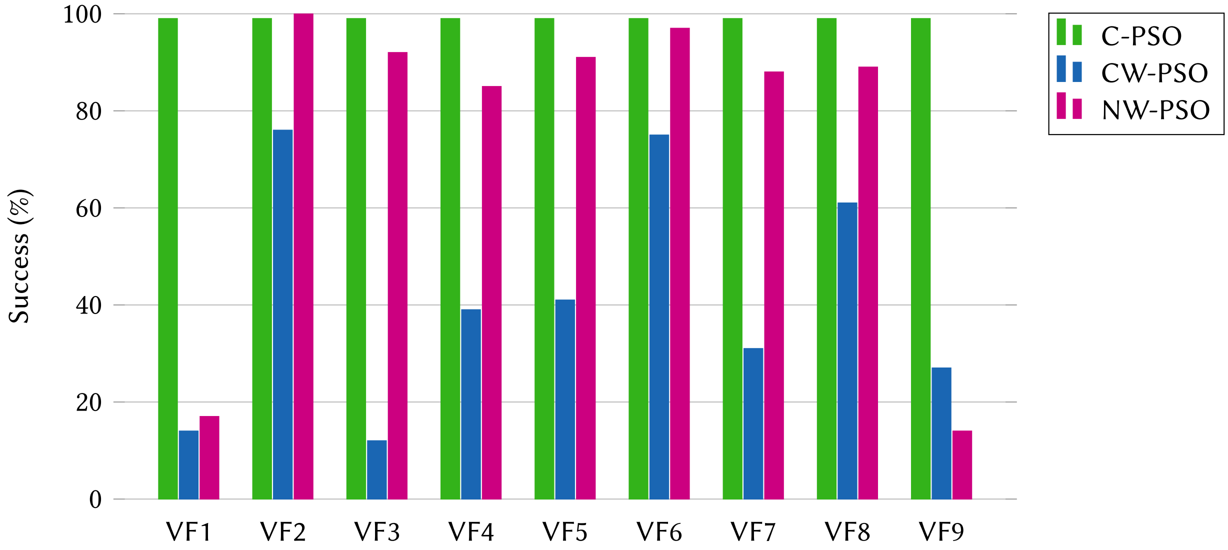


Figure 5.3.: Success Rates Ackley Function

5.3. Summary

The following section summarizes the results of the analysis. First, the new NW-PSO approach was applied on Sphere function. Because of the functions simplicity, the correctness of the proposed model in sense of optimisation could be approved. Nevertheless, NW-PSO was developed for the solving of complex search problems. Thus, it was tested on Rosenbrock function and Ackley function which provided different challenging problems. The challenge for the NW-PSO swarm was, that it is not mislead by minima in order to find the global optimum. In regard to the convergence analysis, the swarm was able to converge in most of the vector fields for Sphere function. However, it did not converge in most of the simulations in the vector fields *Cross* and *Vortex* due to the flow of each vector field. Both vector fields have in common, that the results for NW-PSO are highly depending on the particles start positions. A well-directed choice of the start area would probably improve the convergence results. In contrast, the convergence rate in the vector fields *Rotation*, *Sheared* and *Tornado* stagnated even faster at the optimum than the rate of CW-PSO. The search in the Rosenbrock function and Ackley function is more challenging for NW-PSO which affects the results for the convergence rates. Nevertheless, the rates reach the optimum in the vector fields *Rotation* and *Bi-Directional*. The vector fields have in common that NW-PSO is suitable for them because the environment emphasizes the search process. The particles are able to reach the optimum by moving in the same direction as the flow of the vector fields. Although the exploration of the vector field *Random* is sparely, NW-PSO gains good results because the particles are able to navigate even if there is little information about the environment. Besides, the success rates are of interest. In Sphere function, NW-PSO reaches satisfying success rates which are not falling below 96 percent in all vector fields except [Cross] and *Vortex*. The results in Rosenbrock function show, that NW-PSO is successful in most of the simulations in the vector field

Rotation. In most of the other vector fields, it reaches approximately 35 percent. The difficulty for the swarm is that the particles try to move in the same direction as the flow. This property causes, that they are distributed over the whole fitness landscape. Consequently, only a few particles are able to search in the area of the optimal solution. This hinders the improvement of the search results as particles get stuck at local optima more easily. Especially in Rosenbrock function the number of searching particles has a high impact on the search results. Nevertheless, NW-PSO offers almost as good success rates for Ackley function as for Sphere function. Compared to Sphere function, NW-PSO is still not able to find most of the optima in the vector fields *Cross* and *Vortex*. But in the other vector fields the swarm is able to find the optimum in more than 85 percent of the simulations. The results are always better than for CW-PSO. The new approach NW-PSO is designed to gain good search results in difficult landscapes but also saves energy during the search process. Thus, the energy consumption was also analyzed. The significantly worst energy rate is generated in the vector field *Rotation*. This is due to the reason, that nearly the whole swarm reaches the optimum and needs to compensate the vector field. However, even the worst result is still better than the result of C-PSO and NW-PSO. The alive rate shows that in this vector field still 15 percent of the population stays alive whereas all particles of the other approaches already run out of energy at iteration 75. The results of the other vector fields are even better so that NW-PSO is able to strongly reduce the energy usage for each objective function in each vector field. In consideration of the search quality and the energy results, NW-PSO is able to reduce the energy consumption in Rosenbrock function. However, the success rate is decreased to one-third. In Sphere and Ackley function, the NW-PSO approach is able to offer good search results in most of the vector fields while simultaneously reducing the energy consumption.

6. Conclusion and Outlook

In this chapter, the results of this thesis are summarized and suggestions for future work are offered. First, in Section 6.1 the goal of this work is mentioned as well as the tasks which were performed to reach it. Afterwards, a conclusion based on the evaluation of NW-PSO is presented regarding the previous mentioned goal. In the end, topics for future work are proposed.

6.1. Conclusion

The aim of this work was to develop a preferably energy efficient search mechanism for dynamic environments. To reach this goal, the following steps were performed. First, a model for the simulation of varying dynamic environments was created by using vector fields. This simulated a flow comparable to wind in real world scenarios. Additionally, the particles movement equation was modified so that the vector field influenced their movement. Afterwards, an energy consumption model was created including two parameters. The first one was *eTurn* which represented the energy for rotational movements of the particles to change their direction. The second parameter was *eMove* which measured the used energy for the particles movement in regard to the vector field. The combination of both parameters offered the total energy usage for a particles movement. Subsequently, a new approach called NW-PSO was developed. The goal of this thesis was to provide good search results and simultaneously be energy efficient. This goal was fulfilled by the new developed approach. As it was based on PSO, the standard PSO movement equation was modified with respect to the energy consumption. Furthermore, a correction vector was calculated considering possible target points for the particles movement. Each of these target points was evaluated in matters of its improvement of the search results and energy usage to move towards its position. This evaluation represented a multi-objective optimization problem which was solved by the WSM. The following simulation included two more approaches in order to compare their results with these of NW-PSO. One of the other approaches was standard PSO whereby the known PSO equation was modified so that the environment influenced the particles movement. The approach was called C-PSO. The other one was based on VFM-PSO which is known as CW-PSO in this thesis. It was already used in dynamic environments but without consideration of energy usages. The simulation was run in nine different vector fields and each of them was tested in three different objective functions. The first function was Sphere which was the easiest objective function. It was chosen in order to prove the approaches ability of optimization.

The second function was Rosenbrock which was chosen because of its challenge to find the optimum inside a narrow, parabolic valley. The third function was Ackley which provided multiple local optima which can mislead the search. In the following, the evaluation results of NW-PSO are presented with regard to the goal of this work to develop an energy efficient search mechanism without negligence of the search results quality.
The new NW-PSO approach was first tested in Sphere function to prove its correctness with regard to optimisation. As shown in Section 5.2.3, it successfully found the optimal solution in almost each vector field. Moreover, the convergence rate for vector field *Rotation* (Figure B.12), *Sheared* (Figure B.13) and *Tornado* (Figure B.5) shows that the swarm converged even faster at the optimum than CW-PSO. Another challenging function was Rosenbrock function because the optimum is placed inside a narrow, parabolic valley. Finding the valley is easy but converging to the global optimum is difficult. Nevertheless, NW-PSO reached desirable results in the vector field *Rotation* as in 80 of the simulations the optimum was found. In most of the other functions it was still successful in one-third of the simulations. The success rate for Ackley function is similar to the rate of Sphere function as the optimum is found in all vector fields except vector field *Cross* and *Vortex* in more than 85 percent of the simulations. In addition to the quality of search, the energy reduction was analyzed. The new approach is able to save energy for each objective function in each vector field. Even the worst energy rate of NW-PSO in the vector field *Rotation* are still better than the energy rates of CW-PSO and C-PSO. The results have shown that the new NW-PSO approach in all vector fields successfully reduces the energy usage during the search process. Therefore, in Sphere function without detriment of the search quality in most of the vector fields. Although the search results for Rosenbrock function are slightly worse, the approach gains very good results for Ackley function while simultaneously saving energy. The start position of the particles has a large impact on the quality of the search results. Thus, the approach is suitable in vector fields which allow the particles from any start position to move towards the optimum by following the flow as, for example, in rotational flows. This also increases the number of particles at the position of the optimum and therefore improves the search results especially in Rosenbrock function.

6.2. Outlook

First, the energy calculation in this thesis was abstracted. More specific energy measurements deflected by real world scenarios could be applied. For example energy measurements of a real micro-robot could be implemented in order to make it possible to test the approach with real robots. Secondly, not all energy parameters were regarded as for example landing and launching. Additionally, the implementation could be modified so that the particles quit their work if they have not changed their position for a predefined amount of iterations. In most cases the particles will not find better solutions or even have already found the global optimum if they do not move any longer. However, the results in matters of energy usage will be interesting especially if C-PSO, CW-PSO and NW-PSO are compared with each other. Another research topic are different vector fields and objective functions. As mentioned, the success of NW-PSO is highly depending on the objective function and the applied vector field. Therefore, further vector fields and objective functions can be tested to develop a sample in which cases NW-PSO is more successful. Moreover, the results presented in A.2 show that often only a small percentage of the NW-PSO population stays stable at the best found solution. As a consequence, the chance that the swarm is mislead in challenging fitness landscapes is high. Additionally, finding the global optimum inside a plateau is also challenging. A new approach would be that this percentage is taken into consideration in the decision making process of the particles. An opportunity would be that the particles decision is stronger driven to the improvement of the search than to the reduction of the energy usage if only a small number of particles has converged. Additionally, angle range for the search of possible target points could be increased. As a result, the quality of the search results could be improved.

Appendix

A. Tables

A.1. NW-PSO Weights

Table A.1.: WSM Weights

	2D Pareto		3D Pareto		
	w_1	w_2	w_1	w_2	w_3
Cross	0.3	0.7	0.1	0.6	0.3
Rotation	0.7	0.3	0.2	0.3	0.5
Sheared	0.7	0.3	0.3	0.4	0.3
Wave	0.3	0.7	0.2	0.5	0.2
Tornado	0.2	0.8	0.2	0.6	0.2
Bi-Directional	0.3	0.7	0.6	0.3	0.1
Random	0.9	0.1	0.8	0.1	0.1
Multi-Rotation	0.9	0.1	0.8	0.1	0.1
Vortex	0.8	0.2	0.8	0.1	0.1

A.2. NW-PSO Particles at Global Best Solution

Table A.2.: NW-PSO Particles at Global Best Solution

	Sphere	Rosenbrock	Ackley
Cross	6%	6%	6%
Rotation	94%	88%	94%
Sheared	27%	25%	28%
Wave	22%	22%	22%
Tornado	30%	26%	30%
Bi-Directional	28%	27%	27%
Random	25%	23%	24%
Multi-Rotation	33%	29%	35%
Vortex	6%	13%	8%

B. Plots

B.1. Vector Fields

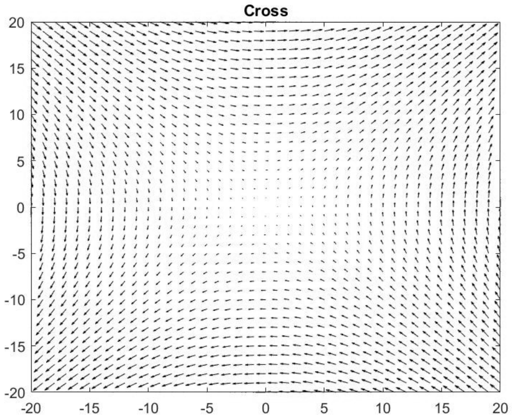


Figure B.1.: Cross

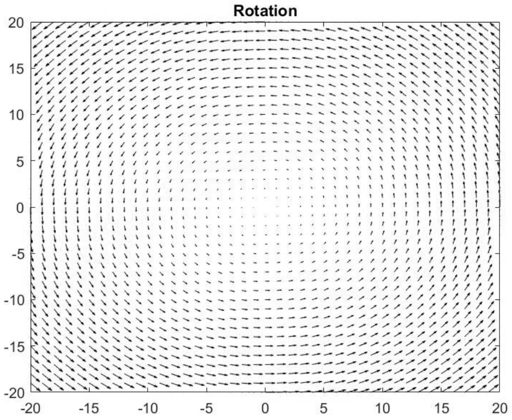


Figure B.2.: Rotation

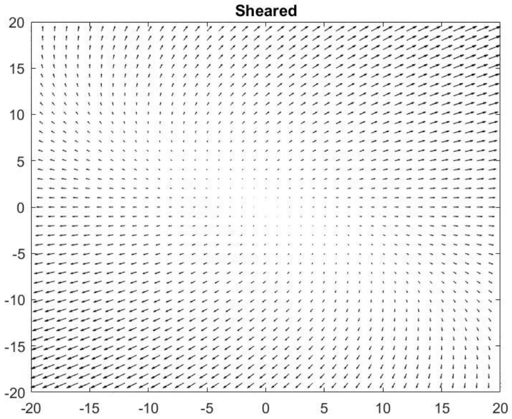


Figure B.3.: Sheared

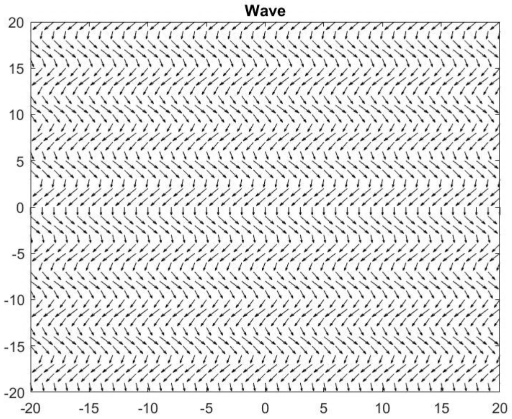


Figure B.4.: Wave

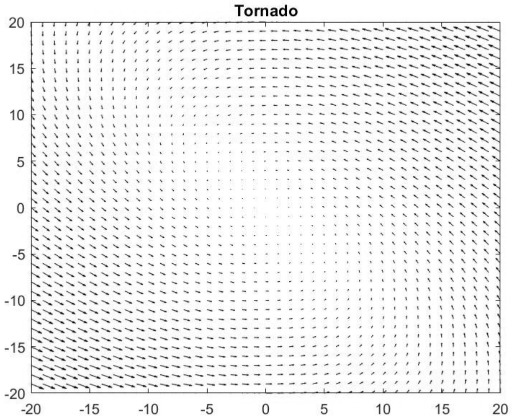


Figure B.5.: Tornado

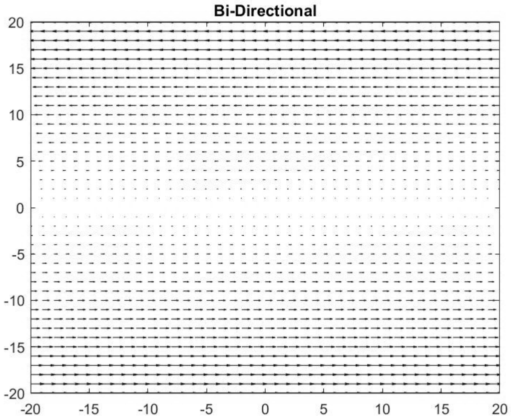


Figure B.6.: Bi-Directional

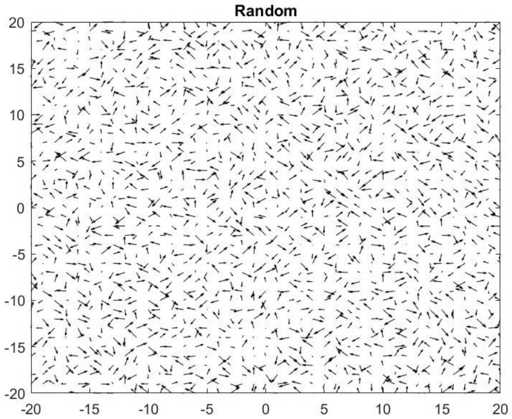


Figure B.7.: Random

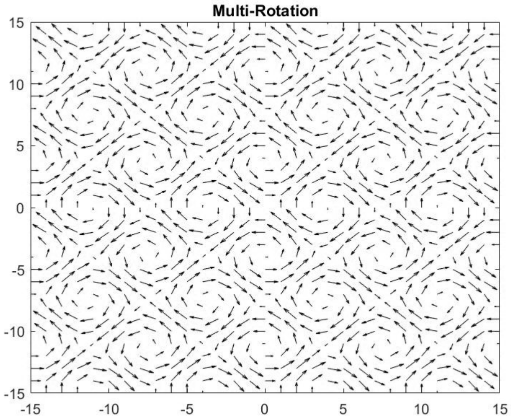


Figure B.8.: Multi-Rotation

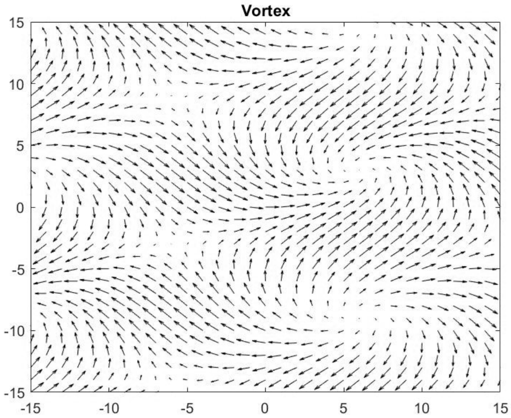


Figure B.9.: Vortex

B.2. Convergence Plots

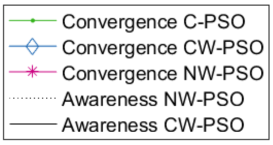


Figure B.10.: Legend Convergence Plots

B.2.1. Sphere Function

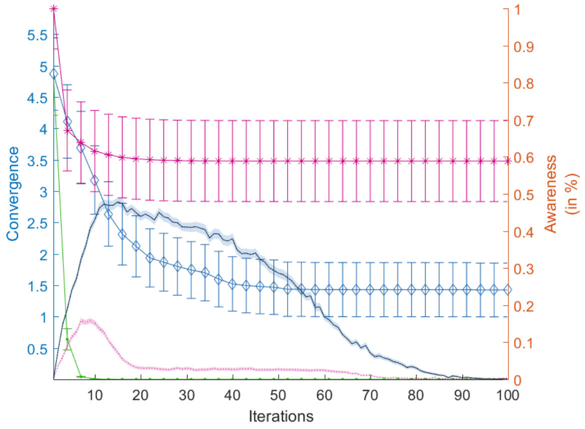


Figure B.11.: Convergence Sphere Cross

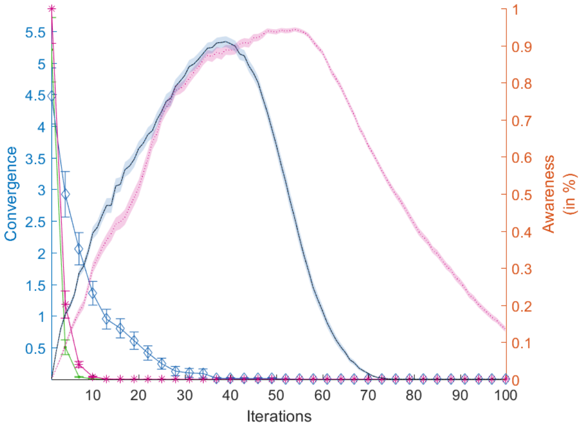


Figure B.12.: Convergence Sphere Rotation

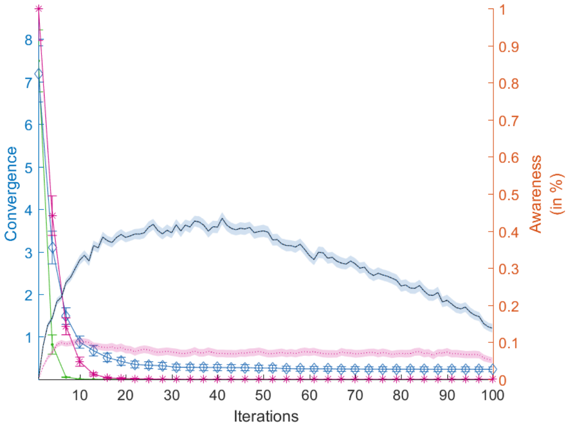


Figure B.13.: Convergence Sphere Sheared

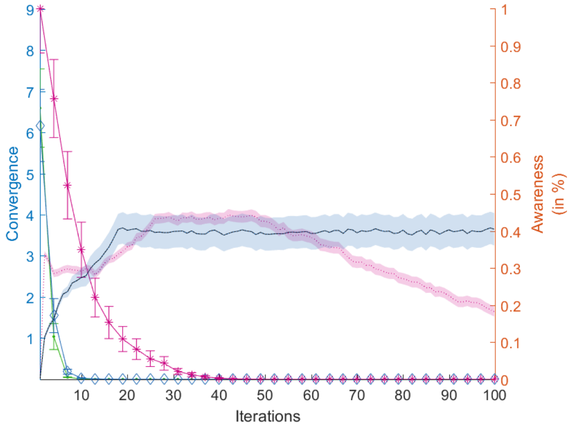


Figure B.14.: Convergence Sphere Wave

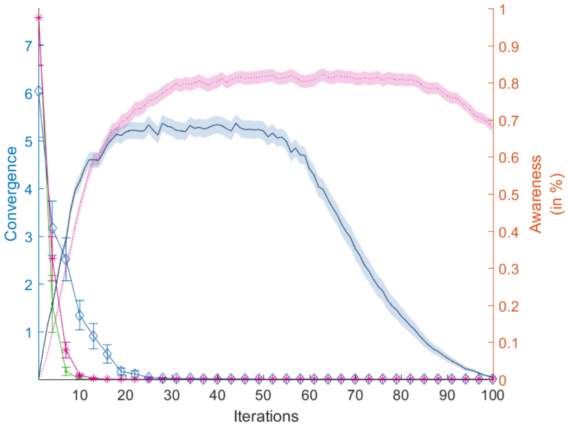


Figure B.15.: Convergence Sphere Tornado

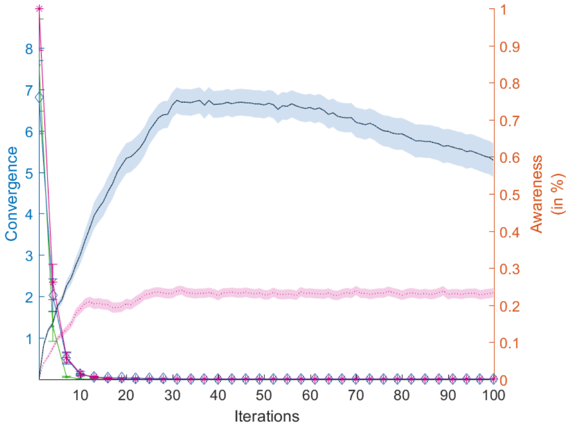


Figure B.16.: Convergence Sphere Bi-Directional

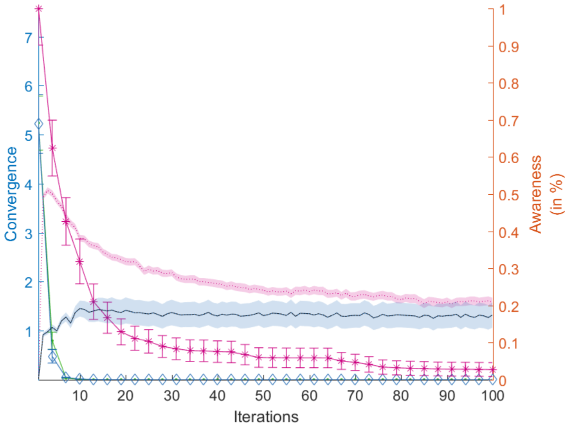


Figure B.17.: Convergence Sphere Random

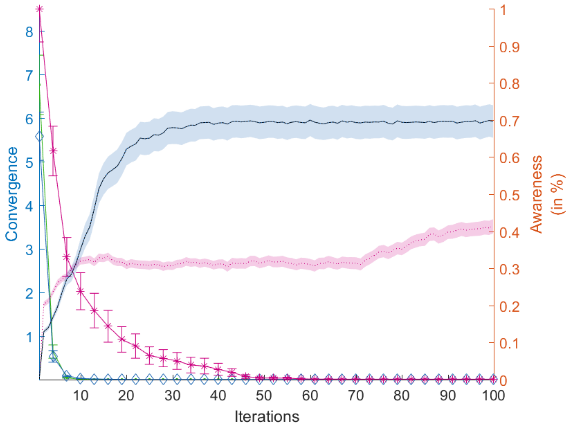


Figure B.18.: Convergence Sphere Multi-Rotation

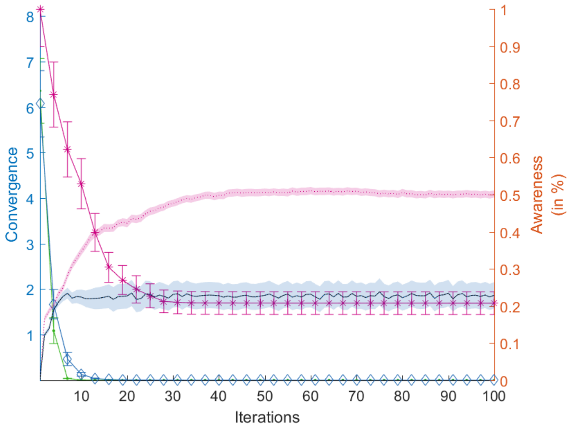


Figure B.19.: Convergence Sphere Vortex

B.2.2. Rosenbrock Function

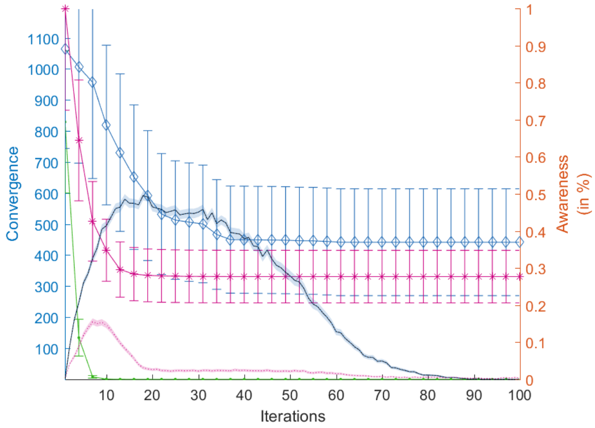


Figure B.20.: Convergence
Rosenbrock Cross

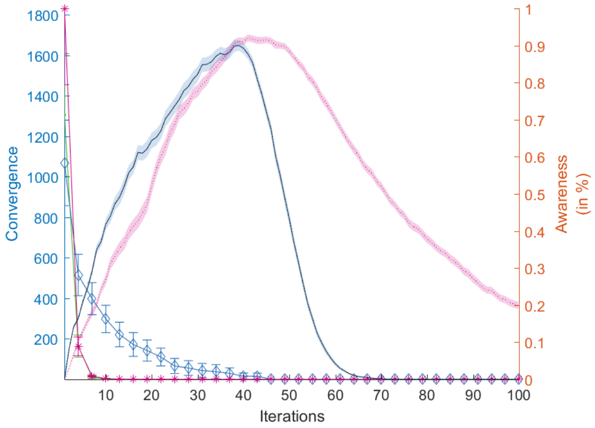


Figure B.21.: Convergence
Rosenbrock Rotation

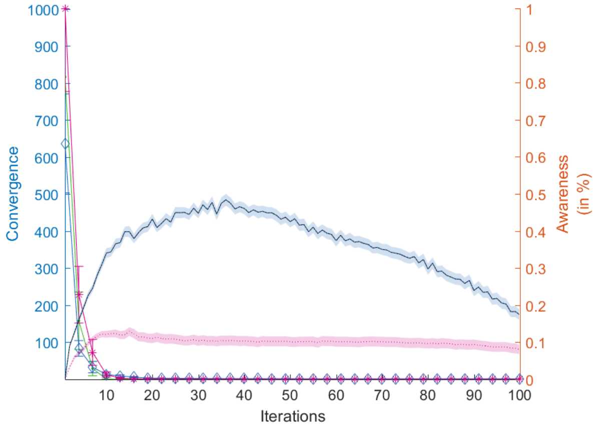


Figure B.22.: Convergence
Rosenbrock Sheared

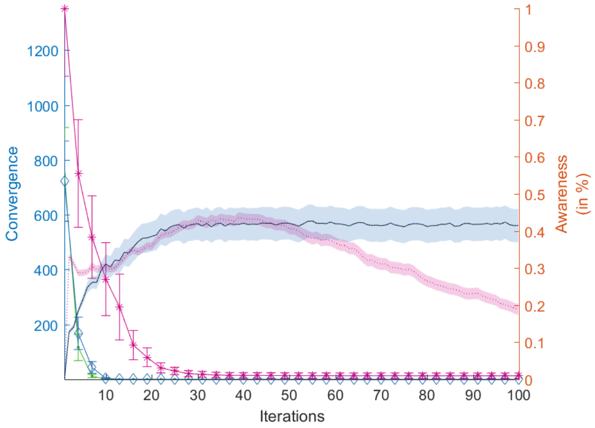


Figure B.23.: Convergence
Rosenbrock Wave

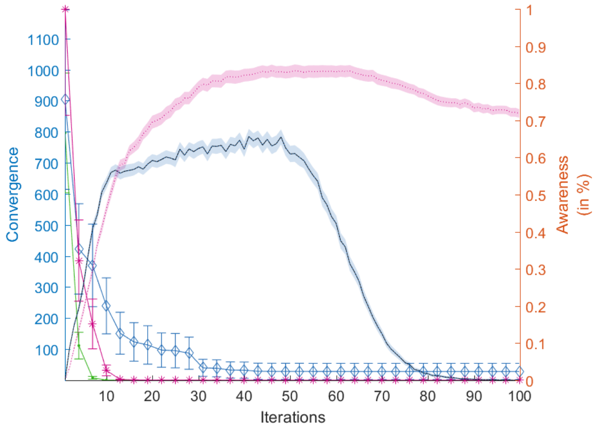


Figure B.24.: Convergence
Rosenbrock Tornado

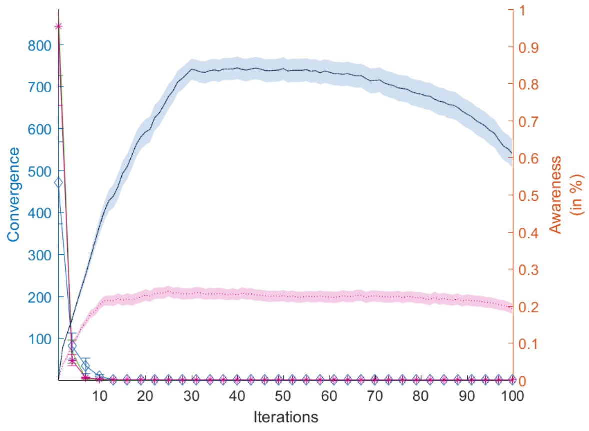


Figure B.25.: Convergence
Rosenbrock Bi-Directional

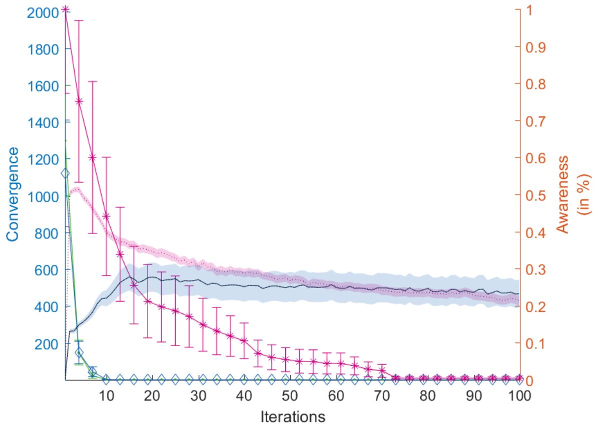


Figure B.26.: Convergence
Rosenbrock Random

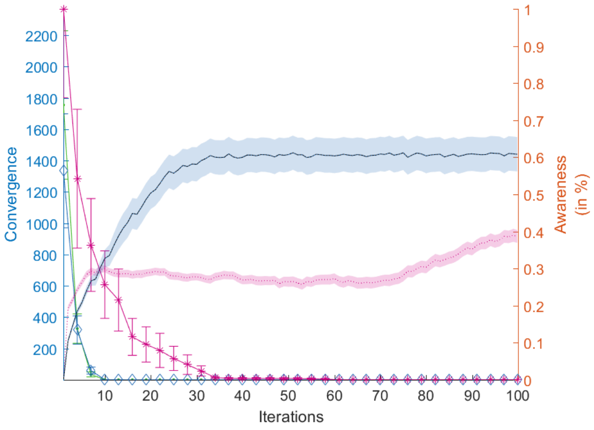


Figure B.27.: Convergence
Rosenbrock Multi-Rotation

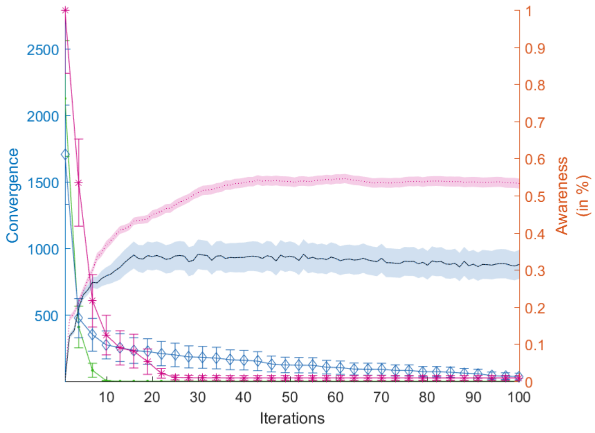


Figure B.28.: Convergence
Rosenbrock Vortex

B.2.3. Ackley Function

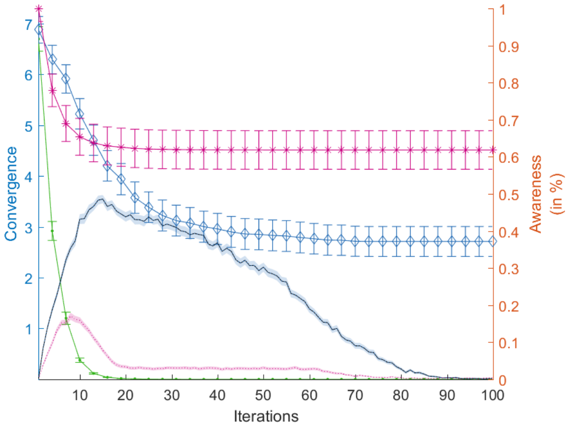


Figure B.29.: Convergence Ackley Cross

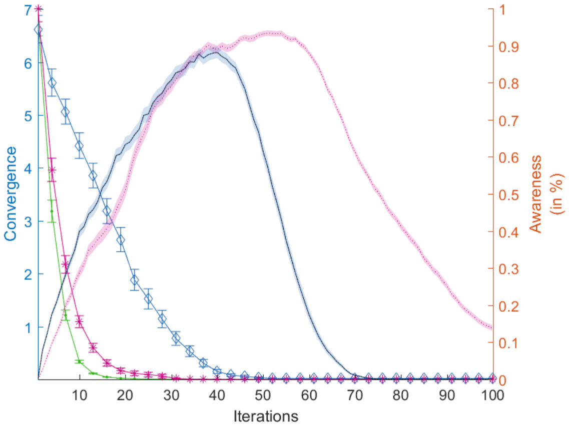


Figure B.30.: Convergence Ackley Rotation

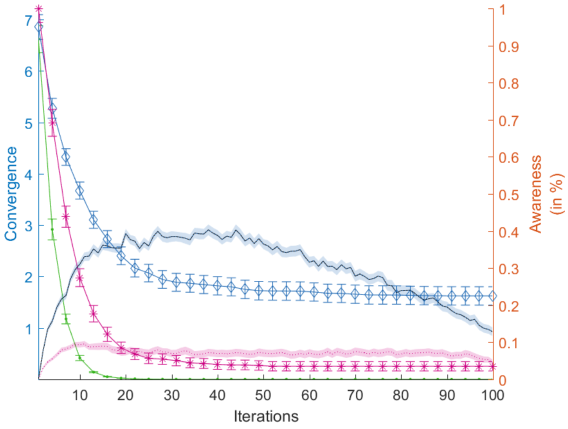


Figure B.31.: Convergence Ackley Sheared

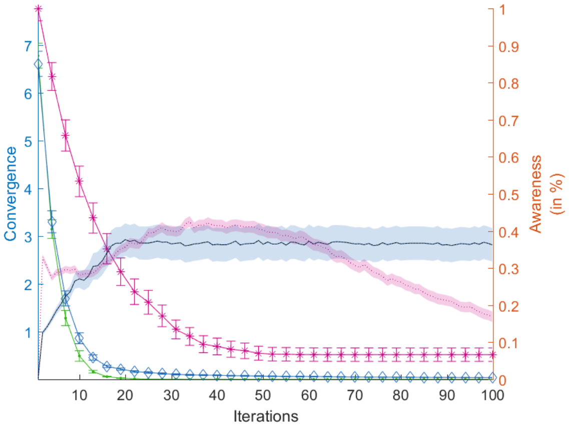


Figure B.32.: Convergence Ackley Wave

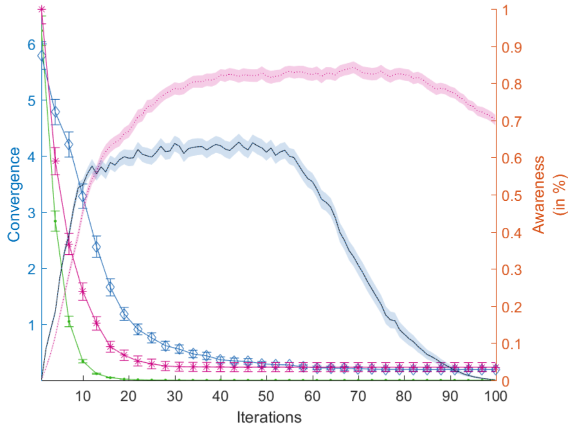


Figure B.33.: Convergence
Ackley Tornado

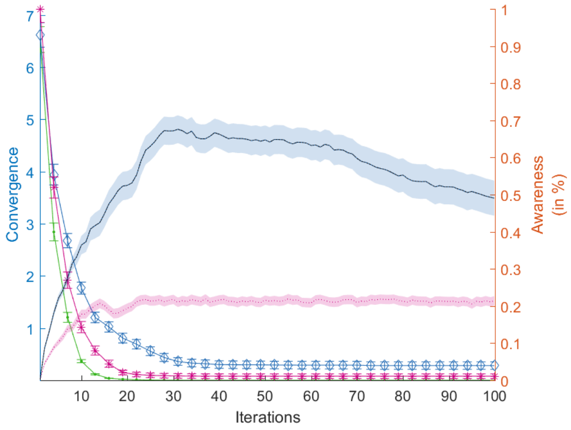


Figure B.34.: Convergence
Ackley Bi-Directional

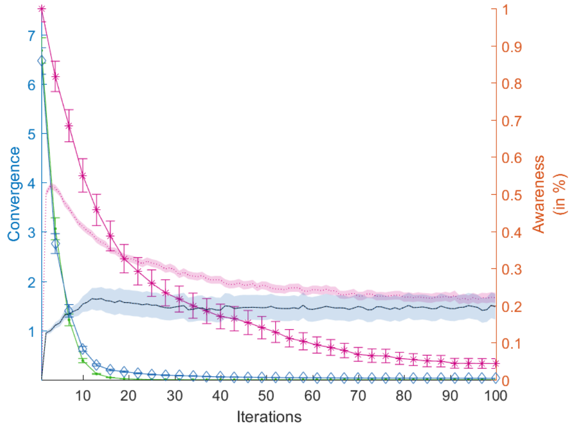


Figure B.35.: Convergence
Ackley Random

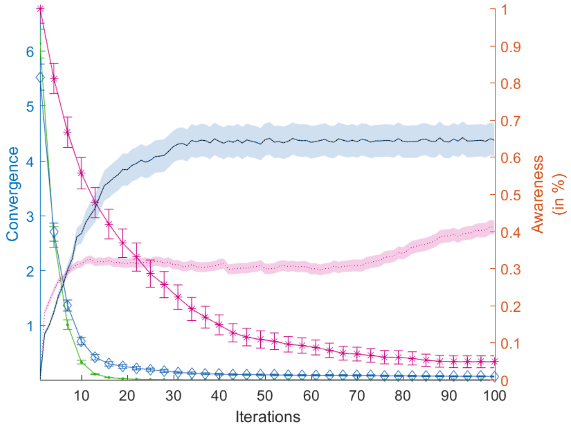


Figure B.36.: Convergence
Ackley Multi-Rotation

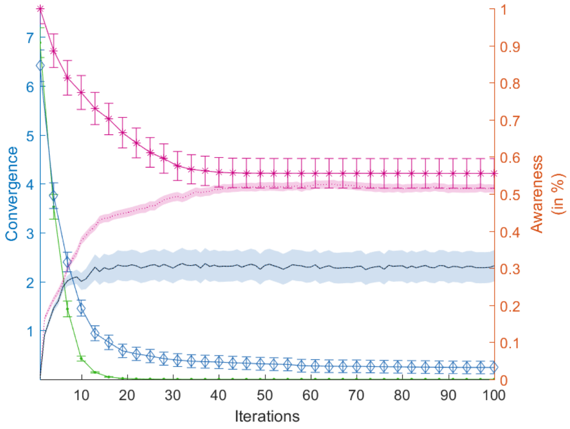


Figure B.37.: Convergence
Ackley Vortex

B.3. Alive Plots

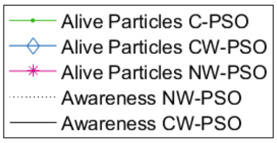


Figure B.38.: Legend Alive Plots

B.3.1. Sphere Function

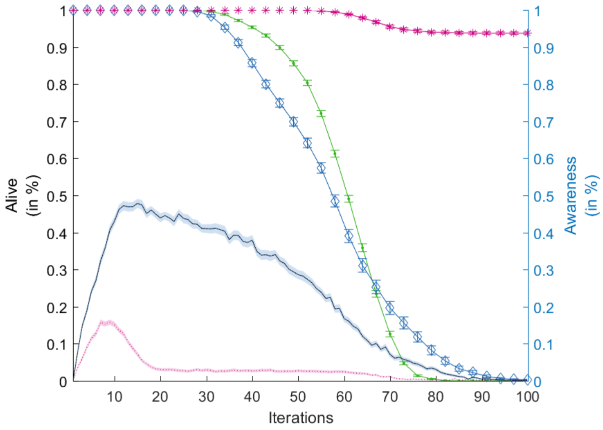


Figure B.39.: Alive
Sphere Cross

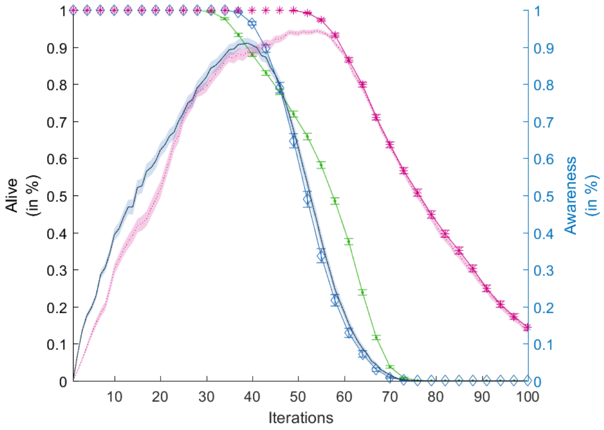


Figure B.40.: Alive
Sphere Rotation

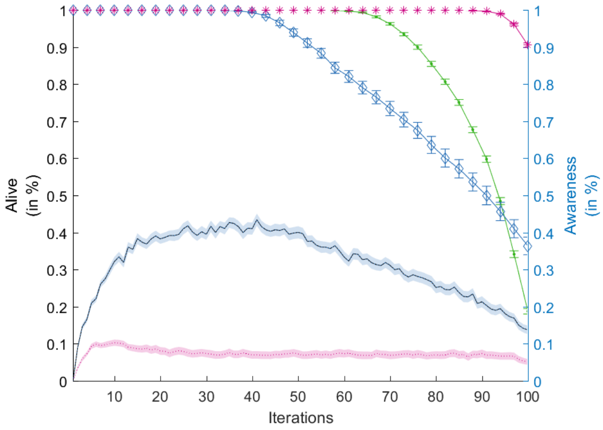


Figure B.41.: Alive
Sphere Sheared

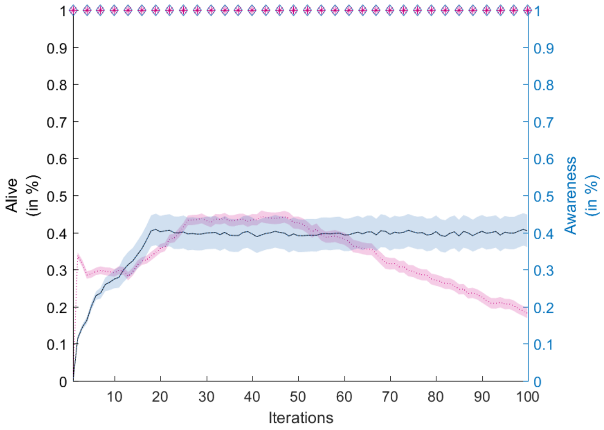


Figure B.42.: Alive
Sphere Wave

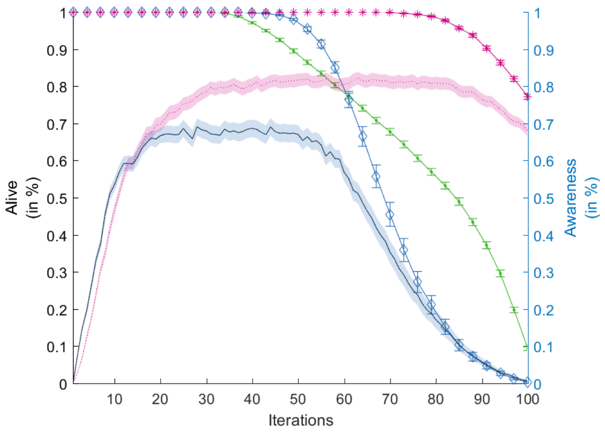


Figure B.43.: Alive
Sphere Tornado

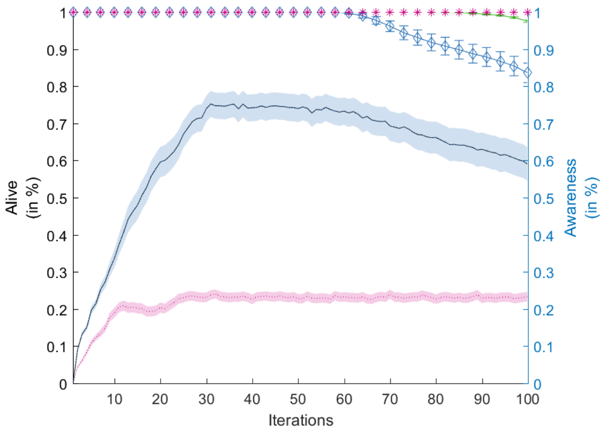


Figure B.44.: Alive
Sphere Bi-Directional

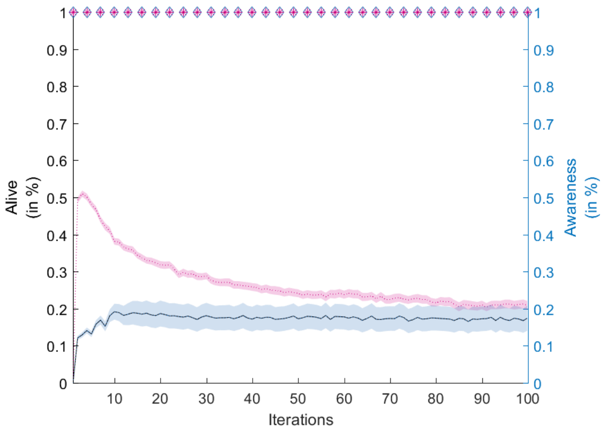


Figure B.45.: Alive
Sphere Random

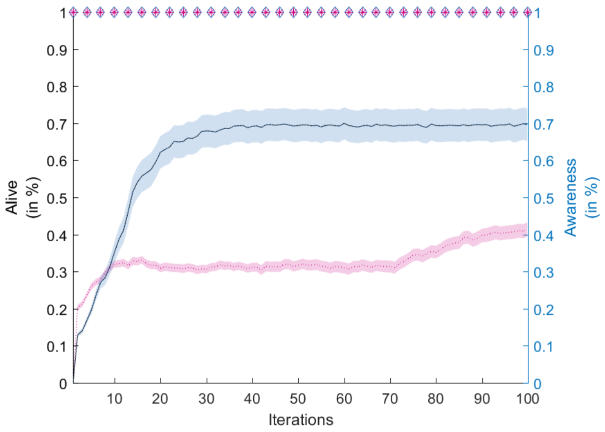


Figure B.46.: Alive
Sphere Multi-Rotation

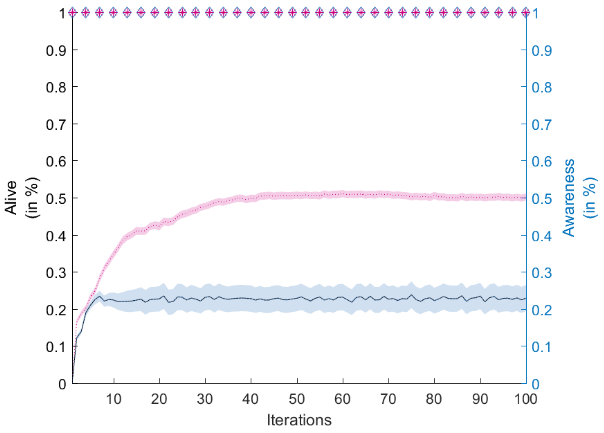


Figure B.47.: Alive
Sphere Vortex

B.3.2. Rosenbrock Function

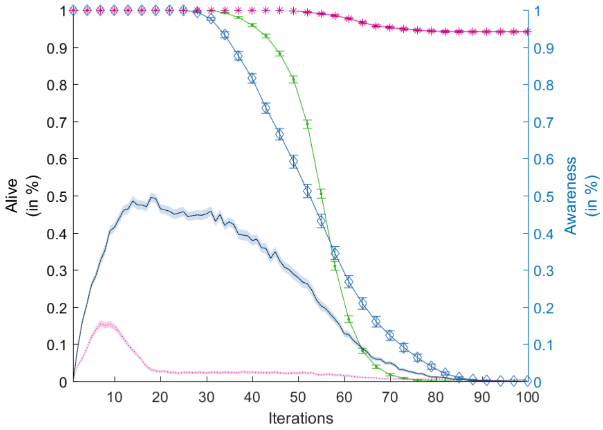


Figure B.48.: Alive
Rosenbrock Cross

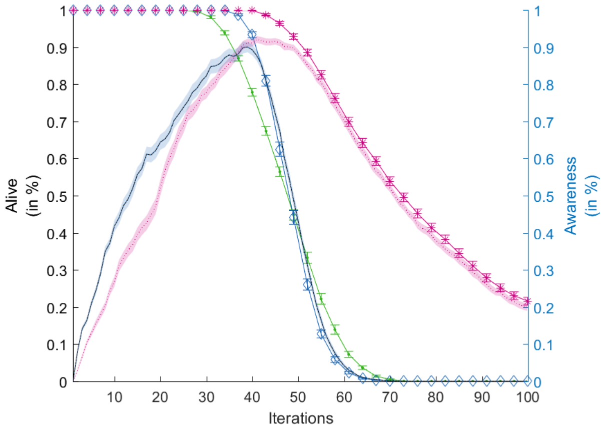


Figure B.49.: Alive
Rosenbrock Rotation

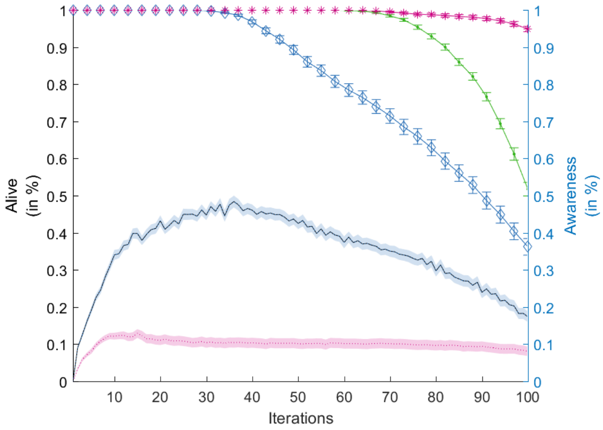


Figure B.50.: Alive
Rosenbrock Sheared

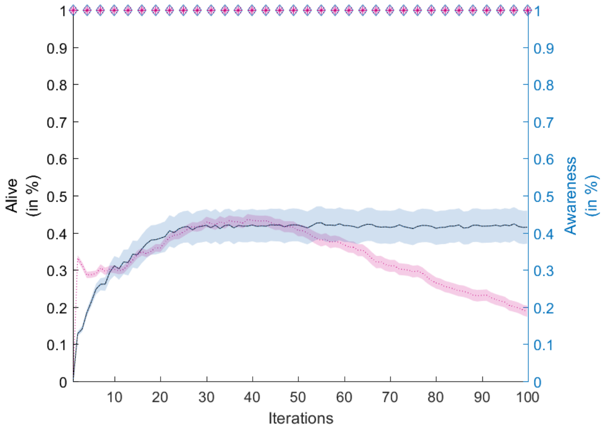


Figure B.51.: Alive
Rosenbrock Wave

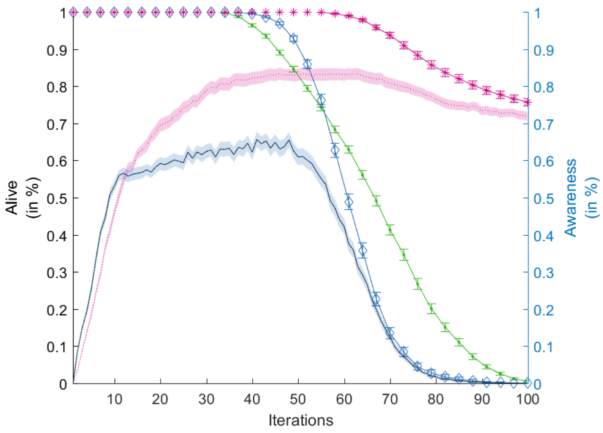


Figure B.52.: Alive
Rosenbrock Tornado

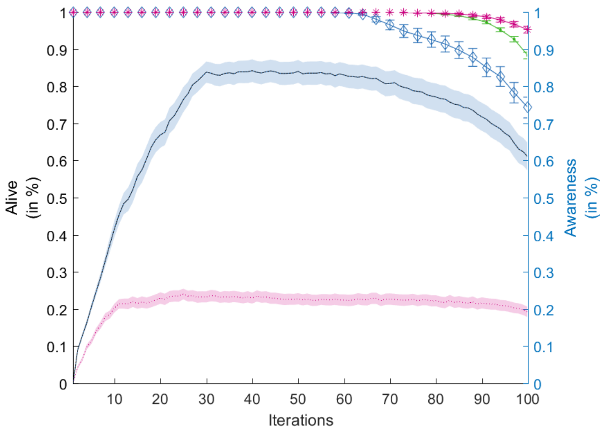


Figure B.53.: Alive
Rosenbrock Bi-Directional

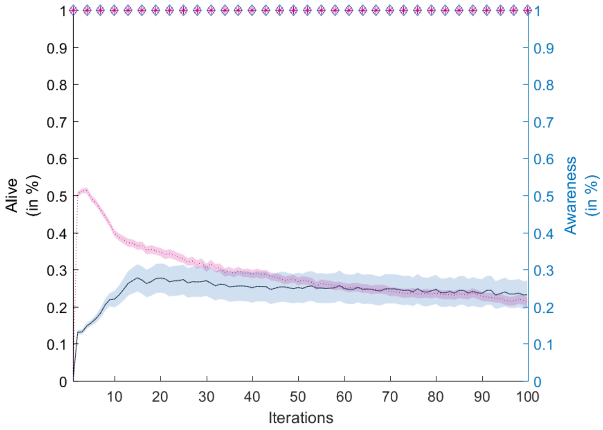


Figure B.54.: Alive
Rosenbrock Random

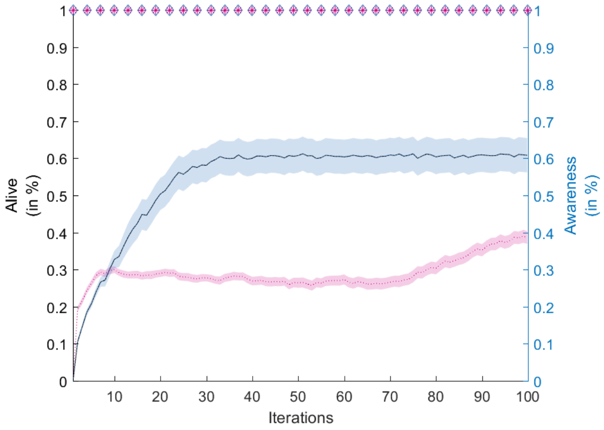


Figure B.55.: Alive
Rosenbrock Multi-Rotation

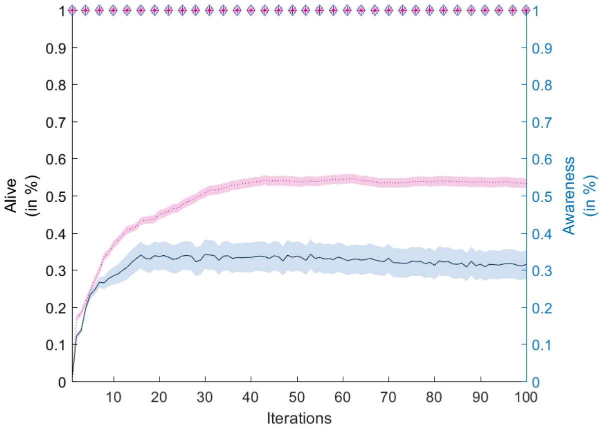


Figure B.56.: Alive
Rosenbrock Vortex

B.3.3. Ackley Function

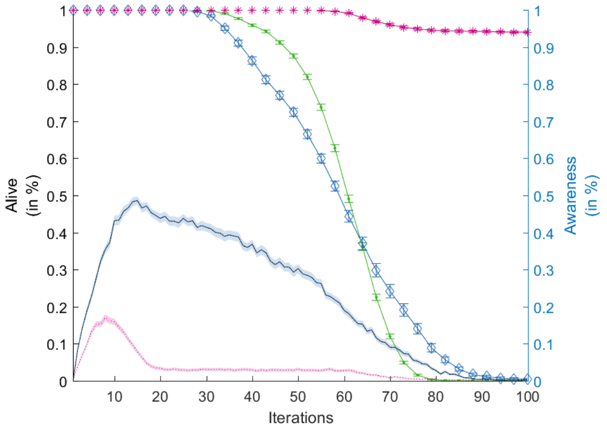


Figure B.57.: Alive
Ackley Cross

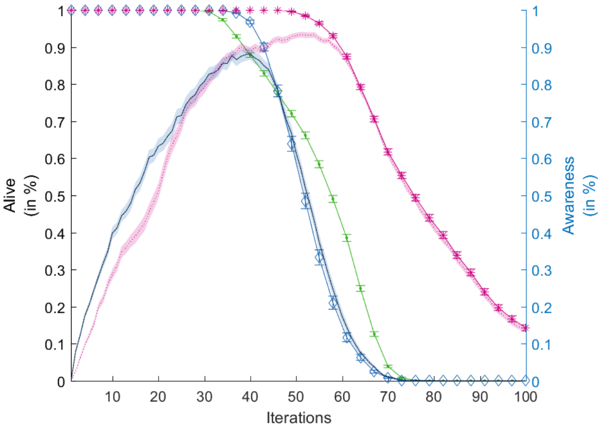


Figure B.58.: Alive
Ackley Rotation

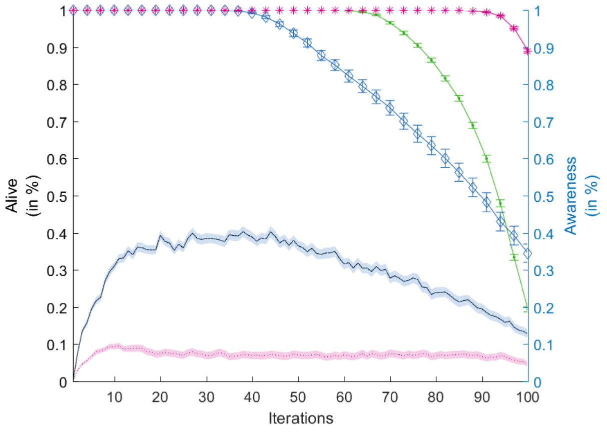


Figure B.59.: Alive
Ackley Sheared

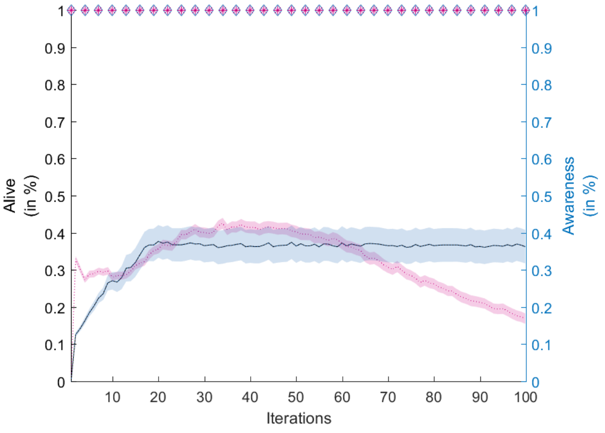


Figure B.60.: Alive
Ackley Wave

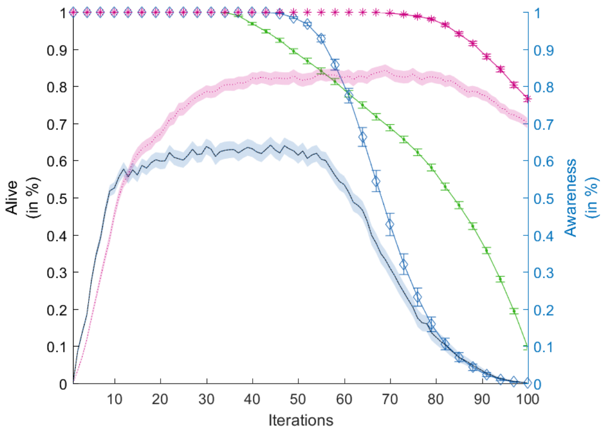


Figure B.61.: Alive
Ackley Tornado

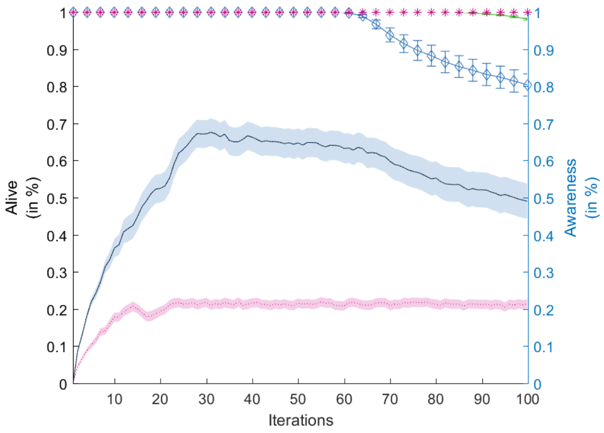


Figure B.62.: Alive
Ackley Bi-Directional

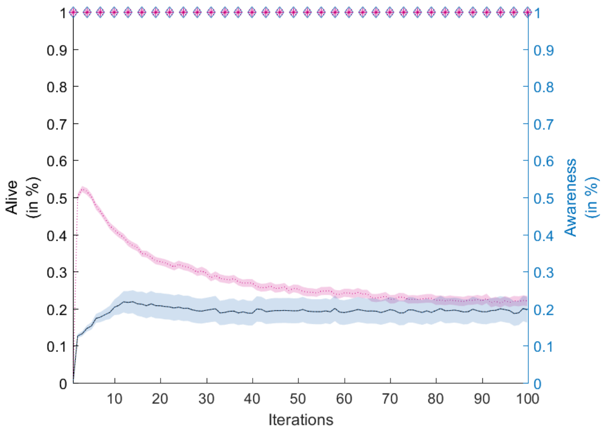


Figure B.63.: Alive
Ackley Random

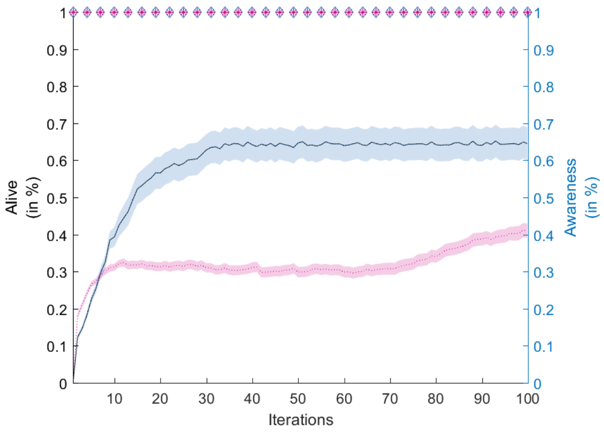


Figure B.64.: Alive
Ackley Multi-Rotation

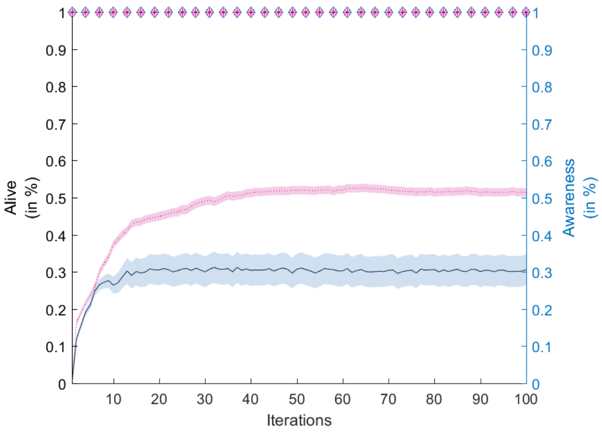


Figure B.65.: Alive
Ackley Vortex

B.4. Energy Plots

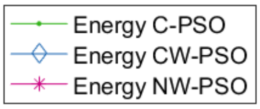


Figure B.66.: Legend Energy Plots

B.4.1. Sphere Function

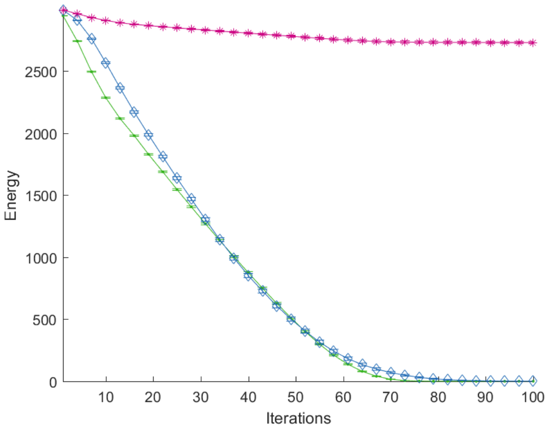


Figure B.67.: Energy Sphere Cross

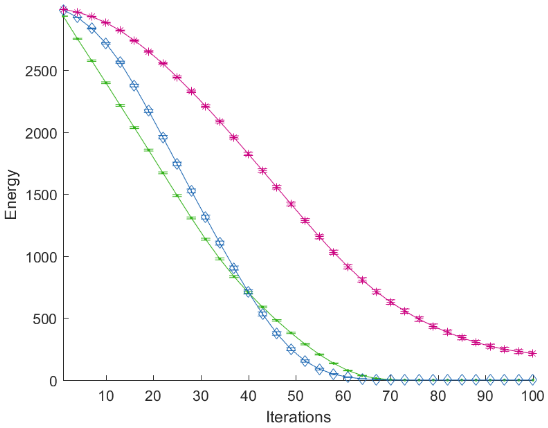


Figure B.68.: Energy Sphere Rotation

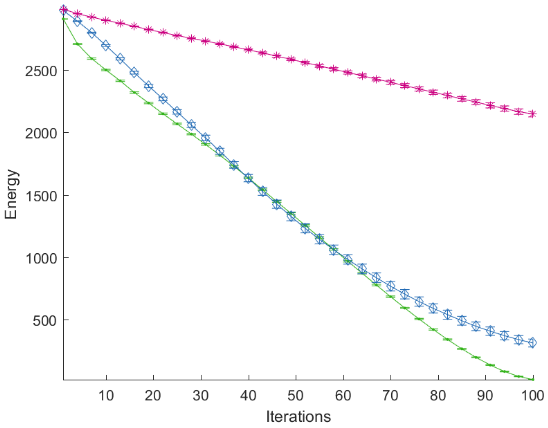


Figure B.69.: Energy Sphere Sheared

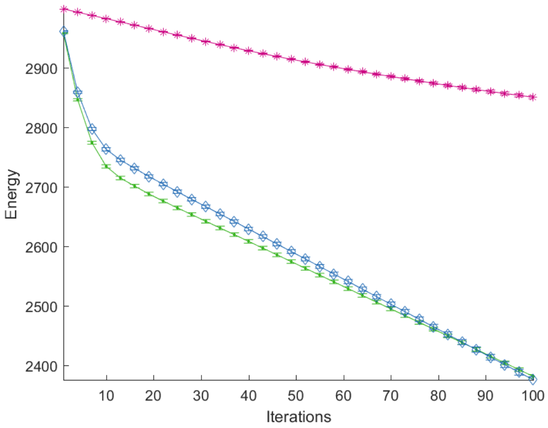


Figure B.70.: Energy Sphere Wave

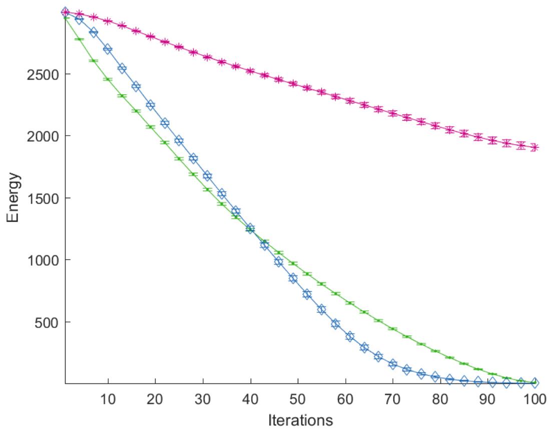


Figure B.71.: Energy
Sphere Tornado

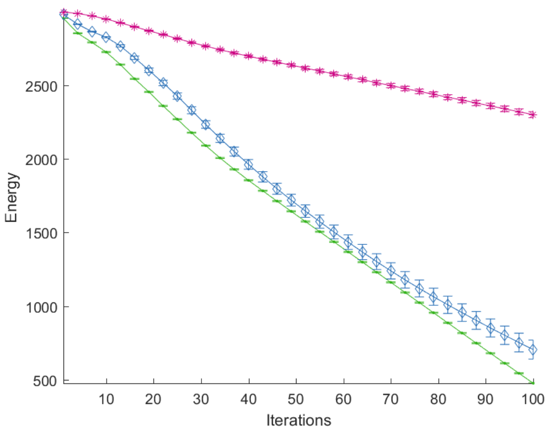


Figure B.72.: Energy
Sphere Bi-Directional

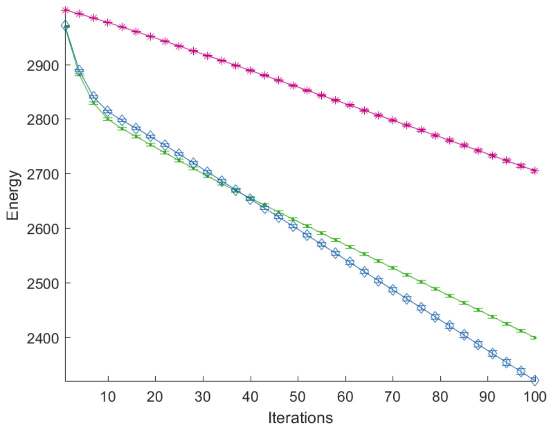


Figure B.73.: Energy
Sphere Random

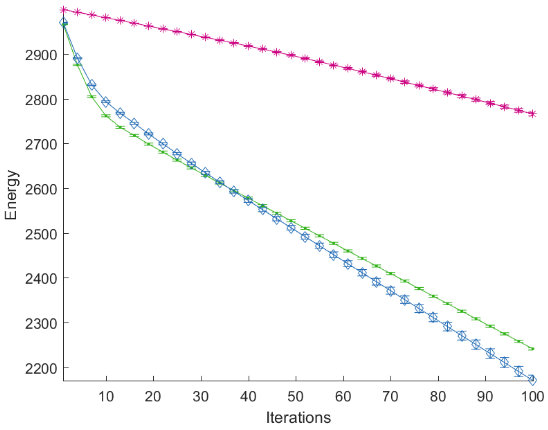


Figure B.74.: Energy
Sphere Multi-Rotation

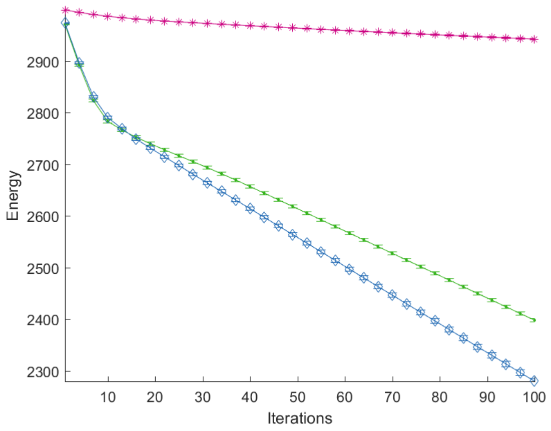


Figure B.75.: Energy
Sphere Vortex

B.4.2. Rosenbrock Function

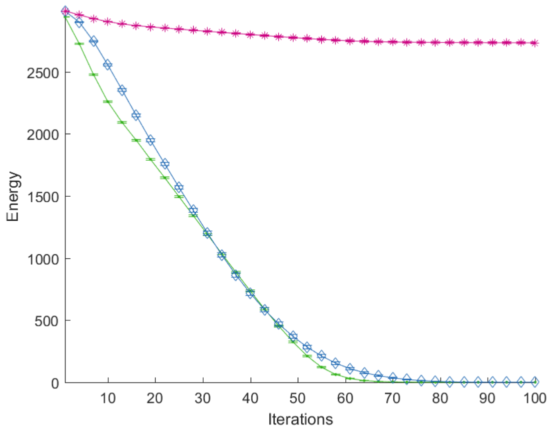


Figure B.76.: Energy Rosenbrock Cross

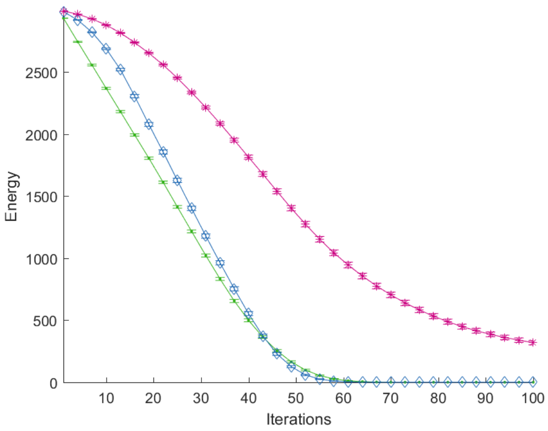


Figure B.77.: Energy Rosenbrock Rotation

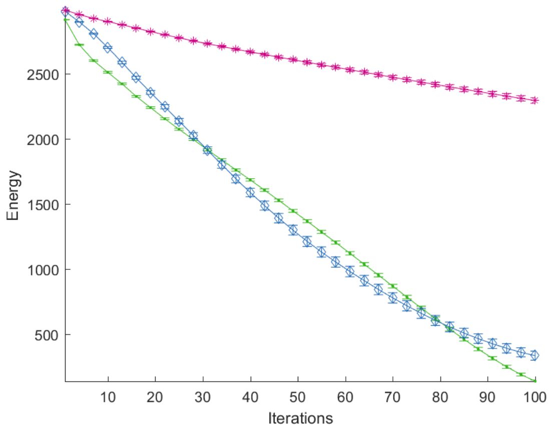


Figure B.78.: Energy Rosenbrock Sheared

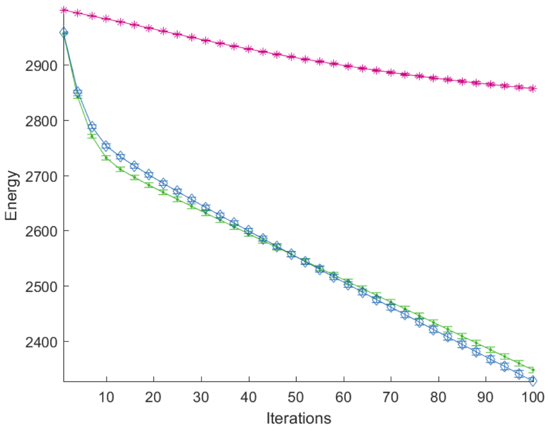


Figure B.79.: Energy Rosenbrock Wave

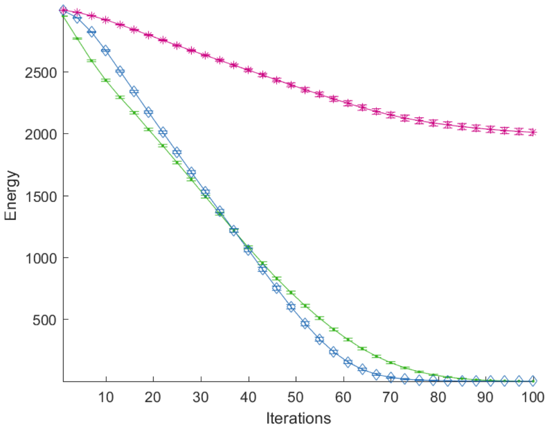


Figure B.80.: Energy
Rosenbrock Tornado

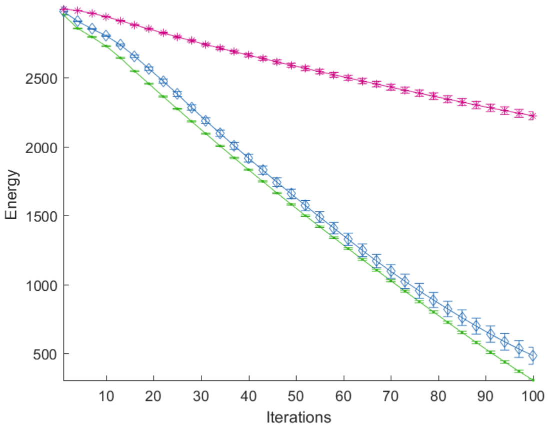


Figure B.81.: Energy
Rosenbrock Bi-Directional

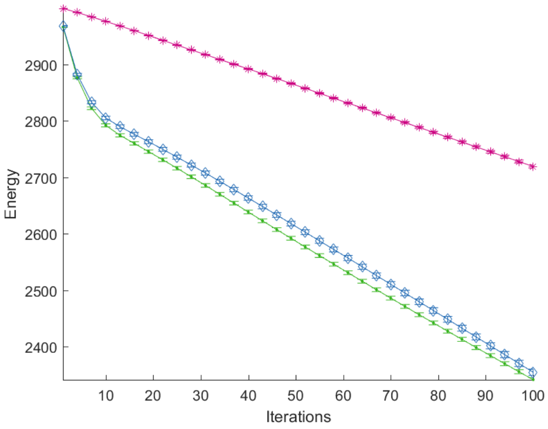


Figure B.82.: Energy
Rosenbrock Random

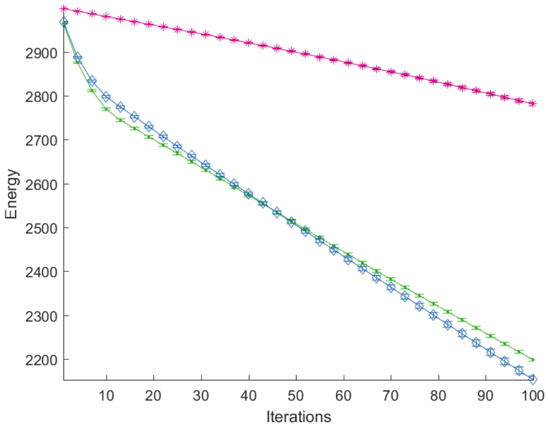


Figure B.83.: Energy
Rosenbrock Multi-Rotation

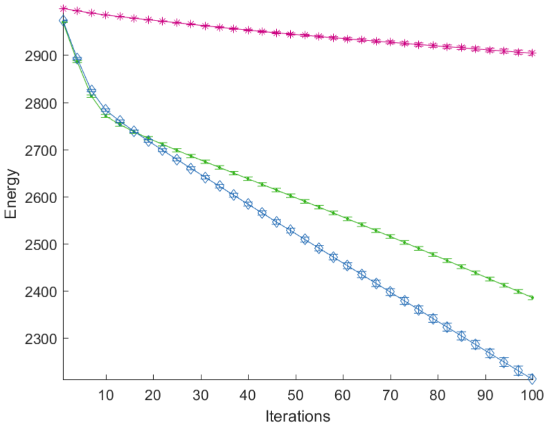


Figure B.84.: Energy
Rosenbrock Vortex

B.4.3. Ackley Function

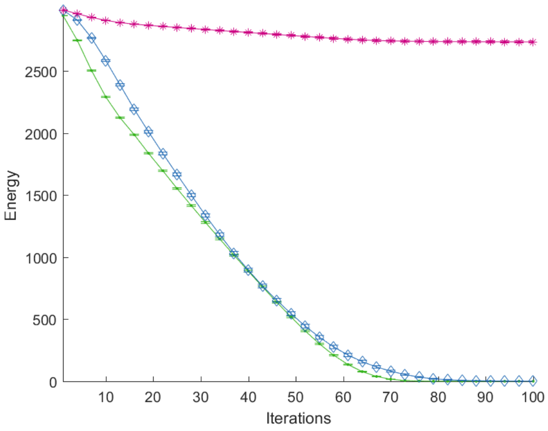


Figure B.85.: Energy
Ackley Cross

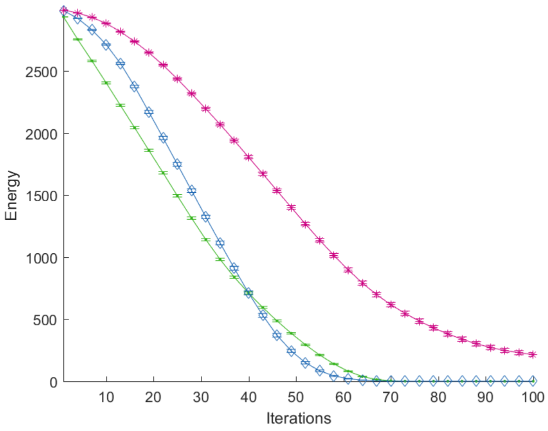


Figure B.86.: Energy
Ackley Rotation

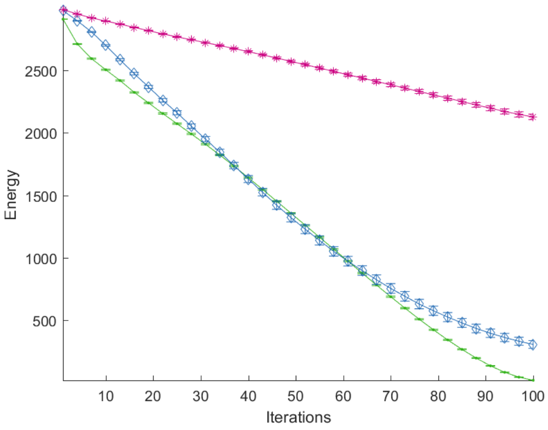


Figure B.87.: Energy
Ackley Sheared

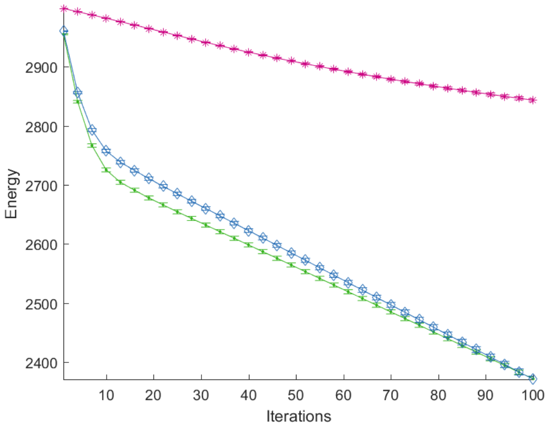


Figure B.88.: Energy
Ackley Wave

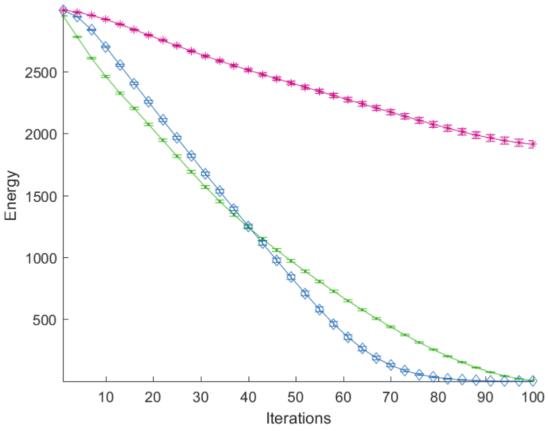


Figure B.89.: Energy
Ackley Tornado

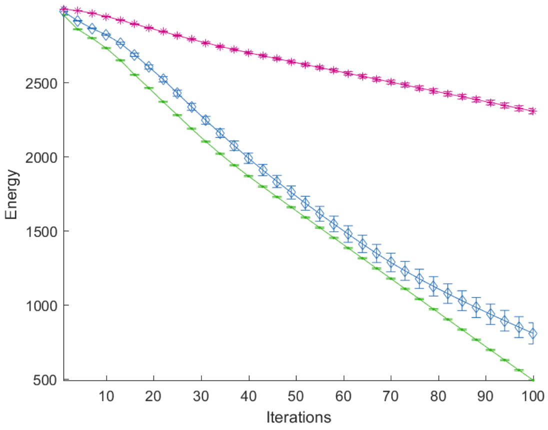


Figure B.90.: Energy
Ackley Bi-Directional

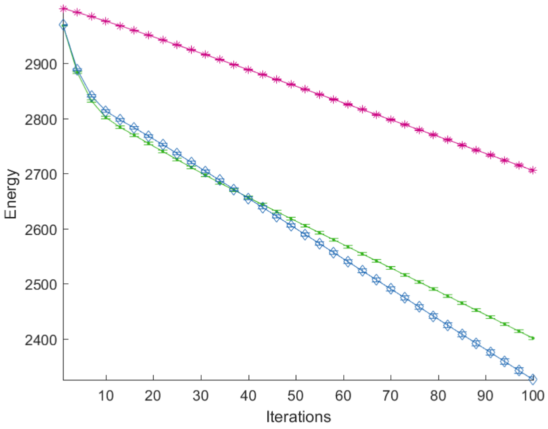


Figure B.91.: Energy
Ackley Random

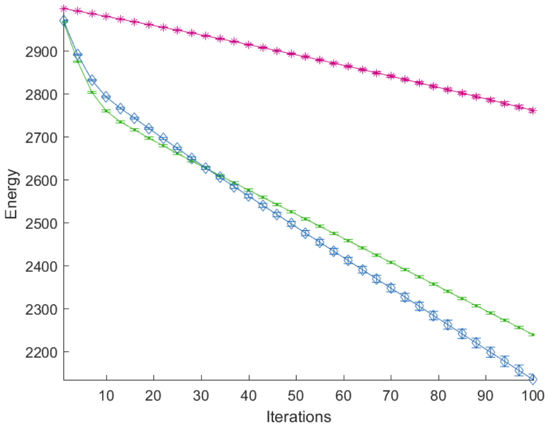


Figure B.92.: Energy
Ackley Multi-Rotation

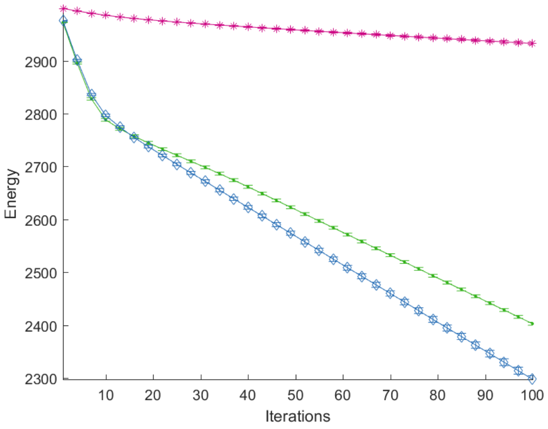


Figure B.93.: Energy
Ackley Vortex

Bibliography

[BB04] Tim Blackwell and Jürgen Branke. "Multi-swarm Optimization in Dynamic Environments". In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2004, pp. 489–500. DOI: 10.1007/978-3-540-24653-4_50.

[BGM17] Palina Bartashevich, Luigi Grimaldi, and Sanaz Mostaghim. "PSO-based Search mechanism in dynamic environments: Swarms in Vector Fields". In: *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 1263–1270. DOI: 10.1109/cec.2017.7969450.

[BL08] Christian Blum and Xiaodong Li. "Swarm Intelligence in Optimization". In: *Natural Computing Series*. Springer Berlin Heidelberg, 2008, pp. 43–85. DOI: 10.1007/978-3-540-74089-6_2.

[EK95] R. Eberhart and J. Kennedy. "A new optimizer using particle swarm theory". In: *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*. IEEE, 1995, pp. 39–43. DOI: 10.1109/mhs.1995.494215.

[Eng07] Andries P. Engelbrecht. *Computational Intelligence*. John Wiley & Sons, Ltd, 2007. DOI: 10.1002/9780470512517.

[ESK01] Russell C Eberhart, Yuhui Shi, and James Kennedy. *Swarm intelligence*. Elsevier, 2001.

[GAO06] B. Garau, A. Alvarez, and G. Oliver. "AUV navigation through turbulent ocean environments supported by onboard H-ADCP". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006. DOI: 10.1109/robot.2006.1642245.

[GCT15] Nuwan Ganganath, Chi-Tsun Cheng, and Chi K. Tse. "A Constraint-Aware Heuristic Path Planner for Finding Energy-Efficient Paths on Uneven Terrains". In: *IEEE Transactions on Industrial Informatics*. Vol. 11. 3. Institute of Electrical and Electronics Engineers (IEEE), 2015, pp. 601–611. DOI: 10.1109/tii.2015.2413355.

[GR06] Oleg Grodzevich and Oleksandr Romanko. "Normalization and other topics in multi-objective optimization". In: *Proceedings of the Fields–MITACS Industrial Problems Workshop*. 2006.

[HBM13] Sabine Helwig, Juergen Branke, and Sanaz Mostaghim. "Experimental Analysis of Bound Handling Techniques in Particle Swarm Optimization". In: *IEEE Transactions on Evolutionary Computation.* Vol. 17. 2. Institute of Electrical and Electronics Engineers (IEEE), 2013, pp. 259–271. DOI: `10.1109/tevc.2012.2189404`.

[Jat+08] Wisnu Jatmiko, Petrus Mursanto, Benyamin Kusumoputro, Kosuke Sekiyama, and Toshio Fukuda. "Modified PSO algorithm based on flow of wind for odor source localization problems in dynamic environments". In: *WSEAS Transaction on System.* Vol. 7. 3. 2008, pp. 106–113.

[JSF07] W. Jatmiko, K. Sekiyama, and T. Fukuda. "A pso-based mobile robot for odor source localization in dynamic advection-diffusion with obstacles environment: theory, simulation and measurement". In: *IEEE Computational Intelligence Magazine.* Vol. 2. 2. Institute of Electrical and Electronics Engineers (IEEE), 2007, pp. 37–51. DOI: `10.1109/mci.2007.353419`.

[KE95] J. Kennedy and R. Eberhart. "Particle swarm optimization". In: *Proceedings of ICNN'95 - International Conference on Neural Networks.* IEEE, 1995. DOI: `10.1109/icnn.1995.488968`.

[LS17] Fran Sergio Lobato and Valder Steffen. *Multi-Objective Optimization Problems.* Springer International Publishing, 2017. DOI: `10.1007/978-3-319-58565-9`.

[Mei+04] Yongguo Mei, Yung-Hsiang Lu, Y.C. Hu, and C.S.G. Lee. "Energy-efficient motion planning for mobile robots". In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004.* IEEE, 2004. DOI: `10.1109/robot.2004.1302401`.

[Pur+14] Robin C. Purshouse, Kalyanmoy Deb, Maszatul M. Mansor, Sanaz Mostaghim, and Rui Wang. "A review of hybrid evolutionary multiple criteria decision making methods". In: *2014 IEEE Congress on Evolutionary Computation (CEC).* IEEE, 2014. DOI: `10.1109/cec.2014.6900368`.

[RP09] P. Raja and S. Pugazhenthi. "Path Planning for Mobile Robots in Dynamic Environments Using Particle Swarm Optimization". In: *2009 International Conference on Advances in Recent Technologies in Communication and Computing.* IEEE, 2009. DOI: `10.1109/artcom.2009.24`.

[RTL13] Vincent Roberge, Mohammed Tarbouchi, and Gilles Labonte. "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning". In: *IEEE Transactions on Industrial Informatics.* Vol. 9. 1. Institute of Electrical and Electronics Engineers (IEEE), 2013, pp. 132–141. DOI: `10.1109/tii.2012.2198665`.

[RZF08] J.F. Roberts, J.-C. Zufferey, and D. Floreano. “Energy management for indoor hovering robots”. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2008. DOI: 10.1109/iros.2008.4650856.

[SH17] Ahmed T. Sadiq and Ali Hadi Hasan. “Robot path planning based on PSO and D$*$ algorithmsin dynamic environment”. In: *2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT).* IEEE, 2017. DOI: 10.1109/crcsit.2017.7965550.

[She+17] He Shen, Ni Li, Holly Griffiths, and Salvador Rojas. “Tracking control of a small unmanned air vehicle with airflow awareness”. In: *American Control Conference (ACC), 2017.* IEEE. 2017, pp. 4153–4158.

[SPC17] Jose Luis Sanchez-Lopez, Jesus Pestana, and Pascual Campoy. “A robust real-time path planner for the collision-free navigation of multirotor aerial robots in dynamic environments”. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS).* IEEE, 2017. DOI: 10.1109/icuas.2017.7991354.

[SW08] Erol Sahin and Alan Winfield. “Special issue on swarm robotics”. In: *Swarm Intelligence.* Vol. 2. 2-4. Springer Nature, 2008, pp. 69–72. DOI: 10.1007/s11721-008-0020-6.

[SWF10] Timothy Stirling, Steffen Wischmann, and Dario Floreano. “Energy-efficient indoor search by swarms of simulated flying robots without global information”. In: *Swarm Intelligence.* Vol. 4. 2. Springer Nature, 2010, pp. 117–143. DOI: 10.1007/s11721-010-0039-3.

[TZ13] Ying Tan and Zhong-yang Zheng. “Research Advance in Swarm Robotics”. In: *Defence Technology.* Vol. 9. 1. Elsevier BV, 2013, pp. 18–39. DOI: 10.1016/j.dt.2013.03.001.

[Zen+12] Zheng Zeng, Andrew Lammas, Karl Sammut, and Fangpo He. “Optimal path planning based on annular space decomposition for AUVs operating in a variable environment”. In: *2012 IEEE/OES Autonomous Underwater Vehicles (AUV).* IEEE, 2012. DOI: 10.1109/auv.2012.6380759.

[Zhe+05] Changwen Zheng, Lei Li, Fanjiang Xu, Fuchun Sun, and Mingyue Ding. “Evolutionary route planner for unmanned air vehicles”. In: *IEEE Transactions on Robotics.* Vol. 21. 4. Institute of Electrical and Electronics Engineers (IEEE), 2005, pp. 609–620. DOI: 10.1109/tro.2005.844684.

[ZK13] Xin Zhou and David Kinny. “Energy-Based Particle Swarm Optimization: Collective Energy Homeostasis in Social Autonomous Robots”. In: *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT).* IEEE, 2013. DOI: 10.1109/wi-iat.2013.87.

Statutory Declaration

I assure that this thesis is a result of my personal work and that no other than the indicated aids have been used for its completion. Furthermore, I assure that all quotations and statements that have been inferred literally or in a general manner from published or unpublished writings are marked as such. Beyond this I assure that the work has not been used, neither completely nor in parts, to pass any previous examination.

Magdeburg, ...