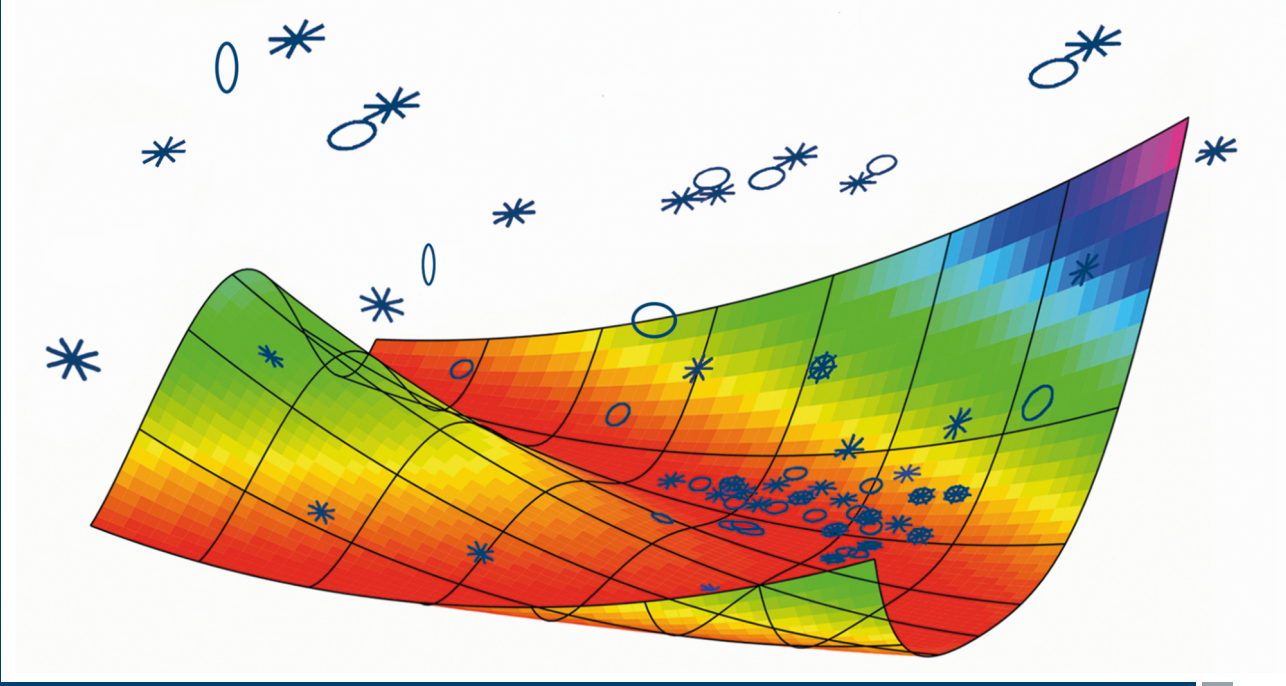


FAU Forschungen, Reihe B, Medizin, Naturwissenschaft, Technik 3

**Berthold Immanuel Schmitt**

# Convergence Analysis for Particle Swarm Optimization

FAU UNIVERSITY PRESS

Berthold Immanuel Schmitt

# Convergence Analysis for Particle Swarm Optimization

FAU Forschungen, Reihe B

Medizin, Naturwissenschaft, Technik

Band 3

Herausgeber der Reihe:
Wissenschaftlicher Beirat der FAU University Press

Berthold Immanuel Schmitt

# Convergence Analysis for Particle Swarm Optimization

Konvergenzanalyse für die Partikelschwarmoptimierung

Erlangen
FAU University Press
2015

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.



Der vollständige Inhalt des Buchs ist als PDF über den OPUS Server der Friedrich-Alexander-Universität Erlangen-Nürnberg abrufbar:
http://opus.uni-erlangen.de/opus/

Verlag und Auslieferung:

FAU University Press, Universitätsstraße 4, 91054 Erlangen

Druck: docupoint GmbH

ISBN: 978-3-944057-30-9
ISSN: 2198-8102

# Convergence Analysis for Particle Swarm Optimization

Konvergenzanalyse für die Partikelschwarmoptimierung

Der Technischen Fakultät
der Friedrich-Alexander-Universität
Erlangen-Nürnberg
zur
Erlangung des Doktorgrades Dr.-Ing.

vorgelegt von

Berthold Immanuel Schmitt

aus Hildesheim

Als Dissertation genehmigt
von der Technischen Fakultät
der Friedrich-Alexander-Universität Erlangen-Nürnberg
Tag der mündlichen Prüfung: 04.02.2015

Vorsitzende des Promotionsorgans: Prof. Dr.-Ing. habil. Marion Merklein

Gutachter: Professor Dr. rer. nat. Rolf Wanka
Professor Dr. rer. nat. Benjamin Doerr

# Abstract

Particle swarm optimization (PSO) is a very popular, randomized, nature-inspired meta-heuristic for solving continuous black box optimization problems. The main idea is to mimic the behavior of natural swarms like, e. g., bird flocks and fish swarms, that find pleasant regions by sharing information and cooperating rather than competing against each other. For optimization purpose, a number of artificial particles move through the $\mathbb{R}^D$ and the movement of a particle is influenced not only by its own experience, but also by the experiences of its swarm members.

Although this method is widely used in real-world applications, there is unfortunately not much understanding of PSO based on formal analyses, explaining more than only partial aspects of the algorithm. One aspect that is target of many researchers' work is the phenomenon of a converging swarm, i. e., the particles converge towards a single point in the search space. In particular, necessary and sufficient conditions to the swarm parameters, i. e., certain parameters that control the behavior of the swarm, for guaranteeing convergence could be derived. However, prior to this work, no theoretical result about the quality of this limit for the unmodified PSO algorithm and a situation more general than considering just one particular objective function have been shown.

In this thesis, we study the convergence process in detail. In order to measure, how far the swarm at a certain time is already converged, we define and analyze the potential of a particle swarm. The potential is constructed such that it converges to $0$ if and only if the swarm converges, but we will prove that in the 1-dimensional case, when the swarm is far away from a local optimum, the potential increases. This observation turns out to be sufficient to prove the first main result, namely that in a 1-dimensional situation, the swarm with probability 1 converges towards a local optimum for a comparatively wide range of objective functions. Additionally, we apply drift theory in order to prove that for unimodal objective functions, the result of the PSO algorithm agrees with the actual optimum in $k$ digits after time $\mathcal{O}(k)$.

In the general D-dimensional case, it turns out that the swarm might not converge towards a local optimum. Instead, it gets stuck in a situation where some dimensions have a potential orders of magnitude smaller than others. Such dimensions with a too small potential lose their influence on the behavior of the algorithm, and therefore, the respective entries are not optimized. In the end, the swarm stagnates, i. e., it converges towards a point in the search space, that is not even a local optimum. In order to solve this issue, we propose a slightly modified PSO that again guarantees convergence towards a local optimum.

# Zusammenfassung

Partikelschwarmoptimierung (PSO) ist eine sehr verbreitete, randomisierte, von der Natur inspirierte Meta-Heuristik zum Lösen von Black-Box-Optimierungsproblemen über einem kontinuierlichen Suchraum. Die Grundidee besteht in der Nachahmung des Verhaltens von in der Natur auftretenden Schwärmen, die vielversprechende Regionen finden, indem sie Informationen austauschen und miteinander kooperieren, anstatt gegeneinander zu konkurrieren. Im daraus gewonnenen Optimierungsverfahren bewegen sich künstliche Partikel durch den $\mathbb{R}^D$, wobei die Bewegung eines Partikels nicht nur von dessen eigener Erfahrung, sondern genauso von den Erfahrungen der übrigen Schwarmmitglieder beeinflusst wird.

Obwohl diese Methode in zahlreichen realen Anwendungen verwendet wird, haben theoretische Betrachtungen bisher nur einige wenige Teilaspekte des Algorithmus erklärt. Ein solcher Aspekt, mit dem sich viele Wissenschaftler auseinandersetzen, ist das Phänomen der Konvergenz des Partikelschwarms. Das bedeutet, dass die Partikel gegen einen Punkt im Suchraum konvergieren. Insbesondere konnten notwendige und hinreichende Bedingungen an die Schwarmparameter, bestimmte Parameter die das Verhalten des Schwarms steuern, ermittelt werden, unter denen Konvergenz gewährleistet ist. Allerdings ist bis jetzt kein theoretisches Resultat über die Qualität dieses Grenzwertes bekannt, das für den unmodifizierten PSO-Algorithmus in einer allgemeineren Situation als beispielsweise nur für genau eine Zielfunktion bewiesen werden konnte.

Diese Arbeit befasst sich detailliert mit dem Prozess der Konvergenz. Um zu messen, wie stark der Schwarm bereits konvergiert ist, wird das Potential eines Partikelschwarms eingeführt und analysiert. Das Potential ist so konstruiert, dass es genau dann gegen $0$ konvergiert, wenn der Schwarm konvergiert. Im 1-dimensionalen Fall ergeben die Betrachtungen, dass sich das Potential erhöht, solange der Schwarm weit vom nächsten lokalen Optimum entfernt ist. Diese Beobachtung führt zum Beweis des ersten Hauptresultats, nämlich dass im 1-dimensionalen Fall der Schwarm fast sicher gegen ein lokales Optimum konvergiert. Dieses Resultat ist für eine vergleichbar

große Klasse von Zielfunktionen gültig. Zusätzlich kann mittels Drifttheorie gezeigt werden, dass das Ergebnis des PSO-Algorithmus nach einer Zeit von $\mathcal{O}(k)$ mit dem tatsächlichen Optimum in $k$ Bits übereinstimmt.

Im allgemeinen $D$-dimensionalen Fall stellt sich heraus, dass der Schwarm nicht zwangsläufig gegen ein lokales Optimum konvergiert. Stattdessen gerät er in eine Situation, in der manche Dimensionen ein um Größenordnungen geringeres Potential haben als andere. Diese Dimensionen mit zu geringem Potential verlieren ihren Einfluss auf das Verhalten des Algorithmus, und daher werden die entsprechenden Einträge nicht optimiert. Die Konsequenz ist Stagnation des Schwarms, das heißt, der Schwarm konvergiert gegen einen Punkt im Suchraum, der nicht mal ein lokales Optimum ist. Um dieses Problem zu lösen wird eine leicht modifizierte Version der PSO vorgeschlagen, die wiederum eine Garantie für Konvergenz gegen ein lokales Optimum zulässt.

**"The PSO algorithm can be compared to a group of nerds randomly spread in the mountains. They are supposed to get close to the highest point in a limited area. As nerds usually are not used to daylight, their only way to navigate is via the GPS in their mobile phones. Furthermore, they are allowed to use facebook to share their position with their friends. Now they walk randomly around, always a bit towards their personal previous best position and a bit towards the best position on facebook."**

Christoph Strößner, participant of Sarntal Ferienakademie, 2014

# Acknowledgments

First of all, I would like to express my sincere gratitude to Prof. Dr. Rolf Wanka for his supervision, support and editorial and scientific advice throughout the whole time of my graduate studies.

Additionally, I am grateful to the staff of the chair of Hardware-Software-Co-Design for the great and interesting time I had there.

Special thanks for their great work and the interesting discussions go to the students whose bachelor and master theses I supervised, namely Vanessa Lange, Bernd Bassimir, Franz Köferl, Stefan Ploner, Lydia Schwab, Alexander Raß and Gabriel Herl.

Finally, I would like to thank my colleague and office roommate Dr.-Ing. Moritz Mühlenthaler for the great years, for many exchanged ideas and for carefully reading wide parts of this thesis.

# Contents

# 1. Introduction and Contribution

Particle swarm optimization (PSO), originally invented by Kennedy and Eberhart [KE95, EK95] in 1995, is a very popular nature-inspired meta-heuristic for solving continuous optimization problems. It is designed to reflect the social interaction of individuals living together in groups and supporting and cooperating with each other, rather than competing against each other. Fields of successful application are Biomedical Image Processing [WSZ+04], Geosciences [OD10], Mechanical Engineering [GWHK09], and Materials Science [RPPN09], to name just a few, where the continuous objective function on a multi-dimensional domain is not given in a closed form, but by a "black box." That means that the only operation that the objective function allows is the evaluation of search points while, e. g., gradient information is not available.

The popularity of the PSO framework in various scientific communities is due to the fact that it on the one hand can be realized and, if necessary, adapted to further needs easily, but on the other hand shows in experiments good performance results with respect to the quality of the obtained solution and the time needed to obtain it. By adapting its parameters, users may in real-world applications easily and successfully control the swarm's behavior with respect to "exploration" ("searching where no one has searched before") and "exploitation" ("searching around a good position"). A thorough discussion of PSO can be found in [PSL11].

To be precise, let an objective function $f : \mathbb{R}^D \to \mathbb{R}$ on a D-dimensional domain be given that (without loss of generality) has to be minimized. A population of *particles*, each consisting of a position (the candidate for a solution), a velocity and a local attractor, moves through the search space $\mathbb{R}^D$. The local attractor of a particle is the best position with respect to $f$ this particle has encountered so far. The population in motion is the *swarm*. In contrast to evolutionary algorithms, the individuals of the swarm cooperate by sharing information about the search space via the global attractor, which is the best position any particle has found so far. The particles move in time-discrete iterations. The movement of a particle is governed by so-

called movement equations that depend on both the particle's velocity and its two attractors and on some additional fixed parameters, controlling the influence of the attractors and the velocity on the next step (for details, see Chapter 2.2).

Although this method is widely used in real-world applications, there is unfortunately not much understanding of the algorithm based on formal analyses, explaining more than only partial aspects like analyzing the trajectories of restricted and deterministic PSO variants. One aspect that is target of many researchers' work is the phenomenon of a converging swarm, i. e., the particles converge towards one point in the search space. This exploitative behavior is desired because it allows the swarm to refine a solution and maybe come arbitrarily close to the optimum. Many experiments support the claim that using appropriate parameters allows the swarm to converge. In particular, necessary and sufficient conditions to the swarm parameters were derived in the literature, which guarantee convergence of the swarm under the assumption that the attractors are constant. However, until now, no theoretical result about the quality of this limit for the unmodified PSO algorithm, valid in a situation more general than considering just one particular objective function, have been shown.

## 1.1 Contributions

The main goal of this thesis is to provide the first *general* mathematical analysis of the quality of the global attractor when it is considered as a solution for objective functions from a very general class $\mathbb{F}$ of functions and therefore of the quality of the algorithm's return value. The set $\mathbb{F}$ consists of all functions that have a continuous first derivative and, roughly spoken, have only a bounded area that is of interest for the particles. Note that the class $\mathbb{F}$ of admissible objective functions, for which our convergence results hold, is much more general than, e. g., the subset of the class of the unimodal functions that is considered in [Jäg07] in the context of restricted $(1 + 1)$ evolutionary algorithms.

First, we propose a mathematically sound model of PSO, which describes the algorithm as a *stochastic process* over the real numbers. Before presenting our convergence analysis, several previous results, in particular different

negative results stating that the PSO algorithm is not an "optimizer", are discussed in the light of this new model.

### Main Tools for the Analysis

As an important tool for the analysis, we introduce the new approach of defining the *potential* of the particle swarm that changes after every step. The potential covers two properties of the swarm: It tends to 0 if the particles converge, but we can show that it increases if the whole swarm stays close to a search point that is no local minimum. In the latter case, we prove that the swarm charges potential and resumes its movement.

Another important tool for our analysis is drift theory, which has already been used for proving runtime bounds in the area of evolutionary algorithms. Drift theorems allow us to transform bounds of the drift, i. e., the expected tendency of a stochastic process in a certain direction, into bounds on the expected time until the process hits a predefined value. In order to use drift theory for analyzing PSO, we formulate and prove a new drift theorem, specifically designed for stochastic processes on the continuous domain $\mathbb{R}$. As far as we know, this thesis contains the first work that analyzes PSO with the help of drift theory.

### Contributions to the Analysis of 1-Dimensional PSO

As our first main result, we can prove an emergent property of PSO for $\mathbb{F}$, namely that in the 1-dimensional case the swarm almost surely (in the well-defined probabilistic sense) finds a local optimum. More precisely: If no two local optima have the same objective function value, then the swarm converges towards a local optimum almost surely.

Since the possible area for the global attractor is bounded, the Bolzano-Weierstrass theorem implies that either the global attractor converges towards a single point in the search space, or there are at least two accumulation points of the global attractor, i. e., points to which the global attractor comes arbitrarily close infinitely often. If the global attractor converges, then the results of previous theoretical work guarantee convergence of the whole

swarm. But, as our potential analysis shows, the swarm cannot converge towards a point that is no local optimum. So, if the global attractor converges, then its limit is a local optimum. If there are two or more accumulation points, then the swarm maintains a certain amount of potentential, depending on the distance between the accumulation points, since this distance is overcome infinitely often. As it will turn out in our analysis, if some of the accumulation points are no local optimum, this potential is sufficient to find a region better than the accumulation point. However, since the global attractor does not accept worsenings, once it has found a better position than a point $z \in \mathbb{R}$, it cannot come arbitrary close to $z$ anymore, so $z$ cannot be an accumulation point — a contradiction. Altogether, only the cases when every accumulation point is a local optimum remain.

In case of unimodal functions, this result implies convergence towards even the global optimum. Therefore, the next step is to bound the runtime in case of 1-dimensional, unimodal functions. Since hitting the optimum exactly within a finite number of steps is not possible, we instead ask for the time until the optimum and the global attractor agree in $k$ digits. The result of our analysis is a runtime bound of $\mathcal{O}(k)$.

To achieve this result, we further study the process of convergence and propose a classification of the particle swarm's possible configurations into "good" configurations, i. e., configurations that allow the swarm to improve the candidate solution directly, and "bad" configurations, from which significant improvements are not directly possible. This may be because either the potential of the swarm is too low, such that the steps width of the particles is insignificant compared to the distance to the optimum, or the potential is too high, such that the probability for an improvement is close to $0$. Indeed, such bad configurations occur with positive probability and therefore frequently if the algorithm is run long enough, but our analysis formally shows another emergent property of the swarm, namely the necessary self-healing property, that enables the swarm to recover from such a bad configuration within finite and actually quite reasonable time.

We construct an appropriate distance function that measures how far the swarm is away from an "optimal state" where every particle is located at the optimum. Note that this state cannot be reached within a finite number of iterations, but the swarm might converge towards it. Our distance measure is composed of the so-called primary measure, i. e., the quality of the attractors with respect to the objective function, and additional secondary measures that measure the "badness" of the respective situation. It is proved that indeed the swarm frequently makes progress, either directly by improv-

ing the attractors or indirectly by reducing the "penalties" of the secondary measures.

Applying our drift theorem to this distance measure leads to our second main result, namely the proof that for 1-dimensional, unimodal objective functions, the convergence speed is linear, i. e., the expected number of iterations necessary until the global attractor and the actual optimum agree in $k$ digits is $\mathcal{O}(k)$, where the constant involved in the $\mathcal{O}$ does not depend on the actual objective function.

### Contributions to the Analysis of $D$-Dimensional PSO

For the general $D$-dimensional case, our studies of the respective processes of PSO encountering bad situations are mostly experimental. This approach reveals that while classical PSO is able to heal itself from most of the bad configurations, there is one type that the swarm cannot recover from and that indeed causes stagnation, i. e., convergence towards a point in the search space that is not a local optimum. Therefore, the classification from the 1-dimensional case is extended by the set of "fatal" configurations, which can cause non-optimal stagnation.

More precisely: During the search process, it might occur that the potentials of the different dimensions are imbalanced, i. e., some dimensions have a significantly smaller potential than others. The entries of the particles' positions and velocities in such dimensions with a too small potential lose their influence on the behavior of the algorithm, therefore the swarm converges towards a point that is not even a local optimum, while the imbalance of potentials between the different dimensions is maintained and actually worsened. That means that while in such a fatal situation, the swarm cannot make significant improvements of its positions in the search space and there is a positive probability that the swarm will never heal itself from this situation. Therefore, we slightly modify the classical PSO in order to respond on this particular weakness. Our modified PSO behaves like the classical PSO as long as the potential of the swarm is larger than a user defined parameter. As soon as the swarm potential falls below this specified bound, the updated velocities are chosen uniformly from some small area. As our third main result, we prove that this modified PSO almost surely finds a local optimum for functions in $\mathbb{F}$.

We present experiments indicating that indeed the modification does not completely alter the behavior of the swarm. Instead, after healing itself from encountering the fatal event, the swarm switches back to the behavior of the classical PSO.

Note that although we present the analysis only for one particular PSO version, in the following called the classical PSO, the general technique of defining a potential, analyzing occuring configurations and measuring their "badness" can be generalized to presumably all variants of PSO developed so far.

## 1.2 Overview

The structure of this thesis is as follows: In Chapter 2, we introduce the PSO algorithm with its applications, variants, extensions and generalizations and provide an overview over related work. First, we motivate the use of PSO by presenting a collection of successful applications of PSO for problems in a black box setting, where a closed form of the objective function is either completely unavailable or too complicated to be useful. In Section 2.2, we introduce the exact version of the classical PSO algorithm that we will analyze in Chapter 4. An overview over selected variants of the classical PSO algorithm is provided in Section 2.3. In Section 2.4, we present an overview over common variants of PSO for multi-objective optimization problems, where the goal is to find points in the search space that are "good" with respect to several, possibly conflicting objective functions. In Section 2.5, we show several adapted PSO variants that are designed for discrete optimization problems. A brief overview over the theoretical results regarding PSO is presented in Section 2.6. Finally, we conclude Chapter 2 with a brief introduction into other important nature-inspired meta-heuristics, namely evolutionary algorithms and ant algorithms.

Chapter 3 provides the formulation of the mathematical model of PSO, which we use for the analysis. Therefore, in Section 3.1 we first recall the relevant definitions from probability theory, namely random variables, stochastic processes, conditional expectations and related concepts. In Section 3.2, we outline the famous No Free Lunch Theorem, a strong negative result in the field of combinatorial optimization, which basically says that in a perfect black box situation when nothing is known about the objective function, any

two search heuristics have the same performance. In particular, this implies that no algorithm is better than blind search, i. e., the best algorithm is just sampling random points of the search space and returning the best. However, as we will explain in Section 3.2, the same result is not true in a continuous situation. In Section 3.3, drift theory is introduced as an important tool for runtime analysis. We recall classical results in drift theory and formulate and prove a modified drift theorem, suitable for the analysis of PSO. In Section 3.4, we finally state the proposed model of the PSO algorithm in terms of stochastic processes. Additionally, we introduce the potential of a particle swarm as a measure for its ability to reach far-off areas of the search space. Finally, in Section 3.5, we point out previous results, which are closely related to the work of this thesis, in detail and discuss some negative results, that on the first sight look as if they were in contradiction with our results.

Chapter 4 contains the main theoretical results about a particle swarm optimizing an objective function from the comparatively large set of functions $\mathbb{F}$ over a 1-dimensional search space. Our first main result is presented in Section 4.1, namely the formal proof that the classical, unmodified PSO algorithm finds a local optimum in the sense that every accumulation point of the global attractor is a local optimum. In Section 4.2, we present our second main result, i. e., the rigorous runtime analysis for the case of unimodal objective functions.

In Chapter 5, we approach the multidimensional case. First, in Section 5.1 we collect the different bad configurations and empirically examine the behavior of the swarm when exposed to such difficulties. In Section 5.2, we propose a slightly altered version of the PSO algorithm, where we made a modification in order to void the weakness of PSO when it is confronted with imbalanced potentials. We experimentally investigate the modified PSO for its capability to actually overcome the fatal event and for the overall impact of the modification.

# 2. Particle Swarm Optimization: State of the Art

Particle swarm optimization (PSO) is a popular meta-heuristic, inspired by the social interaction of individuals living together in groups and supporting and cooperating with each other. Since it was invented in 1995 by Kennedy and Eberhart ([KE95, EK95]), the PSO method has drawn the attention of an increasing number of researchers because of its simplicity and efficiency ([PKB07]). The goal of the PSO algorithm is to find the optimum of an objective function $f : S \subset \mathbb{R}^D \to \mathbb{R}$. For the rest of this thesis, we assume that $f$ is to be minimized. Since maximizing $f$ is equivalent to minimizing $-f$, this is without loss of generality.

Although PSO works for literally any function $f$, it is typically applied when there is no closed form of $f$. In such a situation, information about $f$ can only be gained by evaluating $f$ pointwise. In particular, the information about the gradient of $f$ is unavailable. Figure 2.1 gives a graphical overview over the described situation, which is referred to as a black box optimization problem.

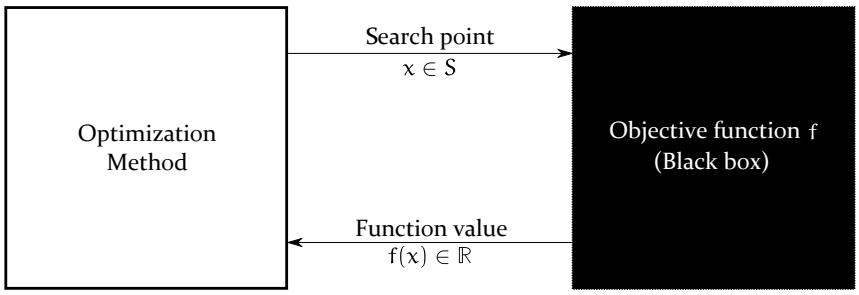



**Figure 2.1:** Black box optimization.

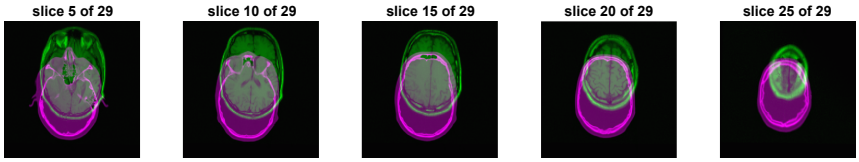
## 2.1 Applications of Particle Swarm Optimization

Black box optimization problems, where the objective function is not explicitly available and function evaluations are expensive, occur in many different areas. Quite often experiments show that PSO or at least some PSO variant is capable to solve them, i. e., to find a solution with a quality sufficiently good for the application, although it is usually not the global optimum. The time to implement PSO for a given application and the optimization time itself are typically sufficiently short, such that PSO is in a lot of cases more attractive than a sophisticated, problem specific and exact method. In the following, a selection of real-world problems with black box flavor, to which PSO was applied successfully, is presented.

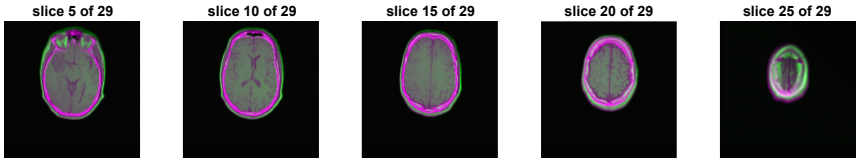
In electrical power systems consisting of several producers (e. g., generators), intermediate nodes and consumers, the control center needs to react on load changes of the consumers. The power transmission loss depends on several parameters like certain automatic voltage regulator operating values. The Volt/Var Control (VVC) problem asks for a configuration that yields the minimal loss subject to certain constraints, e. g., voltage security requirements of the target system and permissible range of the voltage magnitude at each intermediate node. In [YKF+00, MF02], PSO has empirically proved its capability to solve the underlying optimization problem.

Size and shape optimization looks for the optimal geometry of a truss structure with respect to stress, strain and displacement constraints. The authors of [FG02] compare the performance of PSO against a variety of other algorithms by solving a number of instances of this problem.

Three different applications from the field of mechanical engineering optimization are presented in [HES03a]. First, PSO is used to solve the problem of minimizing the total cost of building a cylindrical pressure vessel, depending on its exact shape and form. As a second application, the authors use PSO for minimizing the cost of welding a rigid member onto a beam. The total cost, consisting of the cost of the material and the labor cost, depends on the exact geometry of the member and the beam. This geometry is subject to certain constraints, regarding, e. g., the overall size or the bending stress. Finally, the weight of a tension/compression spring is minimized, which depends on, e. g., the coil diameter, the wire diameter and the number of coils. Again, the optimization problem is due to certain constraints regarding size and overall shape of the spring.




**(a)** Unregistered image showing five slices of both, a CT image (pink) and a MRI image (green) of the human brain ([RIR14])




**(b)** Five slices of both, a CT image (pink) and a MRI image (green) of the human brain after a PSO-based registration done in [Sch14].

**Figure 2.2:** Example of a 3-dimensional CT image (pink) and a 3-dimensional MRI image (green) of the human brain, obtained from [RIR14] before registration and after registration done in [Sch14] by using PSO.

Different properties of an antenna like its weight and its return loss depend on physical and electromagnetic characteristics, e. g., its length, its overall shape and the number of corrugations per wavelength. In [RRS04], the authors present a PSO-based approach to find a design that matches these design goals by finding the optimal design of an antenna.

In biomedical research, PSO is used for image registration ([WSZ+04]). It is common to image the same body part with different methods, e. g., Computer Tomography (CT) provides images of bones while Magnetic Resonance Imaging (MRI) is suitable for scanning soft tissues. As a result, one gets (two-dimensional or three-dimensional) images taken under different modalities that need to be aligned. The search space is the set of all Euclidean transformations, i. e., transformations that preserve length, and the objective function is some similarity measure between the images. Figure 2.2 shows an example of a 3-dimensional CT image (pink) and a 3-dimensional MRI image (green) of the human brain, obtained from [RIR14] before registration and after registration done in [Sch14] by using PSO.

In mineralogy, a variant of PSO is used to find certain mineral-melt equilibria, allowing for a better understanding of the behavior of magmas within the earth's crust ([HM06]). The chemical reactions of silicate melts depend

on a large number of parameters (1000 and more) because the number of different metal atoms can be very high. To calculate the equilibria, one minimizes the change in free energy. In ([HM06]), the authors successfully applied a variant of PSO to this problem.

The life time of metal machine tools can be increased by composite coatings. In [RPPN09], the authors use PSO in order to optimize certain parameters of the nickel-diamond composite coating process, e. g., the temperature and the concentration of diamond particles, such that the resulting hardness is maximized.

In Universal Mobile Telecommunications System (UMTS), a multiple access scheme called Code Division Multiple Access (CDMA) is used. Instead of simply sharing the bandwidth or the time, CDMA distincts users by using codes. Therefore, interference cancellation techniques are necessary. The task to find a good interference cancellation technique can be rewritten as an optimization problem, which is in [ZWLK09] solved using a PSO variant.

For the development of gas and oil fields, finding optimal type and location for new wells is very important. The underlying objective function is very complicated and can only be evaluated pointwise by computationally expensive simulations. In [OD10], the authors present a PSO-based approach to solve this optimization problem.

There are many more fields in which PSO has been used successfully to solve optimization problems originating from real world applications. This collection gives just an impression on how different the application fields and the underlying optimization problems are, for which PSO was the algorithm of choice to produce good, though not optimal, solutions with reasonable effort and time.

## 2.2 The Classical Particle Swarm Optimization Algorithm

The first version of a particle swarm optimization algorithm was published by Kennedy and Eberhart ([KE95, EK95]). The algorithm was built to simulate a population of individuals, e. g., bird flocks or fish schools, searching for a region that is optimal with respect to some hidden objective function, e. g., the amount and the quality of food. In contrast to other popular nature-inspired meta-heuristics like evolutionary algorithms (EAs) (a brief overview over EAs can be found in Section 2.7.1), the particles of a particle swarm work

together and share information about good places rather than competing against each other.

At each time $t$, each particle $n$ has a *current position* $X_t^n$ and a *velocity* $V_t^n$. Additionally, every particle remembers the best position it has visited so far. This position is called the *local attractor* or the *private guide* and is denoted by $L_t^n$. The best of all local attractors among the swarm is called the *global attractor* or the *local guide*. This special position is denoted by $G_t$ and it is visible for every particle. So, by updating the global attractor, a particle shares its information with the remaining swarm.

For some optimization problem with objective function $f : S \subset \mathbb{R}^D \to \mathbb{R}$, the positions are identified with search points $x \in S$ and the velocities are identified with vectors $v \in \mathbb{R}^D$. The actual movement of the particles is governed by the following *movement equations*:

$$V_{t+1}^{n,d} = V_t^{n,d} + c_1 \cdot r_t^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) + c_2 \cdot s_t^{n,d} \cdot (G_t^d - X_t^{n,d}), \quad (2.1)$$

$$X_{t+1}^{n,d} = X_t^{n,d} + V_{t+1}^{n,d}, \quad (2.2)$$

where $t$ denotes the iteration, $n$ the number of the particle that is moved and $d$ the dimension. The constants $c_1$ and $c_2$ control the influence of the personal memory of a particle and the common knowledge of the swarm and are called *acceleration coefficients*. Some randomness is added via $r_t^{n,d}$ and $s_t^{n,d}$, which are drawn uniformly at random in $[0, 1]$ and all independent of each other. The movement equations are iterated, until some fixed termination criterion is reached. Figure 2.3 gives an overview over the particles' movement.

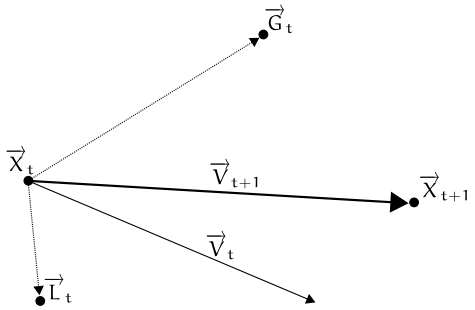



**Figure 2.3:** Particles' movement. The new velocity depends on the old velocity, the local attractor and the global attractor.

In order to prevent the phenomenon of so-called *explosion*, meaning that the absolute values of the particles' velocities grow unboundedly over time,

early versions of PSO used *velocity clamping* ([PKB07]), i. e., whenever a component of the velocity exceeds a certain interval $[-v_{max}, v_{max}]$, it is set to the according interval bound. Then, the movement equations become

$$V_{t+1}^{n,d} = \max\{-v_{max}, \min\{v_{max}, V_t^{n,d} + c_1 \cdot r_t^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) + c_2 \cdot s_t^{n,d} \cdot (G_t^d - X_t^{n,d})\}\}, \tag{2.3}$$

$$X_{t+1}^{n,d} = X_t^{n,d} + V_{t+1}^{n,d}. \tag{2.4}$$

Instead of using velocity clamping to avoid explosion, in [SE98] Shi and Eberhart modified the movement equation by multiplying the previous velocity with some factor $\chi \in (0, 1)$, called the *inertia weight*, leading to the following form of the movement equations:

$$V_{t+1}^{n,d} = \chi \cdot V_t^{n,d} + c_1 \cdot r_t^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) + c_2 \cdot s_t^{n,d} \cdot (G_t^d - X_t^{n,d}), \tag{2.5}$$

$$X_{t+1}^{n,d} = X_t^{n,d} + V_{t+1}^{n,d}. \tag{2.6}$$

The authors could experimentally show that the performance of PSO significantly depends on the inertia weight. Typical choices for the parameters are

- $\chi = 0.72984$, $c_1 = c_2 = 1.496172$ ([CK02, BK07]),
- $\chi = 0.72984$, $c_1 = 2.04355$, $c_2 = 0.94879$ ([CD01]) or
- $\chi = 0.6$, $c_1 = c_2 = 1.7$ ([Tre03]).

The standard swarm parameters for the experiments of this thesis are $\chi = 0.72984$, $c_1 = c_2 = 1.496172$. We use this parameters for every experiment unless the ones in which different parameters are compared and the choices are explicitly stated.

By adjusting the parameters, it is possible to influence the trade-off between *exploration*, the capability to search in areas that have not been visited before, and *exploitation*, the capability to refine already good search points.

The initialization of the positions is usually done uniformly at random over some bounded search space. An alternative is presented in [RV04], where the authors propose a method based on centroidal Voronoi tessellations that should ensure that the particles are distributed over the search space more evenly than just by random distribution. Typical initialization strategies for the velocity are

- **Random**: Like the positions, the velocities are initialized randomly,

- **Zero**: All velocities are initially $0$,
- **Half-Diff**: Additionally to the initial position $X_0^n$, a second point $\hat{X}_0^n$ in the search space is sampled. The initial velocity is then set to $(\hat{X}_0^n - X_0^n)/2$.

**Algorithm 1:** classical PSO

**input** : Objective function $f : S \to \mathbb{R}$ to be minimized
**output**: $G \in \mathbb{R}^D$

```
// Initialization
```
**for** $n = 1 \to N$ **do**
    Initialize position $X^n \in \mathbb{R}^D$ randomly;
    Initialize velocity $V^n \in \mathbb{R}^D$;
    Initialize local attractor $L^n := X^n$;
Initialize $G := \operatorname{argmin}_{\{L^1,\dots,L^n\}} f$;
```
// Movement
```
**repeat**
    **for** $n = 1 \to N$ **do**
        **for** $d = 1 \to D$ **do**
            $V^{n,d} := \chi \cdot V^{n,d} + c_1 \cdot \text{rand}() \cdot (L^{n,d} - X^{n,d}) + c_2 \cdot \text{rand}() \cdot (G^d - X^{n,d})$;
            $X^{n,d} := X^{n,d} + V^{n,d}$;
        **if** $f(X^n) \leq f(L^n)$ **then** $L^n := X^n$;
        **if** $f(X^n) \leq f(G)$ **then** $G := X^n$;
**until** *Termination criterion holds*;
**return** $G$;

Algorithm 1 gives a pseudo code representation of the PSO algorithm. Basically, this algorithm implements the common movement equations including the inertia weight with two specifications: If a particle visits a point with the same objective value as its local attractor or the global attractor, then the respective attractor is updated to the new point. And the global attractor is updated after every step of a single particle, not only after every iteration of the whole swarm.

Another common variant of PSO, sometimes known as *parallel PSO*, only updates the global attractor after every iteration of the whole swarm. However, due to the choice made here, the information shared between the par-

ticles is as recent as possible. We will refer to the exact version of PSO stated in Algorithm 1 as *classical PSO* for the rest of this thesis.

As long as in the experiments performed throughout this thesis nothing else is stated, the velocity initialization strategy is **Random**, i. e., if the positions in dimension $d$ are initialized over the interval $[a_d, b_d]$, then we initialize the velocities' entries in dimension $d$ uniformly at random in the interval $[-(b_d - a_d)/2, (b_d - a_d)/2]$.

## 2.3 Variants of Particle Swarm Optimization

Since its introduction in 1995, the PSO algorithm was frequently altered and improved ([PKB07]). Researchers studied more refined versions of the PSO algorithm voiding certain weaknesses and combined it with other methods to form hybrid optimization methods. This section provides an overview over some of the most important variants of PSO that have been developed.

### 2.3.1 Neighborhood Topologies

In the classical PSO, the particles share information via the global attractor, which is the best solution any particle has found so far and which is known to the whole swarm. That means that every particle interacts with every other member of the swarm.

In order to better reflect social learning processes, the global attractor is replaced by the *local guide*. The local guide of some particle $n$ is the best (with respect to the objective function $f$) local attractor among all neighbors of particle $n$. If two particles are neighbors of each other is defined via the so-called *neighborhood topology*, typically represented as a (sometimes directed) graph, whose nodes are the particles and whose edges connect neighboring particles. An edge pointing from particle $n_1$ to particle $n_2$ means that $n_1$ considers the private guide of $n_2$ as a candidate for its own local guide. The set of all the neighbors of a particle $n$ is denoted as $\mathcal{N}(n)$. The velocity update equation (Equation (2.5)) changes to

$$V_{t+1}^{n,d} = \chi \cdot V_t^{n,d} + c_1 \cdot r_t^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) + c_2 \cdot s_t^{n,d} \cdot (P_t^{n,d} - X_t^{n,d}),$$

where $P_t^n$ is the best position, any of the neighbors of particle $n$ has visited so far (with some additional convention in case of a tie). In terms:

$$P_t^n := \operatorname*{argmin}_{x \in \{\hat{L}_t^{n',n} \mid n' \in \mathcal{N}(n)\}} f(x).$$

where $\hat{L}_t^{n',n}$ is the local attractor of particle $n'$ at the time when particle $n$ makes its move, i. e.,

$$\hat{L}_t^{n',n} = \begin{cases} L_t^{n'}, & \text{if } n' \geq n \\ L_{t+1}^{n'}, & \text{otherwise.} \end{cases} \tag{2.7}$$

Early attempts to form a neighborhood topology depending on the Euclidean distance of the particles in the search space have shown bad results ([PKB07]). Therefore, the neighborhood topology is typically chosen independent of the positions of the particles in the search space, but with respect to the particles' indices.

### Static neighborhood topologies

Common examples from the literature for neighborhood topologies which are static, i. e., are not changed during the optimization process, are:

- The fully connected graph, in which any two particles are neighbors. With this topology, the PSO algorithm behaves like the classical PSO from Algorithm 1. This topology is sometimes also called *gbest topology* (global best, [MKN03]) or *star topology* ([Ken99]). This topology allows the fastest distribution of information.

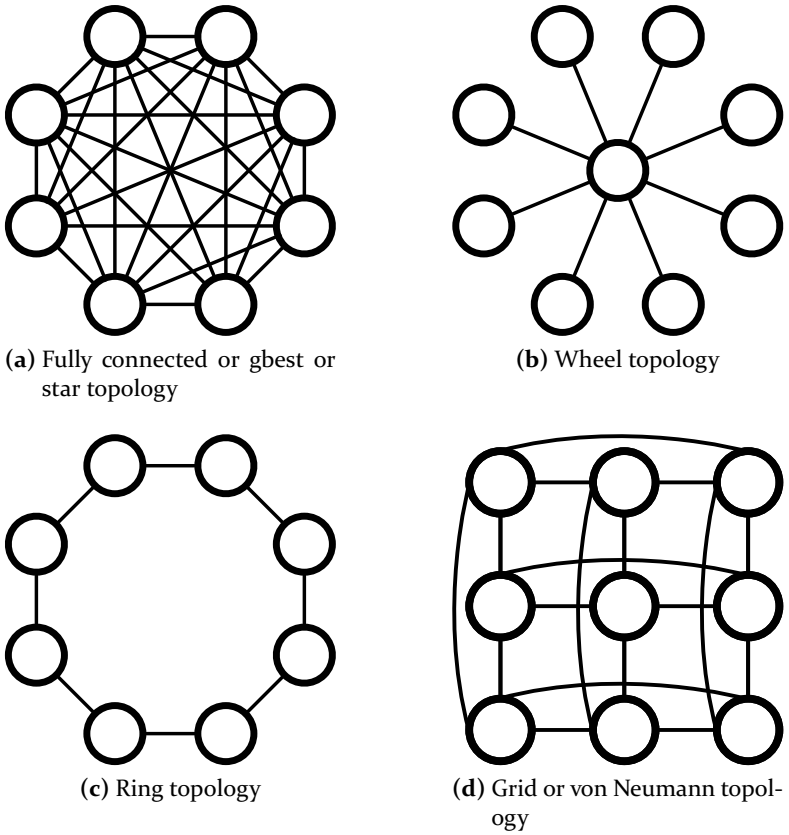
- The *wheel topology* ([Ken99]), in which one specific particle $n_0$ is adjacent to every other particle. Particle $n_0$ acts as a kind of guardian to slow down the distribution of information. Any improvement of some particle has to be confirmed by particle $n_0$ before it gets visible to the whole swarm.

- The *lbest*$(2k)$ *topology* ([EK95]), in which every particle $n$ is a neighbor of the particles $n-k, \ldots, n+k$ (where negative indices $-\ell$ are identified with $N-\ell$). This topology is also called *circles* ([Ken99]). The special case lbest(2) is called the *ring topology*. Especially for small

k, the lbest topology delays the information distribution among the swarm considerably.

- The *grid topology*, also known as *von Neumann topology* ([MKN03]), in which the particles are arranged on a 2-dimensional grid with wrap-around edges, such that every particle has exactly 4 neighbors. This topology is seen as a compromise between the fully connected swarm and the ring topology.

- A *random topology* ([Ken99, KM02]). Instead of choosing one of the above fixed topologies, a random neighborhood graph is generated according to some distribution.



**(a)** Fully connected or gbest or star topology

**(b)** Wheel topology

**(c)** Ring topology

**(d)** Grid or von Neumann topology

**Figure 2.4:** Some commonly used static neighborhood topologies.

Figure 2.4 provides a graphical representation of the different neighborhood topologies. Although a particle could be excluded from its own neighborhood in every of the mentioned topologies, usually each particle is set to be part of its own neighborhood.

In some PSO variants, the topologies occur not in the described pure forms but as a mixture. E. g., in [Ken99], the author uses a ring topology with additional randomly sampled shortcuts. There has been many research on comparing the effects of the different topologies on the quality of the optimization (e. g., [EK95, Ken99, KM02, MKN03]).

### Dynamic neighborhood topologies

According to the literature ([EK95, Ken99, Sug99]), a denser topology increases the convergence speed of the particle swarm, but reduces its capability to explore new areas and therefore increases the risk of the swarm converging towards a local but not particularly good optimum. Explorative behavior is desirable during the early iterations of PSO to find the area around a good local (or maybe even the global) optimum while during the later iterations, the swarm should exploit and converge towards the optimum found in order to provide a good precision, i. e., a solution that agrees with the actual optimum in as many digits as possible. Therefore, the neighborhood topology is in some versions dynamically changed during the runtime.

In [RV03], the neighborhood topology is initially the ring. The whole optimization time is divided into certain time intervals and after the $i$'th interval, every particle $n$ adds particle $n + i$ (where particle $N + \ell$ is identified with particle $\ell$) to its neighborhood. The intervals are calculated such that the swarm becomes fully connected after $4/5$ of the optimization time.

In [MWP04], the neighbors of each particle are initialized randomly in two stages. In the first stage, every particle $n$ chooses the number $|\mathcal{N}(n)|$ of its neighbors randomly between a certain minimum and maximum. In the second stage, $|\mathcal{N}(n)|$ distinct particles are selected as the neighbors visible for particle $n$. Note that in this setting, the neighborhood relation is not symmetric. During the optimization, the topology is dynamically altered by applying a mechanism called *edge migration*. After every iteration, one random particle with more than one neighbor is chosen and one of its neighbors is selected randomly and transferred to another random particle.

In [LS05a], the particles are partitioned into small subswarms of size three to five. Every subswarm is fully connected, but there are no connections between particles of two distinct subswarms. In order to enable information exchange, the particles are periodically and randomly regrouped.

Other approaches change the topology while taking the performance of certain particles or the whole swarm into account. The idea is to increase the influence of successful particles compared to particles that in the past did not contribute much to the optimization task.

In [JM05], the authors propose the *Hierarchical PSO (H-PSO)*, a variant in which the neighborhood topology changes depend on the success of the different particles. The underlying neighborhood graph is a regular tree, implementing a hierarchy with the best particles on top at the root. The neighborhood of every particle $n$ consists of $n$ itself and its parent node. If after an iteration of the H-PSO some particle $n$ has a child with a local attractor better than the one of particle $n$, $n$ exchanges its position with its best child. This is done top-down, i. e., it is possible that one particle moves down several levels within one iteration, but it can move up at most one level.

In [Cle07], different variants of random neighborhood topologies are presented, which are redesigned after either a certain number of iterations or if after a single iteration the best known solution was not improved.

While in all the variants mentioned until now only one member of the neighborhood was actually chosen for the velocity update (see Equation 2.5), there have been some attempts to provide particles with knowledge not only from the best but from every neighbor. This idea leads to the fully informed particle swarm (FIPS) ([MKN04]). Instead of selecting one particular neighbor for the local guide, the mean of the local attractors of every neighbor is calculated. The velocity update equation (Equation (2.5)) changes to

$$V_{t+1}^{n,d} = \chi \cdot V_t^{n,d} + c_1 \cdot r_t^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) + c_2 \cdot s_t^{n,d} \cdot \left(\overline{L}_t^{n,d} - X_t^{n,d}\right),$$

with

$$\overline{L}_t^n = \frac{1}{|\mathcal{N}(n)|} \cdot \sum_{n' \in \mathcal{N}(n)} \hat{L}_t^{n',n}, \tag{2.8}$$

where $\hat{L}_t^{n',n}$ is defined as in (2.7) on page 17.

Comparisons of FIPS and PSO for different neighborhood topologies can be found in [MN04] and in [KM06]. In [JHW08], the authors introduced the *ranked FIPS*, a variant of the FIPS in which the average over the local attractors of all neighbors is not calculated unweighted as in Equation (2.8),

but with weights representing the quality of the neighbors' local attractors. If the neighbors are sorted with increasing function value of the local attractor, then the neighbor $i+1$ has a weight which is one half of the weight of neighbor $i$. By normalizing such that the sum of the weights equals 1, the weights of the local attractors of every neighbor are determined.

### 2.3.2 Constraints and Bound Handling

In most applications, the optimization problem is subject to certain *constraints*, i. e., not every point in $\mathbb{R}^D$ is a feasible solution. For some objective function $f : \mathbb{R}^D \to \mathbb{R}$, constraints are typically given as functions $g_i : \mathbb{R}^D \to \mathbb{R}$, $i = 1, \ldots, m$ where $m$ is the number of constraints and the optimization task is to find the minimum of $f$ over all $x \in \mathbb{R}^D$ with non-positive values for every $g_i$. Formally:

$$\min\{f(x) \mid x \in \mathbb{R}^D, \forall i \in \{1, \ldots, m\} : g_i(x) \leq 0\}.$$

Such constraints are called *inequality constraints*. Some formulations also allow *equality constraints*, i. e., functions $h_j : \mathbb{R}^D \to \mathbb{R}$, which have to be exactly 0 for every feasible solution. Note that an equality constraint $h(x) = 0$ can be formulated as the two inequality constraints $h(x) \leq 0$ and $-h(x) \leq 0$. Since especially equality constraints are hard to fulfill in the black box scenario, it is common to consider every $x \in \mathbb{R}^D$ feasible if the violation of the constraint is below a certain bound $\varepsilon$, which is typically set to $10^{-6}$.

In order to solve constraint optimization problems, many PSO variants that handle such constraints have been developed. The simplest one is to prevent infeasible points from becoming local or global attractor of any particle ([HE02b, HES03a]). This method is equivalent to setting the objective function value $f(x)$ to infinity for every $x$ that violates a constraint. Therefore, this method is sometimes referred to as *Infinity*.

In that sense, when using the Infinity method, all the infeasible positions are treated equally. If the set of feasible solutions is small or disconnected, the Infinity strategy might result in particles that are distracted from the boundary ([Coe02]). Therefore, a generalization ([PC04]) allows to measure the amount of constraint violation. If an infeasible search point is compared to a feasible one, the feasible point is considered the better one. If two infeasible points are compared, the one with the lowest constraint violation wins.

If the constraint functions $g_i$ are sufficiently well-behaved, and if the feasible area is small and hard to find, this mechanism can guide the particles to search points that satisfy the constraints.

Another approach, that does not automatically insist on any infeasible point being worse than any feasible point, is the so-called *penalty* mechanism. In that method, the objective function is altered, such that the resulting optimization problem is unconstrained and has the same optimum as the original problem. This leads to the following modified objective function $F$ ([Coe02]):

$$F(x) = f(x) + \sum_{i=1}^{m} a_i \cdot \max\{0, g_i(x)\}^{\alpha}.$$

Note that this approach requires evaluations of $f$ in infeasible areas and is therefore not always applicable.

The choices of the weights $a_i$ and $\alpha$ are crucial. If they are chosen too high, this method degenerates to the Infinity method. If they are chosen too low, the global minimum of $F$ could be an infeasible point. The right choice of these weights depends on the objective function $f$. Therefore, the Penalty method violates the black box scenario. In order to solve this issue, in [PV02a], the penalty function is altered over time. Another variant of the Penalty approach can be found in [OHMW11], where the private guide and the local guide are chosen with respect to two different Penalty mechanisms.

For the case of more specific constraints, more specialized constraint handling methods have been developed. For example, if the constraints are linear, i. e., if every $g_i$ has the form

$$g_i(x) = \sum_{d=1}^{D} a_{i,d} \cdot x_d - b,$$

then the linear PSO (LPSO) as introduced in [PE03] can be applied, in which the movement equations are altered in a way ensuring that every velocity is in the null space of the matrix $(a_{i,d})_{i=1,\dots,m;d=1,\dots,D}$. Therefore, if the positions are feasible after initialization, they stay feasible forever. In [MS05], one can find a comprehensive discussion and experimental results indicating advantages of such a reduction of the search space dimension in comparison to other bound-handling methods.

Maybe the most important variant of constraints are the so-called *box constraints*, which have the form

$$\forall d \in \{1, \dots, D\} : l_d \leq x_d \leq u_d,$$

i. e., the range of each variable $x_d$ has a lower bound $l_d$ and an upper bound $u_d$. For constraint optimization problems of this form, a large number of constraint handling mechanisms is available in the literature. Examples of methods for handling box constraints are

- **Infinity** ([HE02b]): The function values for points outside the boundaries are set to infinity. This is equivalent to the Infinity method for general constraints.

- **Random** ([HBM13]): If a particle leaves the search space, its position is reinitialized randomly inside the boundaries. A variant is to reinitialize only the position entries in the particular dimensions in which the boundary conditions are actually violated.

- **Absorption**, also known as shrink ([Cle06a]): If at some point in time the updated velocity would result in an updated position outside the search space, it is scaled down by a factor such that the particle ends up on the boundary.

- **Reflect** ([BF05]): The boundaries act like mirrors. If the updated velocity points to a point outside the feasible area, it is reflected at the border.

- **Nearest** ([Cle06a]): If a particle leaves the space of the feasible solutions, it is set to the closest point inside the boundaries.

- **Hyperbolic** ([Cle06a]): Every component of the updated velocity is scaled down with respect to the position and the boundaries. I. e., a positive $V_{t+1}^{n,d}$ is multiplied with

$$\frac{1}{1 + \frac{V_{t+1}^{n,d}}{u_d - X_t^{n,d}}}$$

  and a negative $V_{t+1}^{n,d}$ is multiplied with

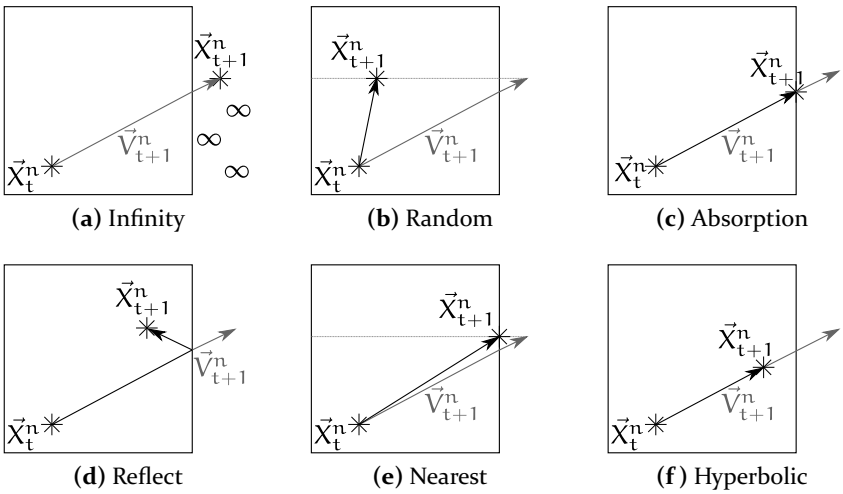
$$\frac{1}{1 - \frac{V_{t+1}^{n,d}}{X_t^{n,d} - l_d}}.$$

- **Periodic** ([ZXB04]): The objective function is periodically repeated, i. e., for $(x_1, \ldots, x_D) \in [l_1, u_1] \times \ldots \times [l_D, u_D]$ and $k_1, \ldots, k_D \in \mathbb{Z}$, one sets

  $$f((x_1 + k_1 \cdot (u_1 - l_1), \ldots, x_d + k_d \cdot (u_d - l_d), \ldots, x_D + k_D \cdot (u_D - l_D))) := f((x_1, \ldots, x_d, \ldots, x_D)),$$

  leading to a function that is defined at every point in $\mathbb{R}^D$.

- **Bounded Mirror** ([HBM13]): The method Bounded Mirror is a combination of Reflect and Periodic. Here, the feasible search space is only doubled in each dimension, and instead of just copying the search space, the objective function is mirrored in order to avoid discontinuities. Additionally, opposite boundaries are connected, i. e., if a particle leaves the extended feasible search space at one boundary, it reenters at the opposite boundary.



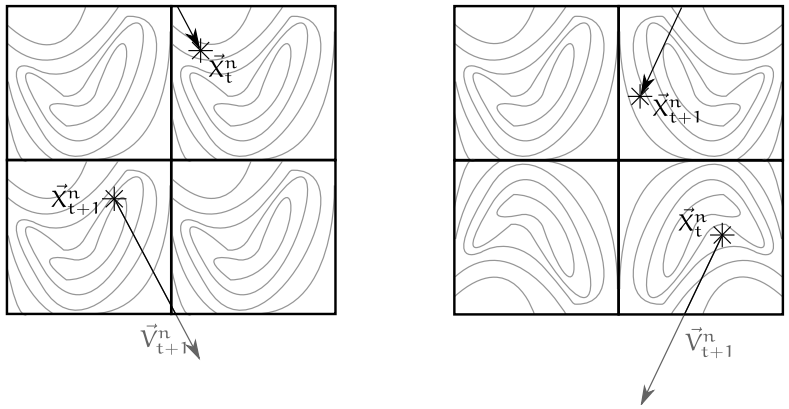

**(a)** Infinity **(b)** Random **(c)** Absorption

**(d)** Reflect **(e)** Nearest **(f)** Hyperbolic

**Figure 2.5:** Constraint handling methods for box constraints (I).

For visualization of the different constraint handling mechanisms, see Figure 2.5 and Figure 2.6.

Additionally to handling positions that are outside the search space, the velocities of a particle violating the box constraints in a certain dimension can also be altered. Typical velocity update strategies are ([HBM13]):




**(a)** Periodic **(b)** Bounded Mirror

**Figure 2.6:** Constraint handling methods for box constraints (II).

- **Zero**: The velocity is set to $0$ in the respective dimension.
- **Deterministic Back**: The respective entry of the velocity is multiplied with $-\lambda$ for some $\lambda > 0$. A typical value for $\lambda$ is $0.5$ ([Cle06a]). Deterministic Back is particularly suitable to be combined with Reflect.
- **Random Back**: Similar to Deterministic Back, but $\lambda$ is drawn uniformly at random from $[0, 1]$.
- **Adjust**: After applying the constraint handling mechanism for the new position, the updated velocity is set to the difference of the new position and the old position.

For a comprehensive study of box constraints and the influence of the different bound handling strategies, see [Hel10].

### 2.3.3 Variants of the Movement Equations

Additionally to adjusting the parameters and the neighborhood topology of the PSO algorithm and additionally to extending the method for the case of constraints, researchers developed variants that significantly deviate from

the classical PSO by substantially modifying the movement equations or hybridization with other methods. The goal is either to make the PSO algorithm even more efficient or to further simplify it without loosing too much of its efficiency.

### Simplifying the Movement Equations

Classical PSO is already a comparatively simple algorithm. Apart from evaluating the objective function, a task which depending on the underlying problem might be expensive, PSO has very moderate demands for resources. For every particle, it stores only a position, a velocity and a local attractor. The computations inside the movement equations consist only of simple arithmetic operations. However, there are still some variants that try to further simplify the algorithm.

The so-called social-only PSO is run without the local attractor ([Ken97, PC10]), i. e., $c_1$ in Equation (2.5) is set to 0. Similarly, the cognition-only PSO is run without the global attractor ([Ken97]) and behaves like N independent "swarms", each consisting of 1 particle.

Another simplification is done in [Ken03], where the Bare Bones PSO is introduced. Experiments suggest that if in the classical PSO both attractors of a particle stay constant, then the distribution of the position of the respective particle approaches a stationary distribution. So, instead of waiting until the distribution converges, the idea of the Bare Bones PSO is to sample the next position according to a D-dimensional Gaussian distribution with mean in the middle between the two attractors and standard deviation equal to the absolute value of the difference between the attractors, which serves as an approximation of the unknown stationary distribution. Therefore, Bare Bones PSO maintains neither an old position nor a velocity.

### Improving the Movement Equations

In some applications, especially those where the evaluation of the objective function is expensive, performance in terms of the quality of the obtained solution is much more important than the simplicity of the underlying algorithm. Over the years, a great variety of extensions has been developed

and experimentally tested. In most extensions, the single particles are made smarter by making them aware of additional information or allowing them to perform more sophisticated operations.

A standard method for improving PSO performance is *parameter adaptation*. Instead of assigning fixed values to the swarm parameters $\chi$, $c_1$ and $c_2$ from Equation (2.5), the values are changed over time either deterministically or randomly. The general idea of parameter adaptation mechanisms is that during the early iterations, the swarm should explore larger areas while in the end of the optimization process, the particles are supposed to converge towards one common point. In order to support this behavior, several parameter adaptation mechanisms have been developed.

The authors of [SE99] use a linearly decreasing $\chi$. In [Fan02], a maximum $V_{max}$ for the absolute value of each entry of the velocity, similar to the variant with velocity clamping described in Equation (2.3), but with a linearly decreasing $V_{max}$. In [CZS06], the value for $V_{max}$ is altered randomly after every iteration. Even more general, in [CCZ09], the authors propose to use a different $V_{max}^{n,d}$ for each particle $n$ and each dimension $d$, where the $V_{max}^{n,d}$ are chosen randomly and independently of each other at every iteration.

The acceleration coefficients $c_1$ and $c_2$ can be made time-varying, too. The PSO version of [RHW04] uses a decreasing $c_1$ and an increasing $c_2$. The reason for that is that the larger the weight $c_2$ of the global attractor is, the faster is the swarm assumed to converge towards the global attractor while a large weight $c_1$ of the local attractor might lure the particles away from the global attractor and therefore prevent too early convergence.

A different and more sophisticated method for parameter adaptation is presented in [RHW10]. Here, the particles are conceptually partitioned into several different groups, called parameter swarms. The parameters $\chi$, $c_1$ and $c_2$ are the same inside each parameter swarm but may vary between different parameter swarms. Depending on the success of the parameter swarms, measured as the update frequency of private and local guides relative to the parameter swarm size, various operations are performed, e. g., parameters are randomly altered, single particles are moved between parameter swarms or the parameters of a parameter subswarm are reinitialized or set to the parameters of some other, better subswarm. This mechanism enables the particle swarm to optimize the objective function and its own parameters in parallel. In particular, this method allows the swarm to adapt its parameters to the exact problem instance instead of making the user of PSO find the suitable parameters for every possible objective function.

In order to prevent particles from stagnating, i. e., from stopping their movement too early, sometimes a particle is given a push if its velocity's absolute value falls below a certain bound. In [RHW04], the velocity of a too slow particle is reinitialized randomly according to a uniform distribution over some interval $[-v, v]$. In [RHW10], the positions of particles with a velocity too close to $0$ are *mutated*, i. e., randomly altered, according to a Gaussian distribution with the old position as its mean and a standard deviation of $c_1/10$.

In the field of evolutionary algorithms (an overview over the methodology of evolutionary algorithms is provided in Section 2.7.1), mutation is a common concept. Mutation stands for a random and typically small variation of an individual, i. e., a point in the search space. The same idea of small random changes can be applied to positions of particles as well. By adding the mutation operation, the movement equations obtain the form

$$\begin{aligned} V_{t+1}^{n,d} &= \chi \cdot V_t^{n,d} + c_1 \cdot r_t^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) + c_2 \cdot s_t^{n,d} \cdot (G_t^d - X_t^{n,d}) + \rho_t^{n,d}, \\ X_{t+1}^{n,d} &= X_t^{n,d} + V_{t+1}^{n,d}, \end{aligned}$$

with some $\rho_t^{n,d}$ chosen randomly according to some distribution.

In TRIBES, a swarm algorithm proposed in [Cle03], the mutation $\rho_t^{n,d}$ is chosen according to a Gaussian distribution with mean $0$ and standard deviation $(f(L_t^n) - f(G_t^n))/(f(L_t^n) + f(G_t^n))$.

The PSO variant from [RHW04] uses an approach where $\rho_t$ equals $0$ if the global attractor has been improved during the previous iteration. Otherwise, a particle $n$ and a dimension $d$ are selected uniformly at random and the corresponding $\rho_t^{n,d}$ is chosen according to a uniform distribution over either a fixed or a time-varying interval.

In the Guaranteed Convergence PSO (GCPSO) as introduced in [vdBE02], $\rho_t^{n,d}$ is chosen uniformly from the interval $[-p, p]$, where $p$ is initially set to 1. If the number of consecutive iterations in which the global attractor is updated reaches a certain bound $s_c$, then it is assumed that there is still much room for improvement and in order to accelerate the particles, $p$ is doubled. Similarly, if the number of consecutive iterations in which the global attractor is not updated reaches a certain bound $f_c$, the authors assume that the area for improvement is small. Therefore, they refine the search by halving the value of $p$.

Rather than just deciding about the mutation range, the information of previous successes or failures of certain particles can be utilized even more. In a PSO variant called TRIBES ([Cle03]), the whole neighborhood topology

depends heavily on the successes of the particles in updating their attractors. More precisely: The particles are conceptually partitioned into different groups called *tribes*. Any two particles inside the same tribe are connected, while the interconnections between different tribes are rather loose.

After every $k$ steps, the neighborhood topology is updated according to the following rules. Particles are called good if they updated their local attractor during the previous iteration and tribes are called good if they contain at least a certain percentage of good particles. Since good tribes are already successful, they lose their worst particle, i. e., the particle with the worst local attractor among all particles in the tribe is discarded and its connections to other tribes are redirected to the best particle of its tribe. Since the bad tribes might need some assistance, but the information inside such tribes is not considered very valuable, every bad tribe generates a new particle that is initialized completely independent of its father tribe. All the particles produced this way form a new tribe and each of the new particles is connected to the best particle of its father tribe. The TRIBES algorithm is started with just a single tribe consisting of only one particle.

A further advanced successor of TRIBES can be found in [RHW10], where, amongst other modifications of the original PSO like parameter adaptation and mutation, every subswarm has a leader, which is the particle with the best local attractor. The leaders see each other and are seen by their respective subswarms but they do not see the information of their subswarms, i. e., the neighborhood topology is not symmetric. Every $N$ iterations, the swarm is adapted, i. e., the particles are treated according to their success during the previous two iterations.

If a particle did not update its local attractor during both previous iterations, it is deleted. If the most recent attractor update of a particle was two iterations ago, then with probability $s$, its velocity is reinitialized. Otherwise, its position and velocity are set to the respective values of its subswarm's leader and a mutation is performed. Here, $s$ is a parameter that decreases over time from 1 to 0. If a particle updated its local attractor during the previous iteration, but did not overcome the best local attractor among all particles inside its subswarm, no changes happen to it. If a particle even found the best position of all particles inside its subswarm, this particle is doubled.

Similar to TRIBES, a subswarm in which at least a share of $s/2$ particles updated their local attractors during the previous two iterations is called good and looses its worst particle. The only exception from this rule is the case when a subswarm consists only of its leader and this particle happens to have

the best local attractor among all leaders. Then, this particle survives. Again similar to the mechanism in TRIBES, subswarms that are not good are called bad and produce a new particles. All the particles born this way form a new subswarm together. Different from TRIBES, there are no direct connections between a generated particles and its father subswarm, only the new leader is connected with all other leaders.

Although they sometimes substantially modify the PSO algorithm, the previously mentioned extensions of PSO all preserve the movement equations. However, there are some variants that completely reinterpret them and therefore alter the core of PSO. Since the coordinate dependent formulation of the movement equations sometimes yields undesirable behavior, e. g., a great amount of movement in some dimensions and almost no movement in others, the authors of [BML14] attempt to improve performance by applying a random rotation in every dimension. In [HNW09], the authors take control over the convergence speed by normalizing the velocity to the length $v$, where $v$ is a parameter that gets doubled or halved if after a certain number of iterations the number of attractor updates is sufficiently high, respectively sufficiently low.

In [Cle03], the movement equations are completely replaced by the so-called pivot method. The particles no longer have a velocity and the new search point is calculated as follows. Let $R$ be the distance between the local and the private guide of a particle. Then, one point $P_p$ inside the ball of radius $R$ around the private guide and one point $P_l$ inside the ball with the same size around the local guide are sampled. The new position is the weighted mean of $P_p$ and $P_l$ with weights depending of the function value at the local and the private guide.

Quantum PSO (QPSO) as introduced in [SFX04] is another variant which substantially modifies the movement equations by transferring the idea of PSO from newton mechanics to quantum mechanics. The resulting movement equations of the QPSO are

$$p_t^n = (c_1 \cdot L_t^n + c_2 \cdot G_t^n)/(c_1 + c_2), \tag{2.9}$$

$$\ell_t^{n,d} = \pm\beta \cdot |X_t^{n,d} - p_t^{n,d}|, \tag{2.10}$$

$$X_{t+1}^{n,d} = p_t^{n,d} + \ell_t^{n,d} \cdot \ln(1/r_t^{n,d}), \tag{2.11}$$

where $c_1$, $c_2$ and $\beta$ are positive parameters and the $r_t^{n,d}$ are uniformly and independently distributed over $[0, 1]$. The sign in Equation (2.10) is also de-

cided uniformly at random every time the equation is applied. In [dSC08], Equation (2.10) was replaced by

$$\ell_t^{n,d} = \pm\beta \cdot |X_t^{n,d} - \text{Mbest}_t^d|,$$

where $\text{Mbest}_t$ is the mean of all local attractors. This version of QPSO can be seen as a Fully Informed QPSO (see Section 2.3.1).

Additionally to modifying the PSO algorithm itself, there is a great variety of hybrid algorithms, consisting of PSO and some other method, which can as well be another nature-inspired algorithm like an evolutionary algorithm ([GAHG05]), a classical mathematical method like the Quasi-Newton method ([LS05b]) or a special algorithm designed for a specific application like the back-propagation algorithm which is used for training neural networks ([ZZLL07]).

## 2.4 Multi-Objective Particle Swarm Optimization

Many applications are subject to not only one but several objective functions, e. g., in hardware design, cost should be minimized while at the same time reliability should be maximized. In such a situation, there is not one single optimum since improving one objective typically worsens some other objective. Therefore, the task of optimization becomes more complicated. The goal is to provide a set of preferably different and "good" solutions to show the possible trade-offs between the different objectives for an external decision maker. A comprehensive survey about multi-objective PSO can be found at [SC06].

### 2.4.1 Multi-Objective Black Box Optimization

Formally, a multi-objective black box optimization problem is represented as a function $f = (f_1, \ldots, f_k) : S \subset \mathbb{R}^D \to \mathbb{R}^k$. The image of $S$ under $f$ is called the objective space of the problem. Similar to the single-objective black box optimization problem, the function $f$ can only be evaluated pointwise. Without loss of generality, one can assume that each $f_i$ is to be minimized. Figure 2.7 gives a graphical representation of the described situation.

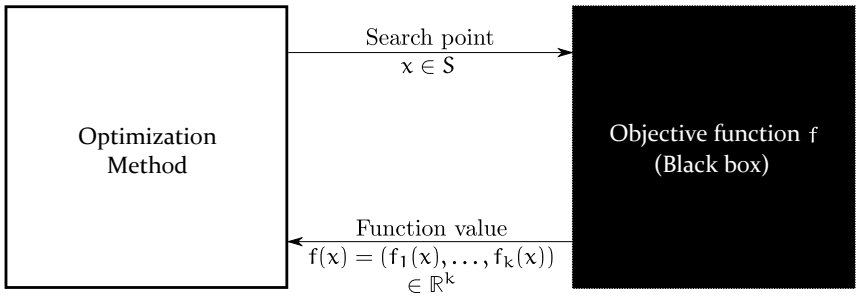



**Figure 2.7:** Multi-objective black box optimization.

An important concept for comparing the quality of different values from the objective space is the so-called dominance. For $y_1, y_2 \in \mathbb{R}^k$, we say that $y_1$ *weakly dominates* $y_2$ ($y_1 \preccurlyeq y_2$), if $y_1 = (y_1^1, \ldots, y_1^k)$ is componentwise less or equal to $y_2 = (y_2^1, \ldots, y_2^k)$. In terms:

$$y_1 \preccurlyeq y_2 :\Leftrightarrow \forall i \in \{1, \ldots, k\} : y_1^i \leq y_2^i.$$

In this case, $y_1$ is at least as good as $y_2$ for the purpose of optimization. Having $y_1$ better than $y_2$ requires additionally, that $y_1$ is strictly less than $y_2$ in at least one component, in terms:

$$y_1 \prec y_2 :\Leftrightarrow \left(\forall i \in \{1, \ldots, k\} : y_1^i \leq y_2^i \wedge \exists i \in \{1, \ldots, k\} : y_1^i < y_2^i\right)$$

In that case, we say that $y_1$ dominates $y_2$. If neither of the points $y_1$ and $y_2$ weakly dominates the other, i. e., if there are $i_1, i_2 \in \{1, \ldots, k\}$, such that $y_1^{i_1} < y_2^{i_1}$ and $y_1^{i_2} > y_2^{i_2}$, $y_1$ and $y_2$ are called *incomparable*. To shorten notation, we say that a search point $x_1$ (weakly) dominates or is incomparable with a search point $x_2$, if $f(x_1)$ (weakly) dominates, respectively is incomparable with $f(x_2)$. A search point $x^* \in S$ is called *Pareto optimal*, if there is no $x \in S$ with $f(x) \prec f(x^*)$. The set of all Pareto optimal points is called the *Pareto optimal set*. The image of the Pareto optimal set is called the *Pareto optimal front* or for short *Pareto front*. Figure 2.8 illustrates the concept of Pareto dominance.

Approximating the Pareto optimal set is the main goal of multi-objective optimization. A solution $A \subset S$, i. e., a set of search points, is generally considered a good approximation of the Pareto optimal set if the points inside $A$ are close to the true Pareto optimal set and if they are not too close to each other, i. e., if they have a certain spread and diversity. In particular, a typical requirement for $A$ is to consist only of points that do not dominate each other.

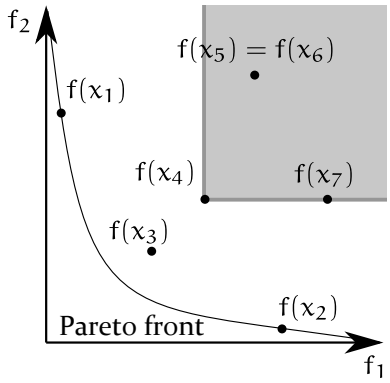



**Figure 2.8:** Example of a 2-dimensional objective space. The gray area indicates the set of all points in the solution space dominated by $f(x_4)$, i. e., $f(x_4)$ dominates $f(x_5) = f(x_6)$ and $f(x_7)$. $x_1$ and $x_2$ are Pareto optimal, therefore their function values are not dominated by any other point. $f(x_3)$ is also not dominated by any of the other drawn points. From the given point set, the non-dominated points $f(x_1)$, $f(x_2)$ and $f(x_3)$ form the best approximation of the Pareto front.

### 2.4.2 PSO for Multi-Objective Black Box Optimization

An early approach for solving multi-objective optimization problems using PSO can be found in [PV02b], where the authors rewrite the multi-objective problem as several single-objective problems. This is done by constructing a weighted sum $F$ of the objective functions, in terms:

$$F(x) := \sum_{i=1}^{k} w_i \cdot f_i(x).$$

For every choice of positive $w_i, i = 1, \ldots, k$, the minimum of $F$ is Pareto optimal with respect to $f$. In order to achieve a good approximation of the Pareto optimal set, the PSO algorithm is restarted with different choices of the weights $w_i$.

Most multi-objective PSO (MOPSO) variants are explicitly aware of the underlying multi-objective problem structure. Such variants need additional mechanisms to take care of multiple incomparable function values, in particular the decision about updates of attractors. It is clear that if a new search

point dominates the local attractor of some particle, then the attractor is updated. If on the other hand the local attractor dominates the new search point, its old value is kept. For the case of incomparability between the local attractor and the new search point, different strategies are known in the literature. Some are easy and natural like picking one of the two values randomly ([Coe02]) or deterministically keeping the old local attractor ([HE02a, MT03, BM06]) or updating the attractor to the new search point ([ABEF05, BM06]). Another strategy proposed in [BM06] is to decide with respect to the sum of all objective values.

Other variants of MOPSO allow each particle to have a list of pairwise non-dominated search points as local attractor list instead of just one point. In that case, we need a strategy to select one point of the list in order to apply the movement equations. This decision can be done uniformly at random ([MC99, BM06]). A collection of other, more refined methods is presented and experimentally evaluated in [BM06]. Here, just the basic ideas are repeated.

For instance, one could choose the point from the list that is closest to the global attractor (after applying some strategy to decide which point serves as global attractor). Since a good solution of a multi-objective optimization problem consists of a set of points with a certain diversity, convergence of the swarm towards a single search point is not desired. In order to keep the diversity up, some variant calculates the distances between the points in the local attractor list of particle $n$ and the positions of the closest other particles. The point that maximizes this distance is chosen as the local attractor. Another similar criterion is to pick the point $L_t^n$ from the list that maximizes the weighted sum of the objectives, where the weights are proportional to the objective values of the current position. In terms, the weighted sum is formulated as follows.

$$\sum_{i=1}^{k} \underbrace{\frac{f_i(X_t^n)}{\sum_{j=1}^{k} f_j(X_t^n)}}_{=:w_i} f_i(L_t^n).$$

Note that the $f_i$ are assumed to be positive here. The idea of this method is to choose the point that is the closest to the current position in the objective space (not in the search space) and therefore make the particles maintain a certain diversity. Another strategy works similar to the idea of the FIPS from Section 2.3.1, i. e., the local attractor is replaced by the mean of all points in the local attractor list.

A completely different approach is to discard the local attractor completely and to use only the global attractor ([BM06]).

The selection of the global attractor, respectively the local guide, is crucial for the success of the MOPSO because of its two opposite goals, namely finding points as close as possible to the Pareto optimal set and therefore being able to decrease the movement of the swarm, but at the same time finding a diverse variety of points to reflect a large part of the Pareto optimal set. In most PSO versions, the concept of neighborhood topology is dropped and the possible global attractors, i. e., all points visited by a particle that are not dominated by any other already visited point, are stored and maintained in a global archive.

One exception is the version proposed in [MC99], where the local guide is chosen randomly from the non-dominated subset of the union of all private guides from neighboring particles ([MC99]). Of course, choosing at random from the global archive is also possible. This idea is further refined in [Coe02] by taking the distance of the search points' function values in the objective space into account. If two members of the set of potential global attractors yield too close objective values, then their probability to be chosen is decreased.

In order to maintain the diversity, it is desirable to choose the point from the archive that is closest to the current position with respect to some distance measure over the objective space. In [MT03], the authors propose a PSO based on such a distance measure, which they call the $\sigma$-method. They define the difference between two search points $x$ and $y$ as the Euclidean distance $\|\sigma(f(x)) - \sigma(f(y))\|_2$ between the respective $\sigma$-values, defined as

$$\sigma(f_1(x), \ldots, f_k(x)) = \left( \frac{(f_i(x))^2 - (f_j(x))^2}{\sum_{\ell=1}^{k} (f_\ell(x))^2} \right)_{1 \le i < j \le k}.$$

Another approach relying on distance in the objective space and specifically designed for the case of $k = 2$ objectives can be found in [HE02a]. Here, the authors select the global attractor for particle $n$ by taking only the $m$ particles which have the closest $f_1$-value into account and selecting the one with the best $f_2$-value.

Most of the approaches based on weighted sums or distance in the objective space lead to irregular behavior if the multi-objective optimization problem that is to be solved contains objective functions with very heterogeneous scales. E. g., if $f_1$ is orders of magnitude higher than the other objective functions, then a weighted sum will prefer points with lower $f_1$-value

and also the σ-method degenerates to measuring the distance of the respective $f_1$-values. Therefore, in [ABEF05], the authors propose three different strategies for determining a global attractor out of the global archive that are solely based on dominance and therefore not susceptible to differences in the scales of the objective functions. These strategies are

- ROUNDS: Among the points of the archive that dominate at least one current positions of the swarm, the one that dominates the smallest number of current positions is determined. Then, this entry of the global archive is randomly assigned as the global attractor to one of the particles it dominates. From the remaining archive members, again the one dominating the smallest number of current positions is selected and randomly assigned and so on. This is repeated until every particle has a global attractor. If there are too few entries in the archive to provide each particle with a global attractor, a new round starts with the remaining particles where the entries have the chance to be assigned to another particle. The idea here is to prefer entries that are in sparse regions of the objective space.

- RANDOM: Every particle selects uniformly at random one of the points in the archive that dominates its current position. If there is no such point, e. g., if the current position of the particle is in the archive, the particle selects uniformly at random a point from the complete archive.

- PROB: Similar to RANDOM, but the probability for selecting a specific entry $a$ of the archive is proportional to $1/N_a$, where $N_a$ is the number of particles with a current position dominated by $a$.

## 2.5 Particle Swarm Optimization for Discrete Problems

Although the method is originally invented to handle continuous search spaces, i. e., search spaces $S \subset \mathbb{R}^D$, some authors proposed PSO versions designed to handle optimization problems over discrete search spaces. Such problem settings require a fundamental reinterpretation of the movement equation because operations like multiplying the difference between two positions with a random number do not necessarily yield a feasible point in the discrete search space.

An early discrete PSO version is the binary PSO ([KE97]), designed to handle optimization problems over the search space $S = \{0,1\}^D$. In the classical PSO, the velocity of a particle represents a move, i. e., the difference between the previous and the updated position, which here only has a discrete range of values. The binary PSO understands the velocity as a probability, without changing the velocity update as defined in Equation (2.5). Instead of applying Equation (2.6) for the position update, each component of the velocity is mapped into the real interval $[0,1]$ by some transformation $T$ and $X_{t+1}^{n,d}$, the $d$'th entry of particle $n$'s updated position, is set to 1 with probability $T(V_{t+1}^{n,d})$. Otherwise, $X_{t+1}^{n,d}$ is set to 0.

A more general approach for a search space of the form $S = \{0,\dots,M-1\}^D$ is proposed in [VOK07]. Here, instead of applying Equation (2.6) for calculating the updated position, the authors sample a value from the real interval $[0, M-1]$ according to a distribution depending on $V_{t+1}^{n,d}$ for every particle $n$ and every dimension $d$. This value is then added to $X_t^{n,d}$ and the result is rounded in order to obtain an updated position entry in $\{0,\dots,M-1\}$.

Structurally even more different from $\mathbb{R}^D$ than the previously mentioned discrete search spaces is the space of all permutations, which is the search space of, e. g., the Traveling Salesperson Problem (TSP). Not only the operation of "multiplying" a search point, i. e., a permutation, with a real number, but also calculating the difference between two positions, e. g., the current position of a particle and its local or global attractor, require a redefinition.

In [Cle04], a PSO version for solving the TSP and therefore for working over the space of permutations is introduced. The positions of the particles are seen as permutations and the velocities are identified with sequences of transpositions, i. e., sequences of exchanges of two elements. The difference between two positions $x$ and $y$ is a permutation that transforms $x$ into $y$. Since that way the "difference" is not uniquely defined, not even if we additionally demand a shortest sequence of transitions transforming $x$ into $y$, the authors state an algorithm for calculating such a sequence and define the difference as whatever the algorithm outputs. Two such transition lists are "added" by simple concatenation. A velocity is added to a position by applying the velocity-permutation to the position. Multiplying the velocity with some factor $c \in \mathbb{R}$ works as follows. If $c \in [0,1]$ and $V_t^n$ consists of $|V_t^n|$ transpositions, then $c \cdot V_t^n$ is defined as the first $\lfloor c \cdot V_t^n \rfloor$ entries of $V_t^n$. If $c$ has the form $k + \tilde{c}$ with $k \in \mathbb{N}$ and $\tilde{c} \in [0,1]$, then $c \cdot V_t^n$ consists of $k$ concatenated copies of $V_t^n$, followed by $\tilde{c} \cdot V_t^n$. Finally, a velocity is multiplied with $-1$ by inverting the order of the transpositions.

Another, even more refined version of a PSO for permutation problems can be found at [SLL+07]. Although both variants are made for the TSP, they can without further changes be applied to any optimization problem over the permutation space.

There is a broad variety of other discrete PSO variants, most of them specific to a certain problem. For instance, the $n$-*Queens-Problem* is the problem to place as many queens as possible on a chess board of size $M \times M$ without any two queens attacking each other, i. e., no two queens can be placed in the same row, the same column or on the same diagonal. This problem has been addressed with a PSO variant in [HES03b]. Another example for a combinatorial optimization problem solved by a discrete PSO variant is the *single-machine total weighted tardiness scheduling problem* ([AP09]), an optimization problem where the task is, roughly speaking, to schedule jobs with a fixed duration and a fixed deadline to a machine, such that as many of the jobs as possible are finished before their deadlines end.

## 2.6 Theoretical Results about Particle Swarm Optimization

Since PSO has empirically shown great successes in many applications, researchers have tried to understand the behavior of the particle swarm. It turns out that formally analyzing the two innocent looking movement equations from Equation (2.5) and Equation (2.6) is a great challenge because of the stochastic nature of the algorithm and the interactions between the particles. Therefore, formal results known in the literature are typically subject to simplifying assumptions or analyze a restricted version of the PSO algorithm.

In order to simplify the swarm dynamics, one assumption generally made is that the global and local attractors are constant ([OM99, CK02, YII03, Cle06b, PB07]). Under this assumption, we can see from the movement equations (Equations (2.5) and (2.6)), that different dimensions and different particles are stochastically independent, so it is sufficient to just analyze one particle in one dimension. Furthermore, in early analyses the random nature of the PSO algorithm is neglected for further simplification. Instead, it is assumed that all the random variables $r_t^{n,d}$ and $s_t^{n,d}$ are constant.

In [OM99, YII03], the authors calculate characteristics of the trajectories of such a deterministic PSO with constant attractors. If local and global at-

tractor of a particle are identical, then the trajectory of the particle has the form of a sine wave around the attractor. A generalization of the trajectory analysis where the *time* is considered continuous rather than discrete as in the classical, iteration based PSO can be found in [CK02]. Analyses on the characteristics of the trajectories for the classical randomized PSO can be found in, e. g., [Cle06b, PB07]. Here, one goal was to provide sufficient conditions on the parameters $\chi$, $c_1$ and $c_2$ to prove that under the given assumptions a swarm does not explode, i. e., accelerate beyond every bound and move arbitrary far away from its attractors. In particular, the desired behavior of a particle swarm with well chosen parameters is that, given both attractors are located at the same point, the particles converge towards this point. For an overview over the most important results about trajectory analysis, see [vdBE06].

In [KSF06], the authors choose a different approach and calculate sufficient conditions for the randomized PSO to behave asymptotically stable in the sense of Lyapunov stability of passive systems.

Another major field for theoretical investigations of the PSO algorithm is the development of parameter selection guidelines for $\chi$, $c_1$ and $c_2$ by studying the expectation, the variance and higher moments of the particles' positions. Again, such examinations are limited to the case of constant attractors. Based on the analysis of the deterministic PSO in [Tre03], the authors of [JLY07b, JLY07a] calculated sufficient conditions for the parameters to avoid *explosion*, the phenomenon of velocities increasing beyond all bounds. They could prove that if certain restrictions on the parameters hold, the expectation and the variance of each particle's position converge towards a value depending on the location of the local and the global attractor. Furthermore, if the local and the global attractor are identical, then the expected position converges towards the attractors and its variance converges towards $0$. The details of their results will be presented in Section 3.5.

The results from [JLY07b, JLY07a] have been generalized to other PSO variants and higher moments. The generalizations can be found in, e. g., [PBBK07] and [Pol08].

The *finite element method* ([PLCS07]) is a completely different approach for simplifying the analysis of continuous optimization methods. The finite element method works as follows. First, the continuous search space $S$ is partitioned into finitely many disjoint subspaces $(S_i)_{i=1,\dots,M}$, e. g., a cube-shaped search space could be partitioned into a grid of subcubes ([PL07]). Then, the algorithm is discretized by only allowing the objective function evaluation from the centers of the $S_i$. In [PL07], this is done for the Bare

Bones PSO (see Section 2.3.3), where the updated position of a particle is chosen according to some distribution, depending on the local and the global attractor. In order to apply the finite element method, the originally continuous update distribution is altered in order to guarantee that the next position is again in the center of some subcube $S_i$. The resulting discrete system is easier to analyze. By choosing $M$ large, one hopes for a good approximation of the continuous situation.

There are indeed some completely rigorous results that do not rely on additional assumptions. In [HW08], it is proved without further assumptions that for a box-constrained search space with a high dimension $D$, all the particles of the swarm leave the feasible area with overwhelming probability, i. e., with probability $1 - e^{-\Theta(D)}$. For the binary PSO (see Section 2.5), the authors of [SW08] provide various runtime results, e. g., a general lower bound of $\Omega(D/\log(D))$ for every function with a unique global optimum and a bound of $\Theta(D \cdot \log(D))$ on the function Onemax, defined as

$$\textsc{Onemax}((x_1, \ldots, x_D)) = \sum_{d=1}^{D} x_d.$$

Furthermore, there are some negative results, for which a quick overview is given here. See Section 3.5 for the full details. Many work in the literature addresses the phenomenon of so-called *stagnation*, which occurs when

$$X_t^1 = L_t^1 = \ldots = X_t^N = L_t^N = G_t = z, \quad V_t^1 = \ldots = V_t^N = 0,$$

i. e., when all particles and attractors are at the same point $z$ and all velocities are zero. Once in this situation, the swarm stops its movement, no matter how good or bad $f(z)$ really is. Therefore, so the claim, PSO cannot be considered an optimizer ([PE03, Wit09, vdBE10]). To resolve this issue, in [Wit09], the GCPSO (see Section 2.3.3) is examined instead, and a runtime result for the GCPSO with only a single particle, optimizing the objective function Sphere is provided.

Another, yet more refined negative result is presented in [LW11]. Here, the authors state that even if the swarm is not in the previously mentioned state of stagnation, it might under certain conditions converge towards a non-optimal point in the search space and therefore the hitting time for any sufficiently small neighborhood of the optimum has an infinite expectation. In particular, that happens frequently if the swarm size $N$ is 1. Their argument will be repeated in detail in Section 3.5. As a consequence, they propose the

so-called Noisy PSO, a PSO variant that adds a small random perturbation to the velocity at every step. The authors of [LW11] prove that under some conditions on the parameters $\chi$, $c_1$ and $c_2$, and for the 1-dimensional objective function $\text{SPHERE}^+$, defined as

$$\text{SPHERE}^+(x) := \begin{cases} x^2, & \text{if } x \geq 0, \\ \infty, & \text{otherwise,} \end{cases}$$

the expected time until the Noisy PSO hits the $\varepsilon$-neighborhood of the optimum at 0 is finite.

## 2.7 Other Nature-Inspired Meta-Heuristics

Additionally to the swarm behavior of bird flocks and fish schools, other phenomena from the nature have been studied in order to build further nature-inspired black box optimization methods. Examples are manifold. Most heuristics try to simulate and utilize the capabilities of certain animals like, e. g., the foraging strategy of honey bees ([LT01]), the capability of fireflies to lure potential mates to their position by producing light flashes ([Yan09]), the echolocation of bats ([Yan10]) and the brood parasitic behavior of cuckoos ([YD09]).

Other heuristics are based on physical phenomena. For instance, the algorithm Simulated Annealing (SA) ([KGV83, SK06, BSMD08]) is based on the behavior of metal slowly cooling down and reaching a stable state of minimal energy. Another meta-heuristic is the so-called Harmony Search, an algorithm based on the improvisation of a music player.

In the following, two of the most popular nature-inspired meta-heuristics, namely EAs and ant algorithms (AAs) are briefly described.

### 2.7.1 Evolutionary Algorithms

The wide class of evolutionary algorithms is inspired by Darwin's theory about biological evolution. A *population* of *individuals*, each one representing a point in the search space, compete against each other and only the "best" survive and reproduce. Concrete representatives of this scheme can be found in, e. g., [SP97, BS02, MS96].

### The Evolutionary Cycle

In general, an EA works according to the so-called *evolutionary cycle*, as described in the following and visualized in Figure 2.9. For some fixed parameters $\mu$ (the population size) and $\lambda$ (the offspring size), $\mu$ individuals are randomly initialized over the search space. Then, a *mating selection* method is applied in order to select $\lambda$ pairs of individuals as parents of the next generation. The selection is done taking into account the objective function value obtained by the respective individuals, i. e., the better an individual's function value is, the higher are its chances to become a parent. Many concrete selection methods are known. For example, *rank-based selection* calculates the ranking of the individuals according to their objective function values. Then, $2 \cdot \lambda$ times an individual is selected randomly and with a probability distribution depending on the rank, i. e., the best individual gets the highest probability. Another example is the *tournament selection*, where $2 \cdot \lambda$ times a subset of two or more individuals is chosen uniformly at random from the population and the best individual of the chosen subset is selected for mating. Note that usually the individuals of an EA have no gender and are not monogamous.

After having the parents selected, the next generation, called the *offspring*, is created from the parents according to a *cross-over operator* $S \times S \rightarrow S$. This operator heavily depends on the structure of the search space $S$. The idea is to combine the strengths of two already good parent individuals in order to create an even better individual, having the advantages of both its parents.

In order to maintain a certain diversity inside the population, a *mutation operator* is applied to the offspring. Typically, the mutation is done randomly and with a small standard deviation, such that the change of the individual is low. The reason for this is that the child is supposed to have as much in common with its parents as possible.

After the mutation, there are $\mu + \lambda$ individuals in the population. Since a constantly growing population is undesirable, a second round of selection is applied, the so-called *environmental selection*. Basically, similar selection operators as in the parental selection can be used to select $\mu$ out of the $\lambda + \mu$ individuals. If the surviving individuals are chosen only out of the offspring, the EA-variant is called $(\mu, \lambda)$-EA. If on the other hand the individuals of the parent generation also have a chance to survive, then the variant is called $(\mu + \lambda)$-EA.

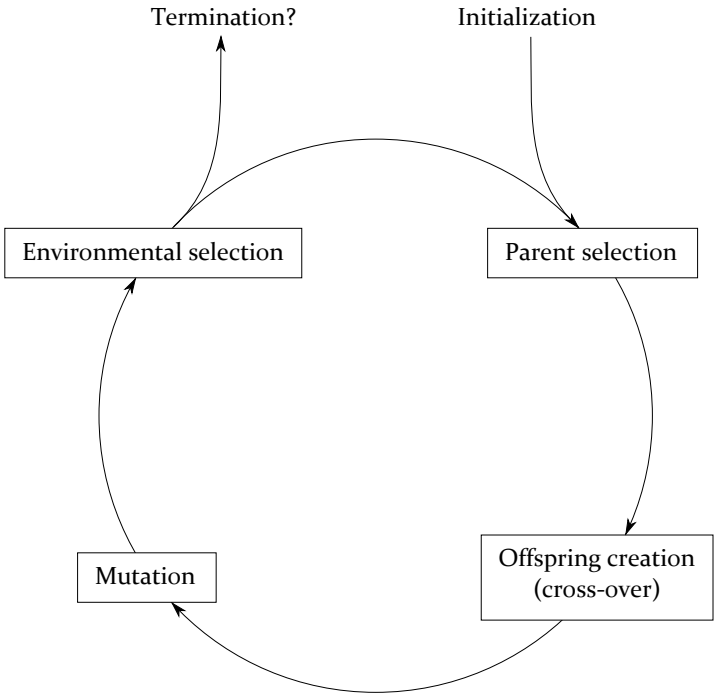



**Figure 2.9:** The evolutionary cycle.

In order to prevent the population from general worsenings, sometimes an additional mechanism called *elitism* is applied, which means that the one individual with the best objective function value among the population is selected to survive automatically.

If after the environmental selection no termination criterion is reached, the cycle is started again with the parental selection phase.

### Theoretical Results about Evolutionary Algorithms

An important difference between EAs and PSO is that an EA can still be effective with only a single individual, while, as stated in Section 2.6, PSO with only one particle does not optimize anything. Therefore, researcher started formal investigations on the $(1 + 1)$-EA, i. e., a simplified variant of the EA with a population consisting of only one individual. Consequently, the $(1 + 1)$-EA does not contain any cross-over mechanism. Instead, the

offspring to mutate is just a copy of the individual. After mutation, instead of a randomized environmental selection from the original and the mutated individual, the one with the best objective function value survives deterministically. The basic pattern of the $(1+1)$-EA is stated in Algorithm 2.

**Algorithm 2:** $(1+1)$-EA

**input** : Objective function $f : S \to \mathbb{R}$ to be minimized
**output**: $X \in S$
// Initialization
Initialize individual $X \in S$ randomly;
**repeat**
  $X' := \text{mutate}(X)$;
  **if** $f(X') < f(X)$ **then**
    $X := X'$;
**until** *Termination criterion holds*;
**return** $X$;

For such a simplified EA and certain search space structures and objective functions, rigorous runtime bounds can be proved. In [DJW02], the authors study a $(1+1)$-EA for optimizing functions over the search space $\{0,1\}^D$. The mutation operator of their EA flips each entry of the individual $X \in \{0,1\}^D$ independently and with probability $1/D$. A number of runtime results are proved. The expected time of the $(1+1)$-EA for finding the global optimum, in the following called the expected optimization time, is at most $D^D$, since mutating any position $X$ into any other position $X^*$, in particular with $X^*$ as the global optimum, has probability at least $D^{-D}$. The authors provide examples of objective functions for which the expected optimization time is indeed $\Theta(D^D)$. On linear functions, i. e., functions of the form

$$f(x) = \sum_{d=1}^{D} w_d \cdot x_d,$$

the $(1+1)$-EA is proved to have an expected optimization time of $\Theta(D \cdot \log(D))$. Furthermore, if the mutation probability is altered by more than a constant factor, also the expected optimization time increases by more than a constant factor. Therefore, the choice of mutation rate is optimal for solving linear objective functions.

The optimization problems over $\{0,1\}^D$ are to some extent artificial and far away from real world problems. Therefore, the actual point of the analysis of

an $(1+1)$-EA is not the resulting runtime bound on its own. Significant are the underlying techniques that can be generalized to analyze EAs in different and more general situations. In [STW04], EAs for sorting are presented. The authors study different objective functions and mutation operators for the respective problem. The resulting runtime bounds for sorting $n$ elements with an EA range from $\Theta(n^2 \cdot \log(n))$ to an in $n$ exponentially increasing lower bound, depending on the exact choice of the objective function measuring the "sortedness" of unsorted sequences and the choice of the mutation operator.

Further examples for combinatorial optimization problems, which can be solved with an EA within certain, formally proved time bounds, are the single source shortest path (SSSP) problem ([STW04]), the maximum matching (MM) problem ([GW03]) and the minimum spanning tree (MST) problem ([NW08]).

In [Jäg03], one can find an attempt to analyze $(1+1)$-EAs for continuous objective functions $\mathbb{R}^D \to \mathbb{R}$. The author presents a version of the $(1+1)$-EA with an adapted mutation operator, where, roughly speaking, the variance of the mutation is altered depending on the success of the algorithm. For functions that behave like the Euclidean distance to some point $o \in \mathbb{R}^D$, i. e., for functions $f$ satisfying

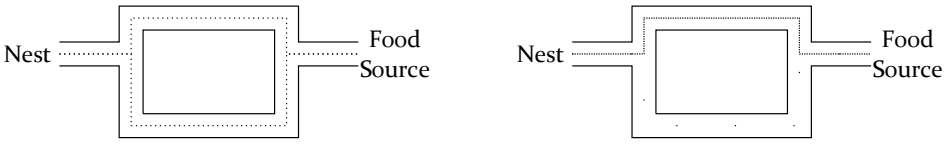
$$|x - o| < |y - o| \Rightarrow f(x) < f(y),$$

the author proves that the time for halving the distance between the individual and the global minimum $o$ is linear in $D$.

### 2.7.2 Ant Algorithms

Ant algorithms are inspired by the famous double bridge experiment of Goss et al. ([GADP89]). A colony of Argentine ants was set into an artificial environment consisting of their nest and a food source which could be arrived via two different ways with different lengths. The situation is shown in Figure 2.10. In the beginning, the ants randomly decide for one of the two paths since the colony has no information obtained yet. While moving along the path, each ant emits pheromones, marking the path it has chosen. The ants that have decided for the shorter path arrive at the food source first and most likely take the same way back to the nest because until the other group also arrives at the food source, the pheromones are only on the shorter path.

When the ants that decided for the longer path finally arrive at the food source, they sense a higher concentration of pheromones on the shorter path than on the longer path because the shorter path is already traversed and marked twice. As more time passes and the ants go several rounds to the food source and back, the intensity difference between the pheromone values on the two ways further increases and the shorter path becomes more and more attractive. In the end, only very few ants use the longer way while the vast majority prefers the shorter path.




**(a)** At the beginning: Ants moving along one of the two paths.

**(b)** After some time: Most ants moving along the shorter path.

**Figure 2.10:** The double bridge experiment.

This capability of ants to find the shortest way to the food source has been modeled in order to create an optimization algorithm for the TSP ([DG97]). The resulting ant algorithm works as follows. Every edge $\{i, j\}$ of the input graph $G$ is assigned a *pheromone value* $\tau_{i,j}$, which is initialized with some positive value. The pheromone values represent the memory of the colony. High pheromone values indicate a high probability for the ants to select the respective edge again. Additionally, every edge $\{i, j\}$ has a certain *visibility* $\eta_{i,j}$, also called *heuristic information*, a value that determines how attractive an edge is for the ants, without taking pheromones into account. A typical choice is $\eta_{i,j} = 1/d_{i,j}$, where $d_{i,j}$ is the length of edge $\{i, j\}$.

Then, until some termination criterion holds, the following simulation of ants' behavior is repeated: The colony of $N$ ants is placed on the nodes of the graph, e. g., all the ants can be placed on the same node or a different node could be selected randomly for every ant. The variable $\pi_k$ describes the partial tour that ant $k$ has already traveled. At the beginning of the tour, $\pi_k$ contains only the starting node of ant $k$. As long as the tour is not complete,

every ant $k$ selects its next node randomly. If the current node of ant $k$ is node $i$, then it moves to node $j$ with probability $p_{i,j}^{(k)}$, defined as

$$p_{i,j}^{(k)} := \begin{cases} \frac{(\tau_{i,j})^{\alpha} \cdot (\eta_{i,j})^{\beta}}{\sum_{s \notin \pi_k} (\tau_{i,s})^{\alpha} \cdot (\eta_{i,s})^{\beta}} & \text{if } j \notin \pi_k \\ 0 & \text{otherwise,} \end{cases} \tag{2.12}$$

where $\alpha$ and $\beta$ are positive constants controlling the influence of $\tau$ and $\eta$. Typical choices are $\alpha = 1$ and $\beta = 2$ ([DG97]). When the ants have com-

**Algorithm 3:** Ant algorithm

**input** : complete graph $G = (V, E)$ with $n$ nodes and edge weights $d_{i,j}$ for $\{i, j\} \in E$
**output**: ordering $\pi$ of the nodes of $G$

```
for {i, j} ∈ E do
    Initialize τ_{i,j} ;            // initialize pheromone values
    Calculate η_{i,j} ;        // calculate heuristic information
repeat
    for k = 1 → N do
        Initialize π_k ;                  // initialize ants' tours
    for ℓ = 2 → n do
        for k = 1 → N do
            Set i := π_k(ℓ − 1);
            Choose j ∈ V with probability p_{i,j}^{(k)} as defined in Equation
            (2.12);
            Set π_k(ℓ) := j;
    Update τ according to Equation (2.13);
until Termination criterion holds;
return Best π_k;
```

pleted their tour, the pheromone values are updates via the equation

$$\tau_{i,j} := \rho \cdot \tau_{i,j} + \Delta\tau_{i,j}, \tag{2.13}$$

where $\rho \in (0, 1)$ is the pheromone decay parameter, describing how much of the pheromone vanishes during one tour of the ants. The value $\Delta\tau_{i,j}$ de-

scribes, by how much the pheromone value of edge $\{i, j\}$ is increased by ants moving over it. One example for a concrete choice of $\Delta\tau_{i,j}$ is ([DG97])

$$\Delta\tau_{i,j} = \sum_{k=1}^{N} \Delta\tau_{i,j}^{(k)}$$

with

$$\Delta\tau_{i,j}^{(k)} = \begin{cases} 1/L^{(k)} & \text{if edge } \{i, j\} \text{ is traversed by ant } k, \\ 0 & \text{otherwise,} \end{cases}$$

where $L^{(k)}$ is the length of ant $k$'s tour. When the termination criterion holds, the algorithm is stopped and the best tour visited by any ant is returned.

An algorithmic overview over the ant algorithm can be found in Algorithm 3. For an overview over variants of the ant algorithm for binary problems, including some theoretical runtime results, see [NSW09].

# 3. PSO as a Stochastic Process

The main goal of this chapter is to provide a reformulation of the classical particle swarm optimization (PSO) algorithm under a mathematical point of view by describing it as a real-valued *stochastic process*. Additionally, we discuss previous results in the light of this new model and formally introduce drift theory as a mathematical tool for obtaining runtime results.

In the first section, we briefly recall some mathematical basics of probability theory, with strong emphasize on the situation of the standard continuous black box optimization problem and differences to the discrete, combinatorial situation.

In the second section, we recall a strong negative result in the area of combinatorial optimization, namely the No Free Lunch (NFL) Theorem. The statement of the NFL Theorem is, roughly speaking, that on average over all possible combinatorial optimization problems, any two algorithms have the same performance. In particular, that means that without at least some knowledge about the objective function inside the black box, no method is better than just randomly guessing search points. However, it turns out that this result does not hold in the continuous setting.

In the third section, we introduce drift theory, an important tool for obtaining runtime results about evolutionary algorithms solving combinatorial optimization problems, which we adapt to the specific situation of PSO. Roughly speaking, if some process is on expectation decreasing by a certain amount, drift theory allows for bounds on the expected time until the process hits, e. g., the value $0$. Since most drift theorems from the literature more or less implicitly rely on the combinatorial structure of the search space, we prove a new drift theorem, which is explicitly designed to handle the continuous case.

Finally, we provide the model of the PSO algorithm in terms of the introduced mathematical concepts. In the light of this new formulation follows a discussion of some of the theoretical results already mentioned in Section 2.6 of Chapter 2. In particular, we revisit the negative results saying that the

expected time for optimization via PSO is infinite, and we point out why they are not in contradiction with the positive results of this thesis.

## 3.1 Basics of Probability Theory

In this section, we briefly recall some essential basics of probability theory, we highlight some subtleties and we point out the crucial differences between the discrete and the continuous situation. The concepts and terms introduced here can be found, e. g., in [Kle06] or [Bau96], usually in a much more general way than stated here.

### 3.1.1 Probability Space, Random Variables, Stochastic Processes and Conditional Expectation

Typically, a random experiment has some set $\Omega$ of possible outcomes, the so-called *sample space*. The cardinality and the type of the elements of $\Omega$ are not restricted. For instance, $\Omega = \{\text{HEAD}, \text{TAIL}\}$, $\Omega = \{1, 2, 3, 4, 5, 6\}$, $\Omega = \mathbb{N}$ and $\Omega = [0, 1]$, where $[a, b]$ denotes the real interval from $a$ to $b$, are possible. If $\Omega$ is at most countable, it is said to be *discrete*. If $\Omega$ is uncountable, it is called *continuous*.

If $\Omega$ is discrete, the "probability" is usually defined by assigning a probability value to every single outcome. However, in a typical continuous situation, every single element of $\Omega$ has probability $0$. So, instead of defining probability as a function $\Omega \to [0, 1]$, the probability is defined as a function that assigns a value to sets of outcomes, so-called *events*, rather than to the outcomes itself. The set $\mathcal{A} \subset \mathcal{P}(\Omega)$, where $\mathcal{P}(\Omega)$ denotes the power set of $\Omega$, defines which subsets of $\Omega$ are considered events. If $\Omega$ is discrete, usually the choice $\mathcal{A} := \mathcal{P}(\Omega)$ is made. However, as we will see in Section 3.1.2, in continuous situations this is in general not possible, i. e., there are subsets $A \subset \Omega$ that do not have a well-defined probability.

The set $\mathcal{A}$ includes $\Omega$ and $\emptyset$, i. e., the case that none of the outcomes actually occurs and the case that any of the outcomes occurs are both events. Additionally, for an event $A \in \mathcal{A}$, the case that $A$ does not happen, that means the set $\Omega \setminus A$, is also an event and therefore included in $\mathcal{A}$. Finally, for

any countable number of events $A_i \in \mathcal{A}$, the case that at least one of the $A_i$ happens, i. e., the set $\bigcup_i A_i$, is also considered an event. With the mentioned properties, $\mathcal{A}$ is called a *sigma-field* or *sigma-algebra*.

The function $P : \mathcal{A} \to [0, 1]$ is called *probability measure* or for short *probability*, if it assigns 1 to $\Omega$, i. e., the event that any of the outcome occurs has probability 1, and if additionally P is countably additive, i. e., if for every countable and mutual disjoint set of events $A_i$, the union of all the $A_i$ has probability $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$. Altogether, we have motivated the following definition of a probability space, consisting of the ground set $\Omega$, the sigma-field $\mathcal{A} \subset \mathcal{P}(\Omega)$ over $\Omega$ and the probability measure $P : \mathcal{A} \to [0, 1]$.

**Definition 3.1** (Probability Space)**.** A *probability space* is a triple $(\Omega, \mathcal{A}, P)$, where

- $\Omega$, the sample space, is an arbitrary, non-empty set,
- $\mathcal{A} \subset \mathcal{P}(\Omega)$, the set of events, is a *sigma-field* over $\Omega$, i. e.,
  - $\emptyset \in \mathcal{A}$,
  - $A \in \mathcal{A} \Rightarrow \Omega \setminus A \in \mathcal{A}$,
  - $A_1, A_2, \ldots \in \mathcal{A} \Rightarrow \bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$,
- P, the *probability measure*, is a function $\mathcal{A} \to [0, 1]$, such that
  - $P(\Omega) = 1$,
  - P is *sigma additive*, i. e., for every sequence $A_1, A_2, \ldots \in \mathcal{A}$ of mutually disjoint events, $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$

The elements of $\mathcal{A}$ are called *measurable with respect to* $\mathcal{A}$, $\mathcal{A}$*-measurable* or P*-measurable*. If P only satisfies the weaker condition "$P(A) \geq 0$ for every $A \in \mathcal{A}$" instead of "$P(A) \in [0, 1]$ for every $A \in \mathcal{A}$ and $P(\Omega) = 1$", then P is called a *measure* and $(\Omega, \mathcal{A}, P)$ a *measure space*. If $P(A) = 1$ for some event A, we say that A *holds almost surely*. If $P(A) = 0$, then A is called a *null set with respect to* P or a P*-null set*.

A very important example for a sigma-field over $\mathbb{R}^D$ is the so-called *Borel algebra* $\mathcal{B}(\mathbb{R}^D)$, which is defined as the smallest sigma-field containing every open subset $O \subset \mathbb{R}^D$, i. e., $\mathcal{B}(\mathbb{R}^D)$ is the intersection of all sigma-fields containing every open subset $O \in \mathbb{R}$. Similarly, $\mathcal{B}(\times_{d=1}^{D} [a_d, b_d])$ with $a_d, b_d \in \mathbb{R}$ and $a_d < b_d$ for every $d = 1, \ldots, D$ denotes the smallest sigma-field containing every open subset of the hyperrectangle $\times_{d=1}^{D} [a_d, b_d]$. More precisely:

$$\mathcal{B}\Big( \bigtimes_{d=1}^{D} [a_d, b_d] \Big) = \Big\{ B \cap \bigtimes_{d=1}^{D} [a_d, b_d] \;\Big|\; B \in \mathcal{B}(\mathbb{R}^D) \Big\}.$$

To complete the measure space over $\mathbb{R}^D$, the *Lebesgue measure* $\mathcal{L}^D$ is defined to serve as the uniform measure, i. e., it is defined as the unique measure over $\mathbb{R}^D$ that satisfies

$$\mathcal{L}^D\Big(\bigtimes_{d=1}^{D}[a_d, b_d]\Big) = \prod_{d=1}^{D}(b_d - a_d)$$

for every $a_1, \ldots, a_D, b_1, \ldots, b_D \in \mathbb{R}$ with $a_d \leq b_d$ for every $d = 1, \ldots, D$, i. e., that assigns to every hyperrectangle the product of its lengths multiplied over all dimensions. The proof of existence and uniqueness of $\mathcal{L}^D$ can be found in, e. g., [Kle06] or [Bau96]. Note that $\mathcal{L}^1$ restricted to $[0, 1]$ is the uniform probability distribution over $[0, 1]$ and, more general, $1/(b - a) \cdot \mathcal{L}^1$ restricted to $[a, b]$ is the uniform probability distribution over $[a, b]$.

The probability space $([0, 1]^D, \mathcal{B}([0, 1]^D), \mathcal{L}^D)$ is suitable to model a random experiment in which D values are uniformly and independently chosen from the interval $[0, 1]$, i. e., the well-known function "rand()" is called D times. However, in the classical PSO, it is not a priori clear how often the function "rand()" is called. Therefore, it is necessary to construct a probability space over $\Omega = [0, 1]^\infty$, the space of all sequences $(\omega_i)_{i\in\mathbb{N}}$ with $\omega_i \in [0, 1]$ for every $t \in \mathbb{N}$, which allows drawing an infinite number of values, chosen uniformly and independently from the interval $[0, 1]$. The appropriate sigma-field is the product sigma-field of infinitely many instances of the sigma-field $\mathcal{B}([0, 1])$. Formally, a product sigma-field of sigma-fields $\mathcal{A}_i$ with $i \in I$, where I is some set of indices, is defined as follows.

**Definition 3.2.** Let $I = (i_1, i_2, \ldots)$ be a set of indices and let be $\Omega_i = [a_i, b_i]$ with $a_i < b_i$ and $\mathcal{A}_i = \mathcal{B}(\Omega_i)$ for every $i \in I$. The product sigma-field of the $\mathcal{A}_i$

$$\bigotimes_{i\in I}\mathcal{A}_i$$

is defined as the smallest sigma-field containing

$$\{\Omega_{i_1} \times \ldots \times \Omega_{i_{k-1}} \times B \times \Omega_{i_{k+1}} \times \ldots \;\big|\; k \leq |I|, B \in \mathcal{A}_{i_k}\}.$$

Note that as a property of the Borel algebra, we have for finite I

$$\bigotimes_{i\in I}\mathcal{B}([a_i, b_i]) = \mathcal{B}\Big(\bigtimes_{i\in I}[a_i, b_i]\Big).$$

To complete the construction of the probability space for calling "rand()" infinitely often, we need to define a probability measure

$$\mathcal{L}^\infty : \bigotimes_{i\in\mathbb{N}_0} \mathcal{B}([0,1]) \to [0,1].$$

In order to construct $\mathcal{L}^\infty$, we use a corollary to the Theorem of Ionescu-Tulcea, which states that such a probability measure exists and is uniquely defined.

**Theorem 3.1** (Corollary to the Theorem of Ionescu-Tulcea, [Kle06])**.** For every $i \in \mathbb{N}$, let $(\Omega_i, \mathcal{A}_i, P_i)$ be a probability space. Then, there is a uniquely determined probability measure P on

$$\Big(\bigtimes_{i=0}^{\infty} \Omega_i, \bigotimes_{i=0}^{\infty} \mathcal{A}_i\Big),$$

such that

$$P\Big(A_0 \times \ldots \times A_n \times \bigtimes_{i=n+1}^{\infty} \Omega_i\Big) = \prod_{i=0}^{n} P_i(A_i)$$

for every $n \in \mathbb{N}$ and every $A_i \in \mathcal{A}_i$ for $i = 0, \ldots, n$.

In the following definition, we summarize the resulting probability space, which we will use as the underlying probability space of the particle swarm.

**Definition 3.3** (PSO probability space)**.** The probability space $R = (\Omega, \mathcal{A}, P)$ is defined via

- $\Omega := [0,1]^\infty$,
- $\mathcal{A} := \bigotimes_{\mathbb{N}} \mathcal{B}([0,1])$,
- $P := L^\infty$,

where $[0,1]^\infty = \{(\omega_0, \omega_1, \ldots) \mid \omega_i \in [0,1] \text{ for every } i \in \mathbb{N}\}$ is the space of all sequences with values in $[0,1]$, $\bigotimes_{\mathbb{N}} \mathcal{B}([0,1])$ is the product sigma-field of countably many instances of $\mathcal{B}([0,1])$ and $L^\infty$ is obtained from Theorem 3.1 by setting $(\Omega_i, \mathcal{A}_i, P_i) = ([0,1], \mathcal{B}([0,1]), \mathcal{L}^1)$.

In most cases, not the outcomes of a random experiment itself are interesting, but some random observations which depend on the outcome. In order to formally describe such random observations, we define the concept of random variables.

**Definition 3.4** (Random Variable)**.** Let $(\Omega, \mathcal{A}, P)$ be a probability space. A function $X : \Omega \to \mathbb{R}^D$ is called a *random variable*, if $X^{-1}(B) \in \mathcal{A}$ for every $B \in \mathcal{B}(\mathbb{R}^D)$. In that case, we also say that $X$ is *measurable with respect to* $\mathcal{A}$, *$\mathcal{A}$-measurable* or *P-measurable*. If $X$ is a random variable, then $(\mathbb{R}^D, \mathcal{B}(\mathbb{R}^D), P \circ X^{-1})$ is a probability space and $P \circ X^{-1}$ is called the *distribution* of $X$. Additionally, $\sigma(X) := X^{-1}(\mathcal{B}(\mathbb{R}^D))$ is called the *sigma-field generated by* $X$.

**Example 3.1** (Indicator Variable)**.** The *indicator variable* of an event $A \in \mathcal{A}$ is defined as

$$\mathbb{1}_A(\omega) := \begin{cases} 1, & \text{if } \omega \in A, \\ 0, & \text{otherwise.} \end{cases}$$

This is a random variable, because the preimage of a set $B \in \mathcal{B}(\mathbb{R}^D)$ under $\mathbb{1}_A$ is

$$\mathbb{1}_A^{-1}(B) = \begin{cases} \emptyset, & \text{if } 0, 1 \notin B, \\ A, & \text{if } 0 \notin B, 1 \in B, \\ \Omega \setminus A, & \text{if } 0 \in B, 1 \notin B, \\ \Omega, & \text{if } 0, 1 \in B. \end{cases}$$

The sets $\emptyset$ and $\Omega$ are included in $\mathcal{A}$ by definition and since $A \in \mathcal{A}$, $\mathcal{A}$ also contains the complement $\Omega \setminus A$.

In order to simplify notation, we let $X \in B$ denote the event $X^{-1}(B) = \{\omega \in \Omega \mid X(\omega) \in B\}$ for a random variable $X$ and a $B \in \mathcal{B}(\mathbb{R}^D)$. Events like $X < a$ and $X > b$ are defined analogously.

Instead of describing only a single random experiment, many randomized systems evolve over time and perform some random state transition at every time step $t$. In order to describe systems with such a time-dependent behavior, we use a sequence of random variables. That leads to the following definition of a stochastic process.

**Definition 3.5** (Stochastic Process, Filtration)**.** A sequence of random variables $X = (X_0, X_1, \ldots)$ is called a *stochastic process*. A sequence of sigma-fields $\mathbb{F} = (\mathcal{F}_0, \mathcal{F}_1, \ldots)$ is called a *filtration*, if $\mathcal{F}_t \subset \mathcal{F}_{t+1}$ for every $t \in \mathbb{N}_0$. If $X_t$ is measurable with respect to $\mathcal{F}_t$, i. e., if for every $t \in \mathbb{N}_0$ and every $B \in \mathcal{B}(\mathbb{R}^D)$ we have that $X_t^{-1}(B) \in \mathcal{F}_t$, then $X$ is called *adapted* to $\mathbb{F}$.

Intuitively, we can think of $\mathcal{F}_t$ as the mathematical object that carries the information about the stochastic process that is determined at time $t$ and

characterize events measurable with respect to $\mathcal{F}_t$ as the events from which we know at time $t$ if they happen or not. The following example illustrates the use of knowledge about the past.

**Example 3.2.** Consider the probability space $R$ from Definition 3.3. For $\omega = (\omega_0, \omega_1, \ldots) \in [0,1]^\infty$ and let the stochastic process $X = (X_0, X_1, \ldots)$ be defined as

$$X_t(\omega) := \sum_{t'=1}^{t} \omega_{t'} \text{ for every } t \in \mathbb{N}_0.$$

The sigma-field $\mathcal{F}_t$ generated by $X_t$ is the union of all sets of the form $B \times [0,1]^\infty$ where $B \in \mathcal{B}([0,1]^t)$ and $X$ is adapted to the sequence of sigma-fields $\mathbb{F} = (\mathcal{F}_t)_{t\in\mathbb{N}}$. That means an element of $\mathcal{F}_t$ contains information about the first $t$ entries of $\omega$, namely that $(\omega_0, \ldots, \omega_t) \in B$, but for $t' > t$, it only says $\omega_{t'} \in [0,1]$.

In the following, we use information about the past to make predictions about the future of a stochastic process, i. e., we want to take the perspective of some time $t$ and make predictions (in terms of, e. g., the expectation value) about the stochastic process at time $s$, where $s > t$, by taking the information at time $t$ into account. For formally stating such predictions, we define the concepts of *conditional probability* and *conditional expectation*.

The standard definitions of the conditional probability for some event $A$, given some other event $B$ as

$$P(A \mid B) := \frac{P(A \cap B)}{P(B)} \tag{3.1}$$

and the conditional expectation of some random variable $X$ given $B$ as

$$E[X \mid B] := \frac{E[X \cdot \mathbb{1}_B]}{P(B)} \tag{3.2}$$

work fine for discrete situations, but for handling random variables over a continuous search space, it is necessary to generalize these concepts. The following example illustrates this.

Consider the stochastic process $X = (X_0, X_1, \ldots)$ from Example 3.2, where $X_0$ is uniformly distributed over $[0,1]$ and knowing the value $x_0$ of $X_0$, $X_1$ is uniformly distributed over $[x_0, x_0 + 1]$. Intuitively, the expectation of $X_1$, knowing that $X_0 = x_0$, should be $x_0 + 1/2$, but the discrete definition of conditional expectation (Equation (3.2)) cannot be applied because for every $x_0$, the "B", i. e., the event $X_0 = x_0$, has probability 0. The solution of

the stated issue is to define the conditional expectation itself as a random variable. In the present example, we would write $E[X_1 \mid \mathcal{F}_0] = X_0 + 1/2$, a random variable measurable with respect to $\mathcal{F}_0$. Formally, the definition of the conditional expectation is generalized in the following way.

**Definition 3.6** (Conditional Expectation)**.** For a probability space $(\Omega, \mathcal{A}, P)$, let $\mathcal{F} \subset \mathcal{A}$ be a sigma-field and $X$ a random variable. Then $E[X \mid \mathcal{F}]$ denotes a random variable that fulfills the following two conditions.

- $E[X \mid \mathcal{F}]$ is measurable with respect to $\mathcal{F}$,
- $E[X\mathbb{1}_A] = E[E[X \mid \mathcal{F}]\mathbb{1}_A]$ for every $A \in \mathcal{F}$.

The random variable $E[X \mid \mathcal{F}]$ is called the *conditional expectation of* $X$ *given* $\mathcal{F}$. Additionally, for an event $B \in \mathcal{A}$, the *conditional probability of* $B$ *given* $\mathcal{F}$ is defined as

$$P(B \mid \mathcal{F}) := E[\mathbb{1}_B \mid \mathcal{F}].$$

We can easily verify that in the previous example, indeed $E[X_1 \mid \mathcal{F}_0] = X_0 + 1/2$ holds.

**Example 3.3.** Consider the stochastic process $X$ and the filtration $\mathbb{F}$ as defined in Example 3.2. The candidate for the conditioned expectation $E[X_1 \mid \mathcal{F}_0]$ of $X_1$ given $\mathcal{F}_0$ is $Y := X_0 + 1/2 = \omega_0 + 1/2$. Since $Y$ depends only on $\omega_0$, it is measurable with respect to $\mathcal{F}_0$. For every $A \in \mathcal{F}_0$, i. e., every $A \subset [0,1]^\infty$ of the form $B \times [0,1]^\infty$ with $B \in \mathcal{B}([0,1])$, we have

$$\begin{aligned}
E[X_1\mathbb{1}_A] - E[Y\mathbb{1}_A] &= E[(X_1 - Y)\mathbb{1}_A] \\
&= \int_A X_1(\omega) - Y(\omega)\ dP(\omega) \\
&= \int_{B\times[0,1]^\infty} X_1(\omega) - Y(\omega)\ dP(\omega) \\
&= \int_{B\times[0,1]} (\omega_0 + \omega_1) - (\omega_0 + 1/2)\ d(\omega_0, \omega_1) \\
&= \int_{B\times[0,1]} \omega_1 - 1/2\ d(\omega_0, \omega_1) \\
&= P(B) \cdot \int_{[0,1]} \omega_1 - 1/2\ d(\omega_1) = 0.
\end{aligned}$$

That means that for every $A \subset [0,1]^\infty$, $E[X_1\mathbb{1}_A] = E[Y\mathbb{1}_A]$, which verifies the second condition of Definition 3.6, so $Y = E[X_1 \mid \mathcal{F}_1]$ actually holds.

As it is proved in, e. g., [Kle06] and [Bau96], the conditional expectation exists and is uniquely determined up to null sets, i. e., the probability that two random variables which both fulfill the conditions of Definition 3.6 differ is 0. We will use the conditional expectation to formally model the *drift* of a stochastic process, i. e., we formalize a statement like "on expectation, the process X decreases by at least $\varepsilon$ in every time step" as

$$\forall t \in \mathbb{N}_0 : \mathrm{E}[X_{t+1} \mid \mathcal{F}_t] \leq X_t - \varepsilon.$$

Sometimes, the notation is slightly abused to state that a certain property of $\mathrm{E}[X_{t+1} \mid \mathcal{F}_t]$ only holds under some condition. In general, for a sigma-field $\mathcal{F} \subset \mathcal{A}$, an $\mathcal{A}$-measurable random variable X, an $\mathcal{F}$-measurable random variables Y and an event $C \in \mathcal{F}$, we write

$$\mathrm{E}[X \mid \mathcal{F}, C] \leq Y :\Leftrightarrow \mathbb{1}_C \cdot \mathrm{E}[X \mid \mathcal{F}] \leq \mathbb{1}_C \cdot Y.$$

This can be read as "if C occurs, then $\mathrm{E}[X \mid \mathcal{F}] \leq Y$." Note that $C \in \mathcal{F}$ means that the information provided by $\mathcal{F}$ is already sufficient to determine if C occurs, i. e., if the actual outcome $\omega \in \Omega$ is included in C. E. g., the statement "on expectation, the process X decreases by at least $\varepsilon$ in every time step as long as X is positive" can be expressed as

$$\forall t \in \mathbb{N} : \mathrm{E}[X_{t+1} \mid \mathcal{F}_t, X_t > 0] \leq X_t - \varepsilon.$$

An important structural property of stochastic processes is the so-called *Markov property*, which describes a process without a memory of its past. A stochastic process has the Markov property, if predictions about its future, taking information about its past and its present state into account are the equal to the predictions using only the information about the current state. The following definition provides a formal description of this property.

**Definition 3.7** (Markov Property)**.** A stochastic process $(X_t)_{t \in \mathbb{N}_0}$ has the Markov property, if

$$\mathrm{P}(X_t \in A \mid \mathcal{F}_s) = \mathrm{P}(X_t \in A \mid \sigma(X_s))$$

for every $A \in \mathbb{R}^D$ and every $s, t \in \mathbb{N}_0$ with $s \leq t$. Here, $\sigma(X_s)$ is the sigma-field generated by $X_s$ as defined in Definition 3.4.

### 3.1.2 Measurability

The existence of non-measurable functions has important implications for analyzing any randomized, continuous optimization method. For any meaningful statement about an optimization scenario, $\mathcal{B}(\mathbb{R}^D)$-measurability of the objective function is a minimum requirement. To illustrate this, we construct an example for a non-measurable function, namely the indicator function $\mathbb{1}_V$ of a set $V \subset [0,1]$ that is itself not measurable with respect to $\mathcal{B}(\mathbb{R})$, i. e., $V \notin \mathcal{B}(\mathbb{R})$. Since sets that are defined constructively are usually $\mathcal{B}(\mathbb{R})$-measurable, e. g., any countable union of intervals is $\mathcal{B}(\mathbb{R})$-measurable, the construction of $V$ is more complex. Consider the equivalence relation $R$ on $[0,1] \times [0,1]$, defined via

$$r \approx_R s :\Leftrightarrow r - s \in \mathbb{Q},$$

i. e., two real numbers $r, s \in [0,1]$ are considered equivalent, if their difference is rational. Then, the equivalence class of some $r \in [0,1]$ is

$$[r] = (r + \mathbb{Q}) \cap [0,1] := \{r + q \mid q \in \mathbb{Q}\} \cap [0,1].$$

Since $\mathbb{Q}$ is countable, every equivalence class contains a countable number of elements and since the interval $[0,1]$ is uncountable, there are uncountably many equivalence classes. The set $V$ is defined as any set that contains exactly one representative of every equivalence class. In the literature (e. g., [Kle06]), $V$ is called *Vitali Set*. Since every real number $r \in [0,1]$ is equivalent to some element $v \in V$ and since $-1 \leq r - v \leq 1$, it follows that

$$[0,1] \subset V + \mathbb{Q} \cap [-1,1] := \{v + q \mid v \in V, q \in \mathbb{Q} \cap [-1,1]\}.$$

Additionally, since $V \subset [0,1]$, we have that

$$V + \mathbb{Q} \cap [-1,1] \in [-1,2].$$

Defining

$$V_q := V + q = \{v + q \mid v \in V\}$$

for some $q \in \mathbb{Q} \cap [-1,1]$ as the set $V$, shifted by $q$, we can rewrite $V + \mathbb{Q} \cap [-1,1]$ as

$$V + \mathbb{Q} \cap [-1,1] = \bigcup_{q \in \mathbb{Q} \cap [-1,1]} V_q.$$

Altogether, it follows that

$$[0,1] \subset \bigcup_{q \in \mathbb{Q} \cap [-1,1]} V_q \subset [-1,2], \tag{3.3}$$

i. e., the countable union of all the $V_q$ covers $[0,1]$ and is contained in $[-1,2]$. Furthermore, it is easy to see that the $V_q$ are disjoint. Assume, for contradiction, that $v \in V_{q_1} \cap V_{q_2}$ for some $q_1 \neq q_2$. Then, we would have $V \ni v - q_1 \neq v - q_2 \in V$, which contradics the definition of $V$ since $v - q_1 \approx_R v - q_2$.

Keeping these properties in mind, the hypothetical optimization problem is defined as follows:

$$\text{Maximize } f(x) = \mathbb{1}_V(x) \text{ over the search space } S = [-1,2].$$

A search heuristic that maximizes $f$ is successful as soon as it finds a point $v \in V$. Assuming that the first step consists of uniformly sampling a point of the search space, the success probability would be the probability of $V$ under uniform distribution. Let $P$ be the respective probability measure, defined as $P(A) := 1/3 \cdot \mathcal{L}(A \cap [-1,2])$. Then, from (3.3), it follows that

$$\frac{1}{3} = P([0,1]) \leq P\Big( \bigcup_{q \in \mathbb{Q} \cap [0,1]} V_q \Big) \leq 1.$$

Furthermore, the sigma-additivity of probability measures and the translation invariance of the uniform distribution imply

$$P\Big( \bigcup_{q \in \mathbb{Q} \cap [0,1]} V_q \Big) = \sum_{q \in \mathbb{Q} \cap [0,1]} P(V_q) = \sum_{q \in \mathbb{Q} \cap [0,1]} P(V).$$

This results in

$$\frac{1}{3} \leq \sum_{q \in \mathbb{Q} \cap [0,1]} P(V) \leq 1,$$

a contradiction because either $P(V) = 0$, then $\sum_{q \in \mathbb{Q} \cap [0,1]} P(V) = 0 < 1/3$ or $P(V) > 0$, then $\sum_{q \in \mathbb{Q} \cap [0,1]} P(V) = \infty > 1$. This illustrates that there is no meaningful way to define the probability of the constructed set $V$ under uniform distribution and therefore the success probability for optimizing $f$ is undefined from the first step on. Similar investigations show that all other interesting characteristics like the expected optimization time are undefined

as well. So, measurability of the objective function is a minimum requirement for making any valid statements about the quality of some randomized optimization method. Although the non-measurable example function looks very artificial, in the following section we will point out that measurable functions have some structural properties that have significant consequences for the continuous black box optimization problem.

## 3.2 (No) Free Lunch and Lebesgue's Differentiation Theorem

A fundamental result in the area of combinatorial black box optimization is the so-called No Free Lunch Theorem ([WM97]). Basically, it states that in presence of a finite search space, no deterministic or randomized search method is superior to any other with respect to the number of objective function evaluations. Therefore, in absence of problem specific knowledge, there is no algorithm that works better than, e. g., randomly sampling points of the search space. The formal statement and the proof can be found in ([WM97]). The key idea of the proof is to assume that the objective function $f$ is chosen randomly from the space of "all" possible objective functions, before the optimization algorithm is applied. Then, for any permutation $\pi$ of the search space, $f \circ \pi$ is also a valid objective function. Therefore, with the $k$ pairs of search points and their function values $(x_1, f(x_1)), \ldots, (x_k, f(x_k))$, the distribution of the function value $f(x)$, given $(x_1, f(x_1)), \ldots, (x_k, f(x_k))$, with some search point $x$ not already sampled, does not depend on the particular choice of $x$. In other words: For every subset $A$ of the image of $f$ and every $x, \tilde{x}$ with $x \neq \tilde{x}$ and $x, \tilde{x} \notin \{x_1, \ldots, x_k\}$, we have $P(f(x) \in A) = P(f(\tilde{x}) \in A)$. Therefore, the information about the values of $f$ at the search points $x_1, \ldots, x_k$ does not imply any preference on the remaining search points.

It is remarkable that in the continuous case, i. e., when the search space is uncountable, the situation is completely different ([AT10]). That is because the set of $\mathcal{B}^D$-measurable functions has a structural property that is known as Lebesgue's Differentiation Theorem.

**Theorem 3.2** (Lebesgue's Differentiation Theorem)**.** For $z \in \mathbb{R}^D$ and $r > 0$, $B_r(z)$ denotes the ball with center $z$ and radius $r$. Let $f$ be a $\mathcal{B}^D$-measurable

function, such that $\int_{\mathbb{R}^D} f(x)dx$ exists. Then for almost every $x \in \mathbb{R}^D$, the limit

$$\lim_{r \to 0} \frac{1}{|B_r(z)|} \int_{B_r(z)} |f(x) - f(z)|dx$$

exists and is equal to 0.

As a stochastic interpretation, we can think about the random experiment that uniformly samples points with a distance of at most $r$ from some fixed point $z$. Then the theorem says that for almost every point $z$, as $r$ is decreased towards 0, the value obtained will approach $f(z)$ almost surely. This means that $f$ behaves very similar to a continuous function. As pointed out in [AT10], this is already enough structural information to prove that the No Free Lunch Theorem can not be generalized to the continuous case. Therefore, there is indeed hope to find optimization methods that are superior to the blind search method of randomly sampling search points.

## 3.3 Drift Theory

Drift Theory has proved its value as an important tool for analyzing evolutionary algorithms for combinatorial optimization problems. Examples for successful application are, among others, the optimization time of a $(1+1)$-EA for optimizing linear functions over $\{0,1\}^D$ ([HY04, Jäg08, DJW12]), calculating minimum spanning trees ([NW07, DG10]), solving the single source shortest path problem ([BBD+09, DJ10, DG10]) and finding a Eulerian cycle ([DG10]). Originally, He and Yao ([HY01]) introduced drift theory to the field of evolutionary algorithms by using mathematical results of Hajek ([Haj82]). The goal of drift analysis is to provide sufficient conditions to bound the time $\tau := \inf\{t \geq 0 \mid X_t \leq 0\}$ until some stochastic process $X$ reaches 0. This is straight-forward if the process is non-increasing and has a bounded minimum decrease, but if the process decreases only on expectation, while it might with some positive probability increase as well, things get more complicated.

### 3.3.1 Classical Drift Theory

Consider a stochastic process $X = (X_t)_{t\in\mathbb{N}}$, adapted to a filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in\mathbb{N}}$, such that one of the following *drift conditions* holds.

$$E[X_{t+1} - X_t \mid X_t > 0] \leq -\delta, \tag{3.4}$$

or

$$E[X_{t+1} - X_t \mid \mathcal{F}_t, X_t > 0] \leq -\delta \tag{3.5}$$

for some $\delta > 0$. Condition (3.4) states that under the condition that $X_t > 0$, the process will decrease by $\delta$, i. e., this is the expectation we have at time 0. Stronger than this, Condition (3.5) states that for every state of the stochastic process $X$ at time $t$, the uniform bound of $-\delta$ holds for the conditional expectation of $X_{t+1} - X_t$. Although Condition (3.5) is formally a stronger assumption than Condition (3.4), it is usually easier to handle and to verify.

If such a drift condition holds, i. e., if we can expect the process $X$ to decrease by at least $\delta$ at each time step, one would intuitively assume that after at most $E[X_0]/\delta$ iterations, the process reaches 0, i. e., the expectation $E[\tau]$ of the time $\tau := \inf\{t \in \mathbb{N} \mid X_t \leq 0\}$ for $X_t$ becoming $\leq 0$ should be bounded from above by $E[X_0]/\delta$. However, in general there is a difference between the time $T := \inf\{t \in \mathbb{N} \mid E[X_t] \leq 0\}$, the deterministic time when the expectation of the stochastic process reaches 0, and the expectation of the random time $\tau$. The following example illustrates this.

**Example 3.4.** Consider the following stochastic process.

$$X_0 := \begin{cases} 2, & \text{with probability } \frac{1}{2}, \\ 4, & \text{with probability } \frac{1}{2}, \end{cases} \qquad X_{t+1} := \begin{cases} 2, & \text{if } X_0 = 2, \\ X_t - 2, & \text{otherwise.} \end{cases}$$

If $X_0 = 2$, which happens with probability $1/2$, then the process will stay 2 forever and $\tau$ is infinite. Otherwise, $X_0 = 2$ and $X_t = 4 - 2 \cdot t$. In that case, $\tau = 2$. Altogether, $\tau$ has an infinite expectation. On the other hand, we have that $E[X_t] = 3 - t$, i. e., $T = 3$.

Typical drift theorems say that under the drift condition (Condition (3.4) or Condition (3.5)) and some additional assumptions, such pathological cases can be avoided and the bound $E[\tau] \leq E[X_0]/\delta$ actually holds. Additionally, by replacing "$\leq$" with "$\geq$" in Condition (3.4) and Condition (3.5), a lower

bound on the drift is achieved and drift theory leads to a lower bound of $E[X_0]/\delta$ for $\tau$.

Depending on the nature of the algorithm which is analyzed and on the underlying stochastic process, sometimes the drift conditions are expressed multiplicatively, i. e., instead of expecting an absolute decrease of $X$ during every iteration, we assume that $X$ decreases on expectation by a constant factor $1-\delta$ ([DJW12]). Then, the drift condition has the form

$$E[X_{t+1} - X_t \mid \mathcal{F}_t, X_t > 1] \leq -\delta \cdot X_t \tag{3.6}$$

and drift theory yields, again under some additional assumptions, the upper bound $E[\tau \mid \mathcal{F}_t] \leq \ln(X_0)/\delta$. Note that in case of a multiplicative drift theorem, in contrast to the additive case, the modified drift condition, where "$\leq$" is replaced by "$\geq$", does not imply any lower bound on $E[\tau]$.

There are many slightly different drift theorems in the literature, varying in the assumptions which are made in addition to the drift condition. Typical additional assumptions are chosen from the following collection:

- $X_t$ has a finite image ([HY04, DJW12]),
- $X_t \notin (0,1)$ for every $t \in \mathbb{N}$ ([DG10]),
- $X_0 \leq b$ for some constant $b \in \mathbb{R}$ ([HY01]),
- $X_t \geq 0$ for every $t \in \mathbb{N}$ ([HY01, DG10]),
- $|X_{t+1} - X_t| \leq c$ for a constant $c \in \mathbb{R}^+$ and every $t \in \mathbb{N}$ ([Jäg07, Jäg08]),

In a discrete scenario, all of these assumptions are trivial or at least very easy to fulfill. However, for calculating runtime bounds for PSO, such assumptions become crucial. In the following section, we provide a new variant of the drift theorem, suitable for the continuous situation and in particular for obtaining runtime results about the optimization time of PSO.

### 3.3.2 Drift Theory for Continuous Search Spaces

When analyzing the drift of some stochastic process, the most important difference between the discrete and the continuous situation is that only in the continuous scenario, the state of the algorithm can be outside but arbitrary close to the region considered optimal. If a process $X = (X_t)_{t\in\mathbb{N}}$, adapted

to a filtration $\mathbb{F} = (\mathcal{F}_t)_{t\in\mathbb{N}}$, is continuous but non-negative, a condition like Condition (3.5) from the previous section is unsatisfiable. That is because for non-negative image of the $X_t$, we have

$$E[X_{t+1} - X_t \mid \mathcal{F}_t, X_t > 0] \geq E[-X_t \mid \mathcal{F}_t, X_t > 0] = -X_t,$$

which can be arbitrary close to 0 for small values of $X_t$. Therefore, we cannot expect Condition (3.5) to hold with any constant $\delta$. To solve this issue, we develop a modified drift theorem, which allows for $X$ to have negative values. If the process $X$ serves as a measure for some optimization algorithm, that means that the elements of the optimized region do not all have the same value 0 as in the discrete setting. Instead, they can have any value $\leq 0$. However, as the following example illustrates, the drift condition alone does not imply any upper bound on the expected time for $X$ to hit 0.

**Example 3.5.** For the constants $x_0 > 0$, $\delta > 0$, we consider the following stochastic process $X = (X_0, X_1, \ldots)$.

$$X_0 := x_0, X_t := \begin{cases} X_{t-1}, & \text{w. p. } 1 - 1/3^t \\ X_{t-1} - 3^t \cdot \delta, & \text{w. p. } 1/3^t, \end{cases} \quad \text{for } t > 0.$$

This process fulfilles Condition 3.5:

$$E[X_{t+1} - X_t \mid \mathcal{F}_t, X_t > 0] = (1 - 1/3^t) \cdot 0 + 1/3^t \cdot (-3^t \cdot \delta) = -\delta.$$

However, to actually arrive at a value $\leq 0$, the second case of the definition of $X_t$ must happen at least once. Therefore, for the probability for $X$ to reach a value $\leq 0$, we have

$$P(\exists t : X_t \leq 0) \leq \sum_{t=1}^{\infty} \frac{1}{3^t} = \frac{1}{2},$$

i. e., the time $\tau$ until $X$ passes 0 is with probability at least $1/2$ infinite and has therefore infinite expectation.

The reason, why in Example 3.5 no meaningful bound on the first hitting time of $X$ hitting $\mathbb{R}_{\leq 0}$ could be achieved, is that the expected drift results in a process that with very little probability decreases dramatically, while it stays constant with overwhelming probability. Therefore, additionally to the drift condition, we need an assumption for our drift theorem that forbids such pathological cases.

As a possible solution, we ask for a drift $\leq -\delta$ even for the modified process that decreases at most by a constant $\Delta$, i. e., we want $\max\{X_{t+1} - X_t, -\Delta\}$ to yield an expected drift $\leq -\delta$. As it turns out, this modification allows us for an upper bound only slightly larger than with the prerequisite of a non-negative process.

The following result is a modified version of the additive drift theorem presented in [HY01].

**Theorem 3.3** (Additive Drift Theorem)**.** Let $\Delta > \delta > 0$ be some constants. Let $X = (X_t)_{t\in\mathbb{N}}$ be a stochastic process adapted to a filtration $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}$, such that

$$E[\max\{X_{t+1} - X_t, -\Delta\} \mid \mathcal{F}_t, X_t > 0] \leq -\delta, \tag{3.7}$$

Let $\tau := \min\{t \geq 0 \mid X_t \leq 0\}$ be the first index when $X_t$ passes 0. Then

$$E[\tau \mid \mathcal{F}_0] \leq \mathbb{1}_{\{X_0>0\}} \cdot (X_0 + \Delta)/\delta.$$

For the proof of Theorem 3.3, we want to apply the Lemma of Fatou, a convergence result from the area of stochastic processes.

**Lemma 3.1** (of Fatou ([Kle06, Bau96]))**.** Let $Y = (Y_0, Y_1, \ldots)$ be a stochastic process, such that $Y_t \geq 0$ almost surely for every $t \in \mathbb{N}$. Then,

$$\liminf_{t\to\infty} E[Y_t] \geq E[\liminf_{t\to\infty} Y_t].$$

*Proof (of Theorem 3.3).* Since $X_0 \leq 0$ implies $\tau = 0$, we can without loss of generality assume that $X_0 > 0$ and therefore $\tau \geq 1$ almost surely. We then define the stochastic process $Y = (Y_0, Y_1, \ldots)$ via

$$Y_t := \max\{X_{\min\{t,\tau\}} + \Delta, 0\} + \delta \cdot \min\{t, \tau\}.$$

Note that every $Y_t$ is non-negative and measurable with respect to $\mathcal{F}_t$. Additionally, we have

$$\begin{aligned}
\mathrm{E}[Y_{t+1} \mid \mathcal{F}_t] &= \mathrm{E}\left[\mathbb{1}_{\{\tau>t\}} \cdot Y_{t+1} \mid \mathcal{F}_t\right] + \mathrm{E}\left[\mathbb{1}_{\{\tau\leq t\}} \cdot Y_{t+1} \mid \mathcal{F}_t\right] \\
&= \mathrm{E}\left[\mathbb{1}_{\{\tau>t\}} \cdot (\max\{X_{t+1} + \Delta, 0\} + \delta \cdot (t+1)) \mid \mathcal{F}_t\right] \\
&\quad + \mathbb{1}_{\{\tau\leq t\}} \cdot (\max\{X_\tau + \Delta, 0\} + \delta \cdot \tau) \\
&\leq \mathbb{1}_{\{\tau>t\}} \cdot \mathrm{E}\left[\max\{X_{t+1} - X_t, -\Delta\} + X_t + \Delta + \delta \cdot (t+1) \mid \mathcal{F}_t\right] \\
&\quad + \mathbb{1}_{\{\tau\leq t\}} \cdot Y_t \\
&\overset{(3.7)}{\leq} \mathbb{1}_{\{\tau>t\}} \cdot (-\delta + X_t + \Delta + \delta \cdot (t+1)) + \mathbb{1}_{\{\tau\leq t\}} \cdot Y_t \\
&= \mathbb{1}_{\{\tau>t\}} \cdot (\max\{X_t + \Delta, 0\} + \delta \cdot t) + \mathbb{1}_{\{\tau\leq t\}} \cdot Y_t \\
&= \mathbb{1}_{\{\tau>t\}} \cdot Y_t + \mathbb{1}_{\{\tau\leq t\}} \cdot Y_t = Y_t,
\end{aligned}$$

i. e., $Y_t \geq \mathrm{E}[Y_{t+1} \mid \mathcal{F}_t]$. By induction and from $X_0 + \Delta = Y_0$, it follows that $X_0 + \Delta \geq \mathrm{E}[Y_t \mid \mathcal{F}_0]$ for every $t$ and therefore

$$\begin{aligned}
X_0 + \Delta &\geq \liminf_{k\to\infty} \mathrm{E}[Y_t \mid \mathcal{F}_0] \overset{(*)}{\geq} \mathrm{E}[\liminf_{t\to\infty} Y_t \mid \mathcal{F}_0] \\
&= \mathrm{E}[\underbrace{\liminf_{t\to\infty} (\max\{X_{\min\{t,\tau\}}, 0\})}_{\geq 0} + \delta \cdot \tau \mid \mathcal{F}_0] \\
&\geq \delta \cdot \mathrm{E}[\tau \mid \mathcal{F}_0].
\end{aligned}$$

The inequality $(*)$ is obtained from Lemma 3.1. This finishes the proof. □

Originally, the authors of [HY01] used Lebesgue's Theorem of Dominated Convergence ([Kle06, Bau96]) instead of Lemma 3.1. Therefore, they needed the additional assumption that every $X_t$ is bounded from above by a constant. While this is obviously true if $X_t$ has a finite image, this requirement cannot be fulfilled in the analysis of PSO.

Next, the drift theorem is transferred from an additive situation to a multiplicative one. The following theorem is basically the multiplicative drift theorem from [DJW12] with just slightly changed preconditions.

**Theorem 3.4** (Multiplicative Drift Theorem)**.** Let $X = (X_t)_{t\in\mathbb{N}}$ be a strictly positive stochastic process adapted to a filtration $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}$, such that

$$\mathrm{E}[\max\{X_{t+1}/X_t, e^{-\Delta}\} \mid \mathcal{F}_t, X_t > 1] \leq 1 - \delta \tag{3.8}$$

for some $\delta \in (0,1)$ and $\Delta > \delta$. Let $\tau := \min\{t \geq 0 \mid X_t \leq 1\}$ be the first index when $X_t$ passes 1. Then

$$E[\tau \mid \mathcal{F}_0] \leq \mathbb{1}_{\{X_0>1\}} \cdot (\ln(X_0) + \Delta)/\delta.$$

*Proof.* The proof is essentially the same as in [DJW12]. The idea is to use the additive drift theorem on the stochastic process $(Z_t)_{t\in\mathbb{N}}$, where $Z_t := \ln(X_t)$. Then (3.8) translates to

$$\begin{aligned}
&\mathbb{1}_{\{Z_t>0\}} \cdot E[\max\{Z_{t+1} - Z_t, -\Delta\} \mid \mathcal{F}_t] \\
&= \mathbb{1}_{\{X_t>1\}} \cdot E[\ln(\max\{X_{t+1}/X_t, e^{-\Delta}\}) \mid \mathcal{F}_t] \\
&\overset{(*)}{\leq} \mathbb{1}_{\{X_t>1\}} \cdot \ln(E[\max\{X_{t+1}/X_t, e^{-\Delta}\} \mid \mathcal{F}_t]) \\
&\leq \mathbb{1}_{\{X_t>1\}} \cdot \ln(1-\delta) \leq \mathbb{1}_{\{Z_t>0\}} \cdot (-\delta)
\end{aligned}$$

For $(*)$, we apply Jensen's inequality. Since $\tau = \min\{t \geq 0 \mid X_t \leq 1\} = \min\{t \geq 0 \mid Z_t \leq 0\}$, Theorem 3.3 finishes the proof. □

For the runtime analysis, we need a more general version of the multiplicative drift theorem. Sometimes, the stochastic process needs more than just one step to reduce its value by a constant factor, e. g., depending on the current configuration, the process might yield $X_{t+1} > X_t$, but for every situation, we have that $X_{t+2} < X_t \cdot (1-\delta)$. So, instead of the drift condition (3.8), only the weaker drift condition

$$E[\max\{X_{t+t_{max}}/X_t, e^{-\Delta}\} \mid \mathcal{F}_t, X_t > 1] \leq 1-\delta$$

holds for some $t_{max} \in \mathbb{N}$. In that case, we would expect $\tau$ to increase by a factor of $t_{max}$.

In a more general situation, the point in time $\sigma$ when $X_t$ decreases on expectation by at least a factor of $1-\delta$, i. e., the value $\sigma(t)$ for which

$$E[\max\{X_{\sigma(t)}/X_t, e^{-\Delta}\} \mid \mathcal{F}_t, X_t > 1] \leq 1-\delta$$

holds, might depend on the exact configuration at time $t$, e. g., we might have that either $X_{t+2} < X_t \cdot (1-\delta)$ or $X_{t+3} < X_t \cdot (1-\delta)$, and which alternative is true is known at time $t$ but is not known a priori. However, as long as $\sigma(t)$ is uniformly bounded by $t_{max}$, we can still expect that $\tau$ increases by at most a factor of $t_{max}$. The following generalized version of the multiplicative drift theorem formally confirms this statement.

**Theorem 3.5** (Multiplicative Drift Theorem with Variable Time Steps)**.** Let $X = (X_t)_{t\in\mathbb{N}}$ be a strictly positive stochastic process adapted to a filtration $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}$, and let for every $t \in \mathbb{N}$ $\sigma(t)$ be a $\mathcal{F}_t$-measurable, $\mathbb{N}$-valued random variable with $t < \sigma(t) \leq t + t_{max}$ almost surely for a constant $t_{max} \in \mathbb{N}$, such that

$$E[\max\{X_{\sigma(t)}/X_t, e^{-\Delta}\} \mid \mathcal{F}_t, X_t > 1] \leq 1 - \delta \tag{3.9}$$

for some $\delta \in (0,1)$ and $\Delta > \delta$. Let $\tau := \min\{t \geq 0 \mid X_t \leq 1\}$ be the first index when $X_t$ passes 1. Then

$$E[\tau \mid \mathcal{F}_0] \leq t_{max} \cdot \mathbb{1}_{\{X_0>1\}} \cdot (\ln(X_0) + \Delta)/\delta.$$

*Proof.* To simplify notation, we write $\sigma_k$ for $\sigma^{(k)}(0)$. For every $k \in \mathbb{N}$, we define $\hat{X}_k := X_{\sigma_k}$ and $\mathcal{F}_{\sigma_k} := \{A \in \mathcal{F} \mid A \cap \{\sigma_k \leq t\} \in \mathcal{F}_t\}$. Then, $\hat{X}_k$ is $\mathcal{F}_{\sigma_k}$-measurable and since

$$\begin{aligned}
&E[\max\{\hat{X}_{k+1}/\hat{X}_k, e^{-\Delta}\} \mid \mathcal{F}_{\sigma_k}] = E[\max\{X_{\sigma_{k+1}}/X_{\sigma_k}, e^{-\Delta}\} \mid \mathcal{F}_{\sigma_k}] \\
&= E\left[\sum_{t=0}^{\infty} \mathbb{1}_{\{\sigma_k=t\}} \cdot \max\{X_{\sigma_{k+1}}/X_{\sigma_k}, e^{-\Delta}\} \mid \mathcal{F}_{\sigma_k}\right] \\
&= \sum_{t=0}^{\infty} \mathbb{1}_{\{\sigma_k=t\}} \cdot E\left[\max\{X_{\sigma_{k+1}}/X_{\sigma_k}, e^{-\Delta}\} \mid \mathcal{F}_{\sigma_k}\right] \\
&= \sum_{t=0}^{\infty} \mathbb{1}_{\{\sigma_k=t\}} \cdot E\left[\max\{X_{\sigma(t)}/X_t, e^{-\Delta}\} \mid \mathcal{F}_t\right],
\end{aligned}$$

$\hat{X}_k > 1$ implies

$$\begin{aligned}
&E[\max\{\hat{X}_{k+1}/\hat{X}_k, e^{-\Delta}\} \mid \mathcal{F}_{\sigma_k}] \\
&= \sum_{t=0}^{\infty} \mathbb{1}_{\{\sigma_k=t\}} \cdot E\left[\max\{X_{\sigma(t)}/X_t, e^{-\Delta}\} \mid \mathcal{F}_t\right] \\
&\leq \sum_{t=0}^{\infty} \mathbb{1}_{\{\sigma_k=t\}} \cdot (1-\delta) = 1 - \delta,
\end{aligned}$$

i. e.,

$$\mathbb{1}_{\{\hat{X}_k>1\}} \cdot E[\max\{\hat{X}_{k+1}/\hat{X}_k, e^{-\Delta}\} \mid \mathcal{F}_{\sigma_k}] \leq \mathbb{1}_{\{\hat{X}_k>1\}} \cdot (1-\delta).$$

Theorem 3.4 yields for $\hat{\tau} = \min\{k \geq 0 \mid \hat{X}_k \leq 1\}$ the expected bound of

$$E[\hat{\tau} \mid \mathcal{F}_{\sigma_0}] \leq (\ln(\hat{X}_0) + \Delta)/\delta = (\ln(X_0) + \Delta)/\delta + 1.$$

Since $\tau \leq \hat{\tau} \cdot t_{max}$, we have that

$$E[\tau \mid \mathcal{F}_0] \leq t_{max} \cdot E[\hat{\tau} \mid \mathcal{F}_0] = t_{max} \cdot E[\hat{\tau} \mid \mathcal{F}_{\sigma_0}] \leq t_{max} \cdot (\ln(X_0) + \Delta)/\delta.$$

That finishes the proof. □

Although the result of our multiplicative drift theorem for continuous search spaces is a bound, only an additive constant larger than in the discrete case, the drift condition on $\max\{X_{t+1}/X_t, e^{-\Delta}\}$ is kind of impractical and technical harder to verify than a drift condition on $X_{t+1}/X_t$. Note that the construction of Example 3.5, showing that in an additive situation a drift condition on $X_{t+1} - X_t$ is insufficient, has no pendant in the multiplicative situation, where the process $X$ is allowed to take only positive values. Indeed, we can provide a multiplicative drift theorem, that is not a rewritten version of an additive drift theorem, but is proved directly.

**Theorem 3.6** (Multiplicative Drift Theorem II)**.** Let $X = (X_t)_{t \in \mathbb{N}}$ be a strictly positive stochastic process adapted to a filtration $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}$, such that

$$E[X_{t+1} \mid \mathcal{F}_t, X_t > 1] \leq (1 - \delta) \cdot X_t \tag{3.10}$$

for some $\delta \in (0, 1)$. Let $\tau := \min\{t \geq 0 \mid X_t \leq 1\}$ be the first index when $X_t$ passes 1. Then

$$E[\tau \mid \mathcal{F}_0] \leq \mathbb{1}_{\{X_0 > 1\}} \cdot (\ln(X_0) + 2)/\delta.$$

*Proof.* First, we define the stochastic process $Y = (Y_0, Y_1, \ldots)$ via

$$Y_0 := X_0, \quad Y_{t+1} := \begin{cases} X_{t+1}, & \text{if } \tau > t, \\ Y_t \cdot (1 - \delta), & \text{if } \tau \leq t. \end{cases}$$

That means $Y_t$ is equal to $X_t$ until time $\tau$. Afterwards, $Y$ decreases deterministically by a factor of $1 - \delta$ in every iteration. In particular, this implies that both processes reach a value $\leq 1$ at the same time, i. e., $\tau = \min\{t \geq 0 \mid Y_t \leq 1\}$, and $Y_t$ is $\mathcal{F}_t$-measurable and fulfilles the stronger drift condition

$$E[Y_t \mid \mathcal{F}_t] \leq (1 - \delta) \cdot Y_t.$$

By induction, it follows that

$$E[Y_{t+1} \mid \mathcal{F}_0] \leq (1 - \delta)^t \cdot Y_0 = (1 - \delta)^t \cdot X_0.$$

Therefore, we have for the first hitting time $\tau$:

$$P(\tau > t \mid \mathcal{F}_0) = P(Y_t > 1 \mid \mathcal{F}_0) \leq (1 - \delta)^t \cdot X_0,$$

where the inequality comes from Markov's inequality. It follows that

$$\begin{aligned}
\mathrm{E}[\tau \mid \mathcal{F}_0] &= \sum_{t=0}^{\infty} \mathrm{P}(\tau > t \mid \mathcal{F}_0) \\
&\leq \sum_{t=0}^{\lfloor -\log_{1-\delta} X_0 \rfloor} 1 + \sum_{t=\lceil -\log_{1-\delta} X_0 \rceil}^{\infty} (1-\delta)^t \cdot X_0 \\
&\leq -\frac{\ln(X_0)}{\ln(1-\delta)} + 1 + \sum_{s=0}^{\infty} (1-\delta)^s \\
&\leq \frac{\ln(X_0)}{\delta} + 1 + \frac{1}{\delta} \leq \frac{\ln(X_0)+2}{\delta}.
\end{aligned}$$

□

Of course, the same generalization, that allowed us to prove Theorem 3.5 from Theorem 3.4, can be applied here in order to obtain a version of Theorem 3.6, that allows for the drift to happen after a bounded random time. The result is the following theorem, which we will use for the analsis of PSO.

**Theorem 3.7** (Multiplicative Drift Theorem with Variable Time Steps II)**.** Let $X = (X_t)_{t\in\mathbb{N}}$ be a strictly positive stochastic process adapted to a filtration $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \ldots \subset \mathcal{F}$, and let for every $t \in \mathbb{N}$ $\sigma(t)$ be a $\mathcal{F}_t$-measurable, $\mathbb{N}$-valued random variable with $t < \sigma(t) \leq t + t_{\max}$ almost surely for a constant $t_{\max} \in \mathbb{N}$, such that

$$\mathrm{E}[X_{\sigma(t)} \mid \mathcal{F}_t, X_t > 1] \leq (1-\delta) \cdot X_t \tag{3.11}$$

for some $\delta \in (0,1)$. Let $\tau := \min\{t \geq 0 \mid X_t \leq 1\}$ be the first index when $X_t$ passes 1. Then

$$\mathrm{E}[\tau \mid \mathcal{F}_0] \leq \mathbb{1}_{\{X_0>1\}} \cdot t_{\max} \cdot (\ln(X_0)+2)/\delta.$$

*Proof.* The proof is completely analog to the proof of Theorem 3.5. □

The reason for the increased bounds for $\tau$ in all the drift theorems presented in this section, compared to "$\mathrm{E}[\tau \mid \mathcal{F}_0] \leq \ln(X_0)/\delta$" in the discrete case, is that over a discrete search space, we can use an argument like "if $X_t < 2$, then $X_t \leq 1$", i. e., as soon as the difference between the process and the optimum is smaller than the distance between two neighbors of the discrete space, the optimum is already reached. In the continuous situation, such an argument cannot hold and the remaining distance between "2" and "1" is not for free like in the discrete case, but needs to be paid with an additional constant number of steps.

## 3.4 The PSO Model

With the prerequisites from the previous sections, we can state PSO as a stochastic process. The following definition describes the positions, velocities and attractors of the classical PSO algorithm as random variables over the probability space $R$, as defined in Definition 3.3.

**Definition 3.8** (Classical PSO Process)**.** A *swarm* $\mathcal{S}$ of $N$ particles moves through the $D$-dimensional search space $\mathbb{R}^D$. Let $f : \mathbb{R}^D \to \mathbb{R}$ be the objective function. For $\mathcal{S}$, over the probability space $R$ from Definition 3.3 the stochastic process

$$S = (S_t)_{t\in\mathbb{N}_0} = ((X_t, V_t, L_t, G_t))_{t\in\mathbb{N}_0} = ((X_0, V_0, L_0, G_0), (X_1, V_1, L_1, G_1), \ldots)$$

is defined, consisting of

- $X_t = (X_t^{n,d})_{1\le n\le N, 1\le d\le D}$ ($d$-th coordinate of the *position* of particle $n$ after step $t$),
- $V_t = (V_t^{n,d})_{1\le n\le N, 1\le d\le D}$ ($d$-th coordinate of the *velocity* of particle $n$ after step $t$),
- $L_t = (L_t^{n,d})_{1\le n\le N, 1\le d\le D}$ ($d$-th coordinate of the *local attractor* of particle $n$ after step $t$),
- $G_t = (G_t^{n,d})_{1\le n\le N, 1\le d\le D}$, ($d$-th coordinate of the *global attractor* before step $t+1$ of particle $n$).

To simplify notation, $X_t^n$ denotes the vector $(X_t^{n,1}, ..., X_t^{n,D})$ (analogously, $V_t^n$, $L_t^n$, $G_t^n$) and $X_t^d$ the vector $(X_t^{1,d}, ..., X_t^{N,d})$ (analogously, $V_t^d$, $L_t^d$, $G_t^d$), whenever the context allows a clear distinction. Furthermore, $\tilde{G}_t^{n,d}$ denotes the $d$-th coordinate of the global attractor *after* the $t$-th step of particle $n$, i. e., $\tilde{G}_t^{n,d} = G_t^{n+1,d}$ if $n < N$, and $\tilde{G}_t^{N,d} = G_{t+1}^{1,d}$. With a given distribution for $(X_0, V_0)$ and the values $G_0^1 := \operatorname{argmin}_{1\le n\le N}\{f(X_0^n)\}$ and $L_0 := X_0$, $S_{t+1} = (X_{t+1}, V_{t+1}, L_{t+1}, G_{t+1})$ is determined by the following recursive equations, the *movement equations*:

- $V_{t+1}^{n,d} = \chi \cdot V_t^{n,d} + c_1 \cdot r_{t+1}^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) + c_2 \cdot s_{t+1}^{n,d} \cdot (G_t^{n,d} - X_t^{n,d})$ for $t \ge 0$,
- $X_{t+1}^{n,d} = X_t^{n,d} + V_{t+1}^{n,d}$ for $t \ge 0$,
- $L_{t+1}^n = \operatorname{argmin}_{\{X_{t+1}^n, L_t^n\}} f$,

- $G_t^{n+1} = \operatorname{argmin}_{\{L_{t+1}^n, G_t^n\}} f$ for $t \geq 0$, $1 \leq n \leq N-1$,
- $G_{t+1}^1 = \operatorname{argmin}_{\{L_{t+1}^N, G_t^N\}} f$ for $t \geq 0$.

In case of a tie when applying argmin, similar to the PSO version in [OH07], the new value prevails, i. e., whenever a particle finds a search point with value equal to the one of its local attractor, this point becomes the new local attractor. If additionally the value is equal to the one of the global attractor, this one is also updated. Here, $\chi$, $c_1$ and $c_2$ are some positive constants called the *fixed parameters* of $\mathcal{S}$, and $r_t^{n,d}$, $s_t^{n,d}$ are uniformly distributed over $[0,1]$ and all independent, i. e., each of the $r_t^{n,d}$ and $s_t^{n,d}$ is identified with a different $\omega_i$ from Definition 3.3.

If after the $t$-th step the process is stopped, the *solution* found by $\mathcal{S}$ so far is $G_t^N$. This process describes exactly the PSO variant of Algorithm 1 in Section 2.2. The major part of this thesis will study the limit of the sequence $(G_t^N)_{t\in\mathbb{N}}$ as $t$ tends to $\infty$.

Definition 3.8 describes the common movement equations with the same two specifications as stated in the algorithmic description of PSO (Section 2.2, Algorithm 1): If a particle visits a point with the same objective value as its current local attractor or the current global attractor, then the respective attractor is updated to the new point. As pointed out in [OH07], this is crucial in the case of objective functions that are constant on some area with a positive Lebesgue measure. As the second specification, the global attractor is updated after every step of a single particle, not only after every iteration of the whole swarm. Another common variant of PSO, sometimes known as *parallel PSO*, only updates the global attractor after every iteration of the whole swarm. However, due to the choice we make here, the information shared between the particles is as recent as possible.

From Definition 3.8, it follows that $(S_t)_{t\in\mathbb{N}}$ has the Markov property, i. e., the distribution of $S_{t+t_0}$ given all the previous values $(S_0, \ldots, S_{t_0})$ is the same as the distribution of $S_{t+t_0}$ given only $S_{t_0}$. This is clear because $S_{t+1}$ is formulated in terms of $S_t$ and some random variables $r_{t+1}^{n,d}$ and $s_{t+1}^{n,d}$ which are independent of the past of $S$. Another interesting property of this stochastic process that follows immediately from the movement equations is the following:

**Observation 3.1.** Let $\mathcal{S}$ be a swarm and $((X_t, V_t, L_t, G_t))_{t\in\mathbb{N}_0}$ its corresponding stochastic process. Let $\mathcal{L}^k$ denote the $k$-dimensional Lebesgue measure, $\mathcal{L}[Y]$ the distribution of a random variable $Y$ and "$\ll$" (just in this observation) absolute continuity between two distributions. Assuming $\mathcal{L}[(X_0, V_0)]$

$\ll \mathcal{L}^{2\cdot N\cdot D}$, it follows $\mathcal{L}[X_t] \ll \mathcal{L}^{N\cdot D}$ for every $t \geq 0$. If $X_t^n \neq G_t^N$ for every $n$, then for every $t' > t$, $\mathcal{L}[X_{t'} \mid S_t] \ll \mathcal{L}^{N\cdot D}$ almost surely.

This observation follows from the structure of the movement equations, i. e., the entries of $X_t$ are weighted sums of the entries of $X_0$, which by assumption has probability 0 to hit any fixed Lebesgue null set, and values that are chosen uniformly and independently from certain intervals. If $X_t^n \neq G_t^N$ for every $n$, then for every $t' > t$, the entries of $X_{t'}$ are sums of constants depending on $S_t$ and a non-empty sum of values, chosen uniformly and independently from certain intervals.

For the coming analysis, we introduce the notion of a *potential* of a swarm. Basically, the potential is an extension of the physical interpretation of the particle swarm model. If the particles move faster and get farther away from their global attractor, the potential increases. If the swarm converges, the potential tends towards 0. At the different stages of the analysis, different measures for the potential will be useful. The following definition describes the two most frequently used formulations for a measure of the potential.

**Definition 3.9** (Potential)**.** For $a > 0$, we define the measure $\Phi_t^{n,d}$ for the potential of swarm $\mathcal{S}$ in dimension $d$ right before the $t$-th step of particle $n$ as

$$\Phi_t^{n,d} := \sqrt{\sum_{n'=1}^{n-1} \left( a|V_t^{n',d}| + |G_{t-1}^{n,d} - X_t^{n',d}| \right) + \sum_{n'=n}^{N} \left( a|V_{t-1}^{n',d}| + |G_{t-1}^{n,d} - X_{t-1}^{n',d}| \right)}.$$

To simplify notation, $\Phi_t^{n,d}$ describes the same potential measure in dimension $d$ *after* the $t$-th step of particle $n$, i. e.,

$$\Phi_t^{n,d} := \sqrt{\sum_{n'=1}^{n} \left( a|V_t^{n',d}| + |G_{t-1}^{n,d} - X_t^{n',d}| \right) + \sum_{n'=n+1}^{N} \left( a|V_{t-1}^{n',d}| + |G_{t-1}^{n,d} - X_{t-1}^{n',d}| \right)}.$$

As another measure for the potential, we define the potential of particle $n$ in dimension $d$ right after the $t$'th step of the last particle $N$ as

$$Y_t^{n,d} := \sqrt{|V_t^{n,d}|} + \sqrt{|G_t^{1,d} - X_t^{n,d}|}.$$

While $\Phi$ only measures the potential of the whole swarm, $Y_t^n$ is able to measure the potential of a single particle. Note that this are not the only possible choices for a meaningful measure of the swarm's potential. However, for technical reasons explained later, this particular potential measures,

including the additional parameter $\alpha$ for later use and the occurring square roots, are necessary. General tendencies towards $0$ or $\infty$ are invariant under the different measures. In particular, $\Phi^1_{t+1}$ and $\sum_{n=1}^{N} Y_t^{n,d}$ are equivalent in the sense that they deviate by at most a constant factor.

## 3.5 Discussion of Previous Results

In the light of this new model for PSO, which we will use for the rest of this thesis, some of the previous positive and negative results, relevant for the results of this thesis, need to be revisited in more detail.

### 3.5.1 Negative Results

Since the goal of this thesis is to prove rigorous results about the convergence of the PSO process and the runtime, a discussion of the various negative results, which appear to be in contradiction to the results of this thesis, is necessary. A very popular negative result comes from considering the following situation (e. g., [vdBE10, Wit09, PE03, BM14]).

$$\begin{aligned} X_t^1 = L_t^1 = \ldots = X_t^N = L_t^N = G_t^1 = z, \\ V_t^1 = \ldots = V_t^N = 0, \end{aligned}$$

for some $z \in \mathbb{R}^D$, i. e., all the attractors and particles' positions are located at the same point $z \in \mathbb{R}^D$ and every velocity is $0$. From Definition 3.8, it follows that

$$\begin{aligned} X_{t'}^1 = L_{t'}^1 = \ldots = X_{t'}^N = L_{t'}^N = G_{t'}^1 = z, \\ V_{t'}^1 = \ldots = V_{t'}^N = 0, \end{aligned}$$

for every $t' \geq t$. That means that the particles will stay at $z$ forever, no matter how good or bad $f(z)$ is. Therefore, so the claim frequently found in the literature, providing any positive result about the quality of the returned solution or even about the runtime is impossible.

However, if the positions and velocities of the initial population are distributed in some natural way, e. g., uniformly at random over $K$, Observation 3.1 states that the swarm has similar restrictions as a process consisting

only of variables that are sampled u. a. r. in the sense that events with probability 0 in the latter case also have probability 0 in the first case. Since $\mathbb{R}^D$ is not enumerable, under uniform sampling no point in $\mathbb{R}^D$ is drawn more than once and therefore the particle swarm also does not visit any point more than once. This implies that this well-studied equilibrium when every particle is at the global attractor and has velocity 0 is a state that the process may converge to but that can never be reached. In other words: The probability for the above mentioned equilibrium to be reached within any finite time $t$ is 0 and therefore, this pathological event has no effect on the expected runtime.

Another, yet more refined negative result can be found in [LW11]. Here, the authors consider the very simple, 1-dimensional objective function Sphere, defined as

$$\text{Sphere}(x) = x^2,$$

and construct an event that indeed has a positive probability to occur and from which the particles will with probability 1 not reach a certain neighborhood of the unique local and therefore global optimum at 0. To be precise, the authors of [LW11] proof the following theorem.

**Theorem 3.8** (Proposition 1 in [LW11])**.** The first hitting time $\tau_\varepsilon$ with respect to $\varepsilon$ is defined as the first time when the algorithm obtains a search point $x \in \mathbb{R}^D$, such that $|f(x)| < \varepsilon$, i. e., the first time when the algorithm finds a search point with a value at most $\varepsilon$ worse than the optimum. Then there is an $\varepsilon > 0$, such that the classical PSO with inertia factor $\chi < 1$ and one particle has infinite expected first hitting time with respect to $\varepsilon$ on the 1-dimensional Sphere.

In the following, we briefly outline the proof idea. If (without loss of generality) the particle is initialized at a negative position and has a positive velocity, then both attractors will be equal to its current position until it passes

the optimum at 0. This implies that until the optimum is passed, we have $V_t^{1,1} = \chi^t \cdot V_0^{1,1}$ and therefore

$$\begin{aligned} X_t^{1,1} &= X_0^{1,1} + \sum_{s=0}^{t-1} \cdot V_s^{1,1} \\ &= X_0^{1,1} + V_0^{1,1} \cdot \sum_{s=0}^{t-1} \chi^s \\ &\leq X_0^{1,1} + V_0^{1,1} \cdot \sum_{s=0}^{\infty} \chi^s \\ &\leq X_0^{1,1} + V_0^{1,1} \cdot \frac{1}{1-\chi}. \end{aligned}$$

Since $\chi < 1$, $1/(1-\chi)$ is a bounded value, there is a positive probability for the event $X_0^{1,1} + V_0^{1,1} \cdot \frac{1}{1-\chi} < -\sqrt{\varepsilon}$, if $\varepsilon > 0$ is sufficiently small. In [LW11], the authors call such an event a *bad initialization event*. Consequently, there is a positive probability for the particle to never hit any value closer than $\sqrt{\varepsilon}$ to the optimum at 0, which implies that for the considered objective function SPHERE, no point with a function value of at most $\varepsilon$ will be visited. Note that the condition $\chi < 1$ is not a true restriction since this is one of the convergence conditions identified in [JLY07b, JLY07a]. Therefore, the authors of [LW11] have proved that the classical PSO with only one particle is not an effective optimizer.

On the other hand, for generalizing the same idea to the case of more than one particle, further restrictions to the parameters $\chi$, $c_1$ and $c_2$ are necessary. The result for the case of two particles is as follows.

**Theorem 3.9** (Theorem 1 in [LW11])**.** Consider the classical PSO with two particles on the 1-dimensional SPHERE. If $\chi < 1$, $1 < c_2 < 2$, $V_0^{1,1}, V_0^{2,1} \leq 0$, $\kappa < 1$ where

$$\begin{aligned} \kappa := \; & \frac{c_2^2 - 2 \cdot c_2 + 2 + 2 \cdot \chi \cdot c_2}{4 \cdot c_2} \\ & + \frac{\sqrt{(c_2^2 - 2 \cdot c_2 + 2 + 2 \cdot \chi \cdot c_2) \cdot (c_2^2 + 6 \cdot c_2 + 2 + 2 \cdot \chi \cdot c_2)}}{4 \cdot c_2}, \end{aligned}$$

and

$$X_0^{1,1}, X_0^{2,1} > 2 \cdot \varepsilon + 2 \cdot c_2 \cdot \left( \frac{|X_0^{2,1} - X_0^{1,1}| + |V_0^{1,1}| + |V_0^{2,1}|}{(1-\chi) \cdot (1-\kappa)} \right)$$

all hold together, then the expected first hitting time with respect to $\varepsilon$ is infinite.

The proof idea is the same as in Theorem 3.8, i. e., under the stated assumptions, the swarm will with positive probability never pass the optimum. However, the crucial prerequisite of Theorem 3.9 is $\kappa < 1$. We can easily verify that this is typically violated by parameters common in the literature. Table 3.1 provides an overview over some parameter choices famous in the literature and the resulting values for $\kappa$.

**Table 3.1:** Famous parameter choices and the resulting values for $\kappa$

| $\chi$ | $c_1$ | $c_2$ | $\kappa$ |
|---|---|---|---|
| 0.72984 | 1.49617 | 1.49617 | 1.78756 |
| 0.72984 | 2.04355 | 0.94879 | 1.91517 |
| 0.6 | 1.7 | 1.7 | 1.66267 |

Therefore, the result from [LW11] does not contradict a runtime analysis of the classical PSO with appropriately chosen parameters.

### 3.5.2 Convergence Analysis

The convergence analysis from [JLY07b, JLY07a] provides results about the expectation and the variance of the particles' positions under the assumption that the local and global attractor are constant. Although in this thesis we do not make this assumption, we will use an altered version of their proof in Chapter 4, therefore we present the main results and the proof here. With constant local attractors $L_t^n =: L^n$ and a constant global attractor $G_t^n =: G$, we can write movement equations can as

$$V_{t+1}^{n,d} = \chi \cdot V_t^{n,d} + c_1 \cdot r_{t+1}^{n,d} \cdot (L^{n,d} - X_t^{n,d}) + c_2 \cdot s_{t+1}^{n,d} \cdot (G^d - X_t^{n,d}), \quad (3.12)$$

$$X_{t+1}^{n,d} = X_t^{n,d} + V_{t+1}^{n,d}. \quad (3.13)$$

Since the processes $((X_t^{n_1,d_1}, V_t^{n_1,d_1}))_{t\in\mathbb{N}}$ and $((X_t^{n_2,d_2}, V_t^{n_2,d_2}))_{t\in\mathbb{N}}$ are independent of each other if $n_1 \neq n_2$ or $d_1 \neq d_2$, it is sufficient to study just one of these processes, or equivalently formulated, to study the case of only one particle in a 1-dimensional search space. Therefore, the upper indices $n$ and $d$ will be omitted for the rest of this section.

**Theorem 3.10** (Theorem 1 in [JLY07b]). Given $\chi, c_1, c_2 \geq 0$, if and only if $0 \leq \chi < 1$ and $0 < c_1 + c_2 < 4 \cdot (1 + \chi)$, iterative process $(E[X_t])_{t\in\mathbb{N}}$ is guaranteed to converge to

$$\mu := \frac{c_1 \cdot L + c_2 \cdot G}{c_1 + c_2}.$$

*Proof.* The proof from [JLY07b] will be briefly recalled. Rewriting Equation (3.13) as $V_{t+1} = X_{t+1} - X_t$, we can in Equation (3.12) replace $V_{t+1}$ by $X_{t+1} - X_t$ and $V_t$ by $X_t - X_{t-1}$. After collecting terms, we obtain

$$X_{t+1} = (1 + \chi - c_1 \cdot r_{t+1} - c_2 \cdot s_{t+1}) \cdot X_t - \chi X_{t-1} + c_1 \cdot r_{t+1} \cdot L + c_2 \cdot s_{t+1} \cdot G \quad (3.14)$$

and consequently

$$E[X_{t+1}] = (1 + \chi - \frac{c_1 + c_2}{2}) \cdot E[X_t] - \chi \cdot E[X_{t-1}] + \frac{c_1 \cdot L + c_2 \cdot G}{2}. \quad (3.15)$$

Equation (3.15) is a linear recursion of second order for the sequence $(E[X_t])_{t\in\mathbb{N}}$ with characteristic polynomial

$$\lambda^2 - \left(1 + \chi - \frac{c_1 + c_2}{2}\right) \cdot \lambda + \chi.$$

The sequence $(E[X_t])_{t\in\mathbb{N}}$ converges if and only if the eigenvalues

$$\lambda_1, \lambda_2 = \frac{1}{2} \cdot \left(\left(1 + \chi - \frac{c_1 + c_2}{2}\right) \pm \sqrt{\left(1 + \chi - \frac{c_1 + c_2}{2}\right)^2 - 4 \cdot \chi}\right)$$

both have an absolute value strictly less than 1. As straightforward calculations show, this is the case for $0 \leq \chi < 1$ and $0 < c_1 + c_2 < 4 \cdot (1 + \chi)$. The actual limit $\mu$ of the sequence is then found by solving the equation

$$\mu = (1 + \chi - c_1 \cdot r_{t+1} - c_2 \cdot s_{t+1}) \cdot \mu - \chi \cdot \mu + c_1 \cdot r_{t+1} \cdot L + c_2 \cdot s_{t+1} \cdot G,$$

which is obtained from Equation (3.14) by replacing $X_{t+1}$, $X_t$ and $X_{t-1}$ with $\mu$. □

Similarly, the authors of [JLY07b, JLY07a] calculated conditions for the sequence $(\mathrm{Var}[X_t])_{t\in\mathbb{N}}$ to converge. Their result is

**Theorem 3.11** (Theorem 3 in [JLY07b])**.** Given $\chi, c_2, c_2 \geq 0$, if and only if $0 \leq \chi < 1$, $c_1 + c_2 > 0$ and $f > 0$ where

$$f := 1 - \left( \left( 1 + \chi - \frac{c_1 + c_2}{2} \right)^2 + \frac{1}{12} \cdot (c_1^2 + c_2^2) - \chi \right) \cdot (1 - \chi) - \chi^3$$

are all satisfied together, iterative process $(\mathrm{Var}[X_t])_{t \in \mathbb{N}}$ is guaranteed to converge to

$$\frac{1}{6} \cdot \left( \frac{c_1 \cdot c_2}{c_1 + c_2} \right)^2 \cdot (G - L)^2 \cdot \frac{1 + \chi}{f}.$$

As a consequence, if the parameter restrictions from Theorem 3.10 and Theorem 3.11 hold and $G = L$, i. e., the global attractor is equal to the local attractor, then both theorems together imply that the positions of all particles converge towards $\mu$ almost surely because the expectation $E[X_t]$ converges towards $\mu$ and the variance $\mathrm{Var}[X_t]$ converges towards 0. Additionally, we can derive a result about the convergence speed. Since the solution of the recursive Equation 3.15 has the form

$$E[X_t] = d_1 \cdot \lambda_1^t + d_2 \cdot \lambda_2^t + \mu$$

for some constants $d_1$ and $d_2$, the expected difference between $X_t$ and $\mu$ decreases exponentially in $t$.

Although this result relies on constant local and global attractors, a prerequisite that in general we cannot assume, the calculations can be generalized to the case of attractors that are not a priori constant, but have a bounded and small area, e. g., a neighborhood of the global optimum, which they do not leave. We will introduce, prove and use this generalization in the second part of Chapter 4.

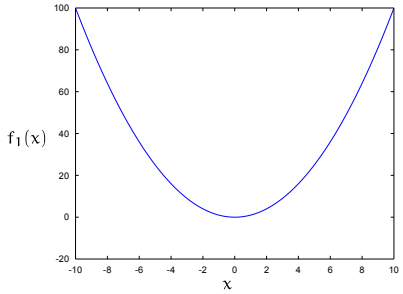
# 4. Convergence of 1-dimensional PSO

After invalidating the different negative results, stating that classical particle swarm optimization (PSO) was unable to converge towards a local optimum, this chapter contains the main theoretical results of this thesis, namely positive results about the quality and the convergence speed of the unmodified PSO algorithm for certain classes of 1-dimensional objective functions. In the first part of this chapter, we present a proof of the convergence of PSO towards a local optimum, showing that under comparatively mild assumptions about the objective function, the particles find at least a local optimum. In the second part, we apply the drift theory methods introduced in Chapter 3 in order to prove that for an also very large class of objective functions, PSO converges towards the optimum with linear convergence speed, i. e., the distance between the global attractor and the actual optimum is halved within a constant number of iterations.

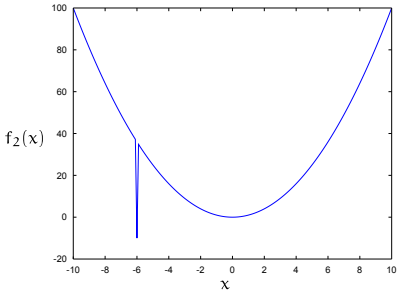
Note that we prove all the results in this chapter completely rigorously, that means we only make certain restrictions on the set of admissible objective functions, while we do not modify the PSO algorithm. Instead, we fully take its stochastic nature into account and make no unproved assumptions about the behavior of the particles.

## 4.1 Particle Swarm Optimization Almost Surely Finds Local Optima

Finding a global optimum of a given objective function $f$ is the ultimate goal of any optimization method. However, in a continuous domain, achieving this goal is hardly possible because two functions with different global optima can be very similar and might differ only on an arbitrary small subset of the search space. For an example of two such functions $f_1$ and $f_2$, see Figure 4.1.




**(a)** Objective function $f_1(x)$.




**(b)** Objective function $f_2(x)$.

**Figure 4.1:** Two possible objective functions $f_1$ and $f_2$. The area on which both are equal is large but the global optima are far away from each other.

Therefore, in order to handle such functions, an algorithm needs to enter every arbitrary small subcube of the search space. This request is in contradiction with convergence because convergence of the algorithm implies that after some finite time only points within a certain neighborhood are queried. Therefore, for the rest of the search space, only a finite number of function evaluations is performed, which is insufficient to hit every arbitrary small subcube. Therefore, the actual goal of our analysis performed here is to prove that the classical PSO algorithm finds at least a local optimum.

PSO is designed to handle any objective function, but for the rest of this section, only objective functions from the set $\mathbb{F}$ defined below are considered.

**Definition 4.1.** Let $f : \mathbb{R}^D \to \mathbb{R}$ be a function. $f \in \mathbb{F}$ if and only if

(i) there is a compact set $K \subset \mathbb{R}^D$ with positive Lebesgue measure, such that $P(X_0^n \in K) = 1$ for every $n$ and $\left\{x \in \mathbb{R}^D \mid f(x) \leq \sup_K f\right\}$ (the island) is bounded;

(ii) $f$ is continuous.

Restriction (i) states that there is a compact set $K$ such that for all $x \in K$, the set of all search points $y$ at least as good as $x$, i. e., all $y$ with $f(y) \leq f(x)$, is a bounded set. For illustration of this restriction, see Figure 4.2.

Since the particles are initialized inside $K$ and since $f(G_t^n)$ is non-increasing in $t$, (i) ensures that the possible area for the global attractor is limited if the positions of all particles are initialized inside of $K$ (being on any point of the

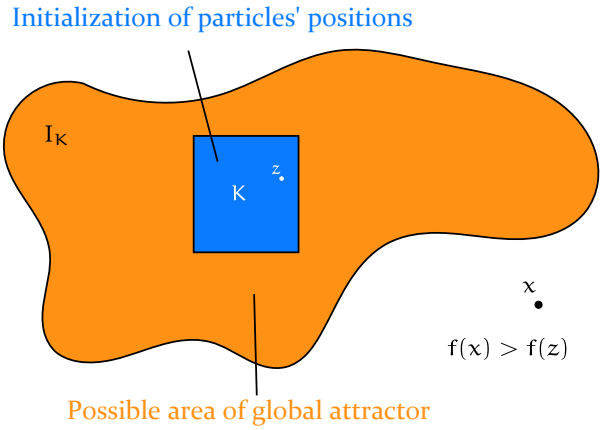
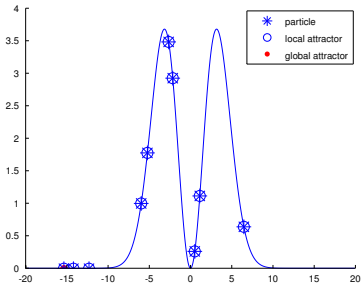
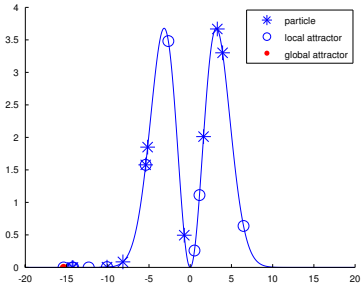



**Figure 4.2:** Valid objective functions.

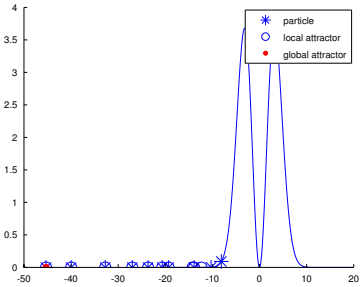
island is better than being in the sea). If for example $\lim_{|x|\to\infty} f(x) = \infty$ or if $f$ has compact support and is negative on $K$, (i) is already satisfied. E. g., common benchmark functions like the function SPHERE or the function ROSENBROCK ([Ros60]) are in $\mathbb{F}$. On functions that violate (i), the swarm might move forever because either they do not necessarily have a local optimum like $f(x) = x$ or they have an optimum, but improvements can be made arbitrary far away from it, like, e. g. in the case of the function $f(x) = x^2/(x^4+1)$, where $x = 0$ is the only local and the global optimum, but if the particles are far away from 0, they tend to further increase the distance because $f$ converges to 0 as $|x|$ approaches $\infty$. Figure 4.3 gives an illustration of this situation. Under such circumstances, convergence cannot be expected and it is necessary to restrict the function class in order to avoid this. However, (ii) might be the only true restriction. Note that every continuous function is in particular measurable, therefore restricting the set of admissible objective functions according to Definition 4.1 avoids the problem mentioned in Section 3.1.2.




**(a)** After initialization




**(b)** After iteration 1




**(c)** After iteration 2

**Figure 4.3:** Particles processing $f(x) = x^2/(x^4 + 1)$ and moving away from the only local optimum because of the misleading landscape.

### 4.1.1 Proof of Convergence Towards a Local Optimum

For proving the convergence of a particle swarm, we first define the exact sense of "swarm convergence."

**Definition 4.2** (Swarm Convergence)**.** Swarm $\mathcal{S}$ *converges* if there almost surely is a point $z$ such that the following two conditions hold:

1. $\lim_{t\to\infty} V_t = 0$ (the movement of the particles tends to zero),
2. $\lim_{t\to\infty} X_t^n = z$ for each $n \in \{1, \ldots, N\}$ (every particle moves towards $z$).

A consequence of the above conditions is that $\lim_{t\to\infty} G_t^n = z$ almost surely and $\lim_{t\to\infty} L_t^n = z$ almost surely for every $n \in \{1, \ldots, N\}$. Although the convergence analysis in the literature ([JLY07a]) usually makes the assumption that at least the global attractor is constant forever, a prerequisite

that because of Observation 3.1 cannot be assumed here, the generalization of the convergence proof from [JLY07a], showing that their results still hold under the weaker assumption of only the convergence of the attractors, is straight-forward.

Our goal is to prove more, namely, that under the stated assumptions about $f$ the swarm is able to find a local minimum. Here, the notion of the potential of a swarm comes into play. Roughly speaking, as long as the swarm has potential high enough to overcome the distance to at least one local minimum, the probability to find it within a few steps is positive. A problem occurs when the value of the potential is too low for the swarm to overcome the distance to the next optimum by only a small number of steps. In other words, if $f$ is monotonically decreasing in some direction and on an area that is large in comparison to the potential of the swarm, the particles must be able to "run down the hill," i. e., they must be able to surpass every distance as long as $f$ decreases. The following definition formally describes the situation of a swarm while it is "running down the hill."

**Definition 4.3** (Running Particle Swarm)**.** Let $d_0 \leq D$ be an arbitrary dimension. The swarm $\mathcal{S}$ is called *positively running* in direction $d_0$ at time $t$, if the following properties hold for every $n \in \{1, \ldots, N\}$:

- $G_t^{n,d_0} = \max_{1 \leq i \leq N}\{X_{t'}^{i,d_0}\}$ for $t' = t+1$ if $i < n$ and $t' = t$ otherwise,
- $\operatorname{sign}(L_t^{n,d_0} - X_t^{n,d_0}) \geq 0$,
- $V_t^n \geq 0$ for every $n$.

Analogously, the swarm is called *negatively running* in direction $d_0$ at time $t$, if

- $G_t^{n,d_0} = \min_{1 \leq i \leq N}\{X_{t'}^{i,d_0}\}$ for $t' = t+1$ if $i < n$ and $t' = t$ otherwise,
- $\operatorname{sign}(L_t^{n,d_0} - X_t^{n,d_0}) \leq 0$,
- $V_t^n \leq 0$ for every $n$.

Intuitively, one may think of running as the behavior a swarm shows when it moves through an area that is monotone in one dimension $d_0$, while changes in any other dimension are insignificant. Therefore, in case of a positively running swarm, the larger the $d_0$'th entry of a position or attractor is, the better is its function value. Note that if the state of being running is maintained long enough, all particles will eventually overcome their own local

attractors. From that point on, as long as the swarm stays running, the local attractors of the particles are identical to their current positions. An example of a running swarm is presented in Figure 4.4.

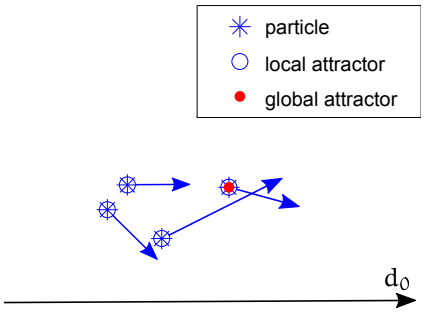



**Figure 4.4:** A particle swarm in the state called "running".

**Example 4.1.** Consider a 1-dimensional particle swarm and the objective function $f(x) = -x$. Assume that the velocities of the particles are all positive. Then the swarm is positively running in direction 1 forever. It is obvious that the position with the greatest $x$-value leads to the smallest value of $f(x)$ and therefore becomes the global attractor. It remains to prove that the velocity of every particle stays positive. Given the old velocity $V_t^{n,1}$, the new velocity $V_{t+1}^{n,1}$ is a positive linear combination of the three components $V_t^{n,1}$, $G_t^{n,1} - X_t^{n,1}$ and $L_t^{n,1} - X_t^{n,1}$. The value for $V_t^{n,1}$ is positive by assumption, $G_t^{n,1} - X_t^{n,1}$ and $L_t^{n,1} - X_t^{n,1}$ are non-negative since $G_t^{n,1} \geq L_t^{n,1} \geq X_t^{n,1}$. Therefore, the velocity stays positive and the swarm will stay positively running forever. In that situation, a good behavior would be increasing (or at least non-decreasing) $\Phi$.

The negative results of [LW11], recalled in Section 3.5.1 of the previous chapter, can be read in the following way: "Under certain conditions on the parameters $\chi$, $c_1$ and $c_2$, a running swarm loses potential and therefore converges."

Informally speaking, if a swarm $\mathcal{S}$ has a too small $\Phi$ to make it to the next local minimum, it is necessary that $\Phi$ increases after $\mathcal{S}$ has become running, and so $\Phi$ enables the swarm to overcome every distance. The following lemma is the central technical observation of this section and makes a statement about how to choose the parameters to make sure that a running swarm has an exponentially increasing potential.

**Lemma 4.1** (Running to Infinity Lemma)**.** For certain parameters $N$, $\chi$, $c_1$ and $c_2$, there is a $q$, $0 < q < 1$, such that the event that the swarm $\mathcal{S}$ is positively (negatively, resp.) running in direction $d_0$ and has a positive potential $\Phi$ implies that on expectation the reciprocal of the potential $\Phi$ decreases by at least a factor of $q$, in terms:

$$\mathrm{E}\left[\frac{\Phi_t^{1,d_0}}{\Phi_{t+1}^{1,d_0}} \mid \Phi_t^{1,d_0} > 0 \wedge \mathcal{S} \text{ positively running}, \mathcal{F}_{t-1}\right] < q. \tag{4.1}$$

Therefore, if $\mathcal{S}$ stays positively running forever, then $V_t^{n,d_0} + X_t^{n,d_0}$ ($-V_t^{n,d_0} - X_t^{n,d_0}$, resp.) tends to $\infty$ for every $n$ almost surely and the swarm leaves every bounded set $B \subset \mathbb{R}^D$ almost surely.

*Proof.* To ease up notation, the upper index $d_0$ is omitted for the rest of this proof. Without loss of generality, we can assume that the swarm is positively running. Note that due to Observation 3.1, the case $\Phi_t^n = 0$ for some $t$ and some $n$ has probability 0 and can therefore be neglected.

First, we need to bound the expression $\mathrm{E}\left[\Phi_t^1/\Phi_{t+1}^1 \mid \Phi_t^1 > 0 \wedge \mathcal{S} \text{ positively running}, \mathcal{F}_{t-1}\right]$ for the concrete choice of potential from Definition 3.9.

Values for $N$, $\chi$, $c_1$ and $c_2$, for which this potential fulfills Equation (4.1) for a $q < 1$ need to be determined. In other words, during one iteration of all particles, the reciprocal of the potential should decrease on expectation by at least a factor of $q$. By inserting the definition of $\Phi$ and applying the fact that the swarm is positively running and therefore $X_t^n \le G_t^1$ and $V_{t-1}^n \ge 0$ for every $n$, we obtain:

$$\begin{aligned}
\frac{\Phi_t^1}{\Phi_{t+1}^1} &= \sqrt{\frac{\sum_{n'=1}^{N}\left(a \cdot |V_{t-1}^{n'}| + |G_{t-1}^1 - X_{t-1}^{n'}|\right)}{\sum_{n'=1}^{N}\left(a \cdot |V_t^{n'}| + |G_t^1 - X_t^{n'}|\right)}} \\
&= \sqrt{\frac{\sum_{n'=1}^{N}\left(a \cdot V_{t-1}^{n'} + G_{t-1}^1 - X_{t-1}^{n'}\right)}{\sum_{n'=1}^{N}\left(a \cdot V_t^{n'} + G_t^1 - X_t^{n'}\right)}} \\
&= \sqrt{\frac{\sum_{n'=1}^{N}\left(a \cdot V_{t-1}^{n'} + G_{t-1}^1 - X_{t-1}^{n'}\right)}{\sum_{n'=1}^{N}\left(a \cdot V_t^{n'} + N \cdot (G_{t-1}^{n'} - G_{t-1}^{n'}) + G_{t-1}^1 - X_t^{n'}\right)}} \\
&= \sqrt{\frac{1}{\sum_{n'=1}^{N}\frac{w_{n'}}{\chi_{n'}}}} \le \sum_{n'=1}^{N} w_{n'} \cdot \sqrt{\chi_{n'}},
\end{aligned}$$

where

$$w_{n'} := \frac{a \cdot V_{t-1}^{n'} + G_{t-1}^{1} - X_{t-1}^{n'}}{\sum_{i=1}^{N} \left(a \cdot V_{t-1}^{i} + G_{t-1}^{1} - X_{t-1}^{i}\right)}$$

and

$$x_{n'} := \frac{a \cdot V_{t-1}^{n'} + G_{t-1}^{1} - X_{t-1}^{n'}}{a \cdot V_{t}^{n'} + N \cdot (\tilde{G}_{t-1}^{n'} - G_{t-1}^{n'}) + G_{t-1}^{1} - X_{t}^{n'}}.$$

The last inequality follows from the generalized weighted mean inequality between the $(-1)$-mean and the $(1/2)$-mean with weights $w_{n'}$. Note that the $w_{n'}$ sum up to 1 and that they only depend on $S_{t-1}$, i. e., they are $\mathcal{F}_{t-1}$-measurable. Therefore, it follows

$$\begin{aligned} E\left[\frac{\Phi_t^1}{\Phi_{t+1}^1} \;\middle|\; \mathcal{F}_{t-1}\right] &\leq E\left[\sum_{n'=1}^{N} w_{n'} \cdot \sqrt{x_{n'}} \;\middle|\; \mathcal{F}_{t-1}\right] \\ &= \sum_{n'=1}^{N} w_{n'} \cdot E\left[\sqrt{x_{n'}} \;\middle|\; \mathcal{F}_{t-1}\right] \leq \max_{n'=1\ldots N} E\left[\sqrt{x_{n'}} \;\middle|\; \mathcal{F}_{t-1}\right] \end{aligned}$$

It remains to show, that $E\left[\sqrt{x_{n'}} \;\middle|\; \mathcal{F}_{t-1}\right] \leq q$ for every $n'$ and a $q < 1$. By applying the definition of $x_{n'}$ and the movement equations and replacing the non-negative term $c_1 \cdot r_t^{n'} \cdot (L_{t-1}^{n'} - X_{t-1}^{n'})$ with 0 in the first "$\leq$", we obtain

$$\begin{aligned} x_{n'} &= \frac{a \cdot V_{t-1}^{n'} + G_{t-1}^{1} - X_{t-1}^{n'}}{a \cdot V_{t}^{n'} + N \cdot (\tilde{G}_{t-1}^{n'} - G_{t-1}^{n'}) + G_{t-1}^{1} - X_{t}^{n'}} \\ &\leq \frac{a \cdot V_{t-1}^{n'} + G_{t-1}^{1} - X_{t-1}^{n'}}{a \cdot (\chi \cdot V_{t-1}^{n'} + c_2 \cdot s_t^{n'} \cdot (G_{t-1}^{n'} - X_{t-1}^{n'})) +} \\ &\qquad \overline{\cdots + N \cdot \max\{0, X_{t-1}^{n'} + \chi \cdot V_{t-1}^{n'} + c_2 \cdot s_t^{n'} \cdot (G_{t-1}^{n'} - X_{t-1}^{n'}) - G_{t-1}^{n'}\} +} \\ &\qquad \overline{\cdots + G_{t-1}^{1} - X_{t-1}^{n'} - \chi \cdot V_{t-1}^{n'} - c_2 \cdot s_t^{n'} \cdot (G_{t-1}^{n'} - X_{t-1}^{n'})} \\ &= \frac{a \cdot V_{t-1}^{n'} + G_{t-1}^{1} - X_{t-1}^{n'}}{(a-1) \cdot (\chi \cdot V_{t-1}^{n'} + c_2 \cdot s_t^{n'} \cdot (G_{t-1}^{n'} - X_{t-1}^{n'})) + G_{t-1}^{1} - X_{t-1}^{n'} +} \\ &\qquad \overline{\cdots + N \cdot \max\{0, X_{t-1}^{n'} + \chi \cdot V_{t-1}^{n'} + c_2 \cdot s_t^{n'} \cdot (G_{t-1}^{n'} - X_{t-1}^{n'}) - G_{t-1}^{n'}\}} \end{aligned}$$

There are two distinct cases. In the first case, the position of particle $n'$ before its step is equal to the global attractor, in terms $G^{n'}_{t-1} = X^{n'}_{t-1}$, which in particular implies that $G^1_{t-1} = X^{n'}_{t-1}$ since $X^{n'}_{t-1} \leq G^1_{t-1} \leq G^{n'}_{t-1}$. Then its move is deterministic and its new position will be the new global attractor. In this case, we obtain:

$$\begin{aligned} \mathrm{E}\left[\sqrt{\chi_{n'}} \mid \mathcal{F}_{t-1}\right] &\leq \sqrt{\frac{a \cdot V^{n'}_{t-1}}{(a-1)\cdot\chi\cdot V^{n'}_{t-1} + N\cdot\chi\cdot V^{n'}_{t-1}}} \\ &= \sqrt{\frac{a}{(a-1)\cdot\chi + N\cdot\chi}}, \end{aligned}$$

which is less than $q$ if and only if $a < (N-1)\cdot\frac{\chi\cdot q^2}{1-\chi\cdot q^2}$.

The second case when $G^1_{t-1} > X^1_{t-1}$ requires more exhaustive but still straight-forward calculations. The expression

$$\begin{aligned} &\sup_{v>0, g\geq d>0} \int_0^1 \sqrt{\frac{av+d}{(a-1)(\chi v + c_2 s g) + N\max\{0, \chi v + c_2 s g - g\} + d}}\,ds \\ &= \sup_{v'>0, g'>1} \int_0^1 \sqrt{\frac{av'+1}{(a-1)(\chi v' + c_2 s g') + N\max\{0, \chi v' + c_2 s g' - g'\} + 1}}\,ds \end{aligned}$$

needs to be bounded.

The calculation of the integral can be done explicitly and finding the values for $v'$ and $g'$ maximizing it for given $a$, $\chi$, $c_2$ and $N$ using standard techniques from analysis is straight-forward. Obviously, the greater the number of particles is, the smaller is the value of the integral, so we calculate the minimal number of necessary particles ensuring that the integral is less than 1 for three common parameter choices obtained from the literature. For the choice of $\chi = 0.72984$, $c_2 = 1.496172$ recommended in [CK02, BK07], we choose $a := 2.3543$ and obtain for $N = 2$ an upper bound of $q \leq 0.9812$. For the choice $\chi = 0.72984$ and $c_2 = 0.94879$ ([CD01]), we require at least $N = 3$ particles and the choice of $a := 5.1298$ leads to $q \leq 0.9964$. Finally, for the choice $\chi = 0.6$, $c_2 = 1.7$ as proposed in ([Tre03]), for $N = 3$ and $a := 2.5847$ we obtain a value of $q \leq 0.9693$. □

In Figure 4.5, we can see the borderlines between choices for $c_2$ and $\chi$ that satisfy the conditions of Lemma 4.1 and those that do not. We will refer to the parameters that satisfy both Lemma 4.1 and the convergence requirements from Theorem 3.11 as "good" parameters. They are a counterpart to

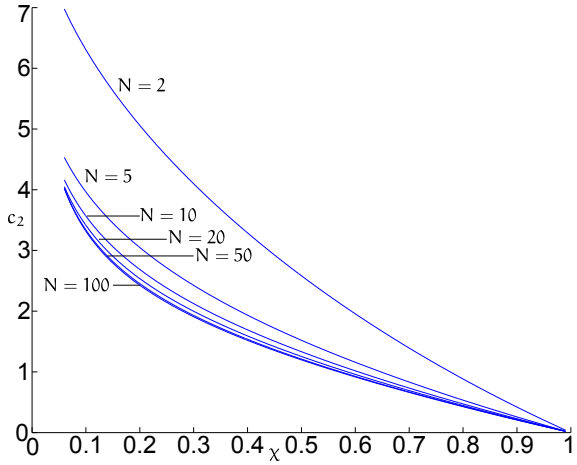



**Figure 4.5:** Borders between the too low values for $c_2$ and $\chi$ and the ones large enough to satisfy the requirements in Lemma 4.1 for some swarm sizes $N$.

the parameters satisfying the conditions of [LW11] from Section 3.5.1 in the previous chapter, which can be seen as "bad" parameters that allow stagnation on arbitrary search points with positive probability. However, there are still parameters that satisfy neither the conditions from Section 3.5.1 nor the conditions here, so they are neither good nor bad. From here on, we assume that the parameters are good.

Lemma 4.1 says that, given the parameters are good, a swarm that moves into the right direction can overcome every distance and increase its potential, no matter how small it was in the beginning. In other words: The equilibrium, when all attractors and particles are on the same point and every velocity is zero, is not stable because arbitrary small changes of an attractor, a position or a velocity can be sufficient to lead the swarm far away from this equilibrium, as long as there is a direction with decreasing value of the objective function. This is already sufficient to prove the main result of this section.

**Theorem 4.1.** If $D = 1$, then every accumulation point of the sequence of global attractors $G = (G_t^n)_{n=1,\dots,N;t\in\mathbb{N}}$ is a local minimum of $f$ almost surely.

*Proof.* Assume for contradiction, that there is an accumulation point $z \in \mathbb{R}$ such that (w. l. o. g.) $f$ is monotonically decreasing on $B_\tau(z) = (z-\tau, z+\tau)$ for some $\tau > 0$. Since $z$ is an accumulation point, for every $\varepsilon > 0$ $G$ is inside the

$\varepsilon$-neighborhood $B_\varepsilon(z) = (z-\varepsilon, z+\varepsilon)$ of $z$ infinitely often. Figure 4.6 gives a visualization of the described situation. Note that $G_t^n$ entering $[z, z+\tau]$ for any $t$ and any $n$ violates the assumption of $z$ being an accumulation point. That is because the global attractor does not accept worsenings, so for any $\varepsilon$ with $\varepsilon < |z - G_t^n|$, $B_\varepsilon(z)$ will not be entered by the global attractor anymore.

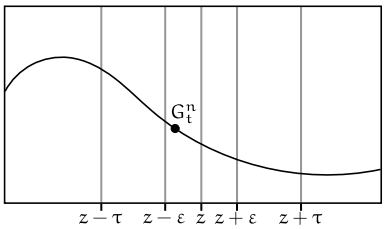



**Figure 4.6:** Objective function $f$ monotonic on $B_\tau(z)$, global attractor in $\varepsilon$-neighborhood of $z$

Now two cases need to be considered. The first case is that there is at least a second accumulation point (that might even be a local optimum). This second accumulation point must yield the same function value as $z$ and can therefore not be inside $B_\tau(z)$. Consequently, infinitely often, a particle has a distance of at least $\tau$ between its local and its global attractor. From this situation, the probability for hitting $[z, z+\tau]$ within the next few steps is positive and the probability for never hitting $[z, z+\tau]$ would be 0. To prove this, it is sufficient to outline a sequence that moves the particle into $[z, z+\tau]$.

Several different situations could occur: There are essentially three possible orders of the particle, its local attractor and $z$, depending on which of this three points is between the other two. Furthermore, the velocity could be close to 0 or large and it could point in two different directions. Since the calculations showing that hitting $[z, z+\tau]$ has a positive probability from each of the situations are straight-forward, we present them exemplary only. Consider the case where $z$ is located between the particles position $X_t^n$ and its local attractor $L_t^n$ and has a velocity $V_t^n$ with $|V_t^n| \leq (c_1 \cdot |L_t^n - X_t^n| - |z - X_t^n| - \tau)/\chi$ pointing away from $L_t^n$. Then, any choice of $r_t^n \in I$ with

$$I = \left[ \frac{|z - X_t^n| + \tau/4 - \chi \cdot |V_t^n|}{c_1 \cdot |L_t^n - X_t^n|}, \frac{|z - X_t^n| + 3/4 \cdot \tau - \chi \cdot |V_t^n|}{c_1 \cdot |L_t^n - X_t^n|} \right] \cap [0, 1]$$

leads to $X_{t+1}^n \in [z, z+\tau]$. Since

$$|z - X_t^n| + \tau/4 - \chi \cdot |V_t^n| \leq |z - X_t^n| + \tau/4 \leq c_1 \cdot |L_t^n - X_t^n| - 3/4 \cdot \tau$$

and

$$|z - X_t^n| + 3/4 \cdot \tau - \chi \cdot |V_t^n| \geq c_1 \cdot |L_t^n - X_t^n| - \tau/4,$$

we have $|I| \geq \min\{1, \frac{\tau}{4 \cdot c_1 \cdot |L_t^n - X_t^n|}\}$. If $|V_t^n|$ is larger than assumed above, we require at most two additional steps in order to decrease it sufficiently. Similar calculations for all the other cases show that there is indeed always a positive probability to hit $[z, z+\tau]$ within a constant number of iterations. Additionally, since $|L_t^n - X_t^n|$ is stochastically bounded due to the choice of admissible objective functions, the probability can not converge to 0. It follows that the probability to never hit $[z, z+\tau]$ in case of a second accumulation point is 0.

If $z$ is the only accumulation point, the attractors converge towards $z$. As a consequence of the results in [JLY07a], this implies that the whole swarm converges to $z$. That is the point were Lemma 4.1 becomes useful. Since $f$ is monotonic on $B_\tau(z)$, the local and the global attractor are always greater or equal to the current position of the particle. Therefore the velocities will with probability 1 all become positive after a finite number of iterations and stay positive. It follows that each particle will exceed its local attractor almost surely after a finite number of iterations. At that time, the swarm becomes positively running at least until the first particle surpasses a local minimum and therefore leaves $B_\tau(z)$. With Lemma 4.1, this will happen after a finite number of iterations almost surely, a contradiction to the convergence of the swarm towards $z$.

So, $z$ is no accumulation point of $G_t^n$. □

Although this theorem does not state that the swarm converges at all, it is easy to derive a corollary about convergence from it by either only taking the sequence $(f(G_t^n))_{t \in \mathbb{N}}$ into account or adding more restrictions to the set of admissible objective functions.

**Corollary 4.1.** If $D = 1$, then $f(G_t^n)$ converges towards the value of a local minimum. Particularly, if no two local minima have the same value, then $G_t^n$ converges towards a local minimum. If the swarm converges towards a point $z \in \mathbb{R}$, then $z$ is a local minimum.

*Proof.* The first statement follows directly from Theorem 4.1. From Definition 4.1, it follows that the sequence of the global attractors over the time is bounded and therefore has at least one accumulation point. If there is more than one accumulation point, then $f$ has the same value on each of them because $f$ is continuous. Due to Theorem 4.1, every accumulation point is a local minimum, so if there are no two local minima with the same value,

then there is only one accumulation point that therefore is the limit of $G_t^n$. That proves the second statement. The third statement again is a direct consequence of Theorem 4.1 because convergence of the swarm implies convergence of $G_t^n$. □

### 4.1.2 Experimental Setup

In order to supplement the theoretical findings, we present a number of experimental results. Since the model of the PSO algorithm is formulated in terms of real numbers, standard double precision machine numbers turned out to be insufficient to capture all the interesting phenomena. If the number of iterations is large, certain values leave the range of double precision numbers and become either zero or infinity. For example, if the position of a particle has an absolute value which is, say, $10^{20}$ times larger than its velocity, then the sum of the old position and the new velocity exactly equals the old position and the particle does not move at all. In order to avoid such artifacts of machine numbers, an implementation based on the mpf_t data type of the GNU Multiple Precision Arithmetic Library ([mpfl4]) is used, which was developed as a part of [Raß14]. Initially, the implementation starts with a precision of 2.000 binary digits and increases the precision when necessary. More precisely: On every addition and subtraction, a test whether the current precision needs to be increased is performed.

Most of the observed curves behave either exponentially decreasing or exponentially increasing, such that an arithmetic mean of several test runs would reflect more or less only the one test run which obtained the largest value. Therefore, whenever nothing else is stated, every presented data point stands for the *geometric mean* of 1.000 test runs. Note that the geometric mean of some data set $x_1, \ldots, x_k$ can be formulated as $2^{\overline{m}}$, where $\overline{m}$ is the arithmetic mean of $\log_2 x_1, \ldots, \log_2 x_k$. Therefore, it is a good measure for the average behavior of some process which decreases or increases exponentially.

Whenever in experiments we measure the potential via $\Phi_t$ as defined in Definition 3.9, we set the parameter $\alpha$ to 1 as long as nothing else is stated.

### 4.1.3 Experimental Results on the Potential gain

We examine the behavior of a 1-dimensional PSO with respect to the potential experimentally. If the swarm is close to a local optimum and there is no second local optimum within range, the attractors converge and it is well-known that with appropriate choices for the parameters of PSO, convergence of the attractors implies convergence of the whole swarm. Such parameter selection guidelines can be found, e. g., in [JLY07a].

On the other hand, if the swarm is far away from the next local optimum and the objective function is monotone on an area that is large compared to the current potential of the swarm, the preferred behavior of the swarm is to increase the potential and to move in the direction that yields the improvement until a local optimum is surpassed and the monotonicity of the function changes. As pointed out in Section 3.5.1, the authors of [LW11] show that there are non-trivial choices of parameters for which the swarm converges even on a monotone function. In particular, if $N = 1$, every parameter choice either allows convergence to an arbitrary point in the search space, or it generally prevents the one-particle-swarm from converging, even if the global attractor is already at the global optimum. Therefore, the minimum size of a swarm with optimizing behavior is $N = 2$.

In order to measure the increase of potential, the particle swarm algorithm is run on a monotone function to measure the course of the potential over time.

**Experiment 4.1.** We choose the 1-dimensional function $f(x) = -x$ as objective function, wanting the swarm always "running down the hill." Note that this choice is not a restriction, since the particles compare points only qualitatively and the behavior is exactly the same on any monotone decreasing function: The new attractors are the points with greater $x$-coordinate. Therefore, we use only one function in the experiment. The parameters for the movement equations are common choices obtained from the literature. The number of iterations is set to 2.000 iterations and the potential (measured as $\Phi$ according to Definition 3.9 with $\alpha$ set to 1) is stored at every iteration.

The geometric means for different configurations of the swarm parameters $N$, $\chi$, $c_1$ and $c_2$ are presented in Figure 4.7.

As stated in the proof of Lemma 4.1, the increase of the potential of a running particle swarm is expected to be exponential in the number of steps the swarm makes. The cases (a), (c) and (e) are covered by the analysis and show

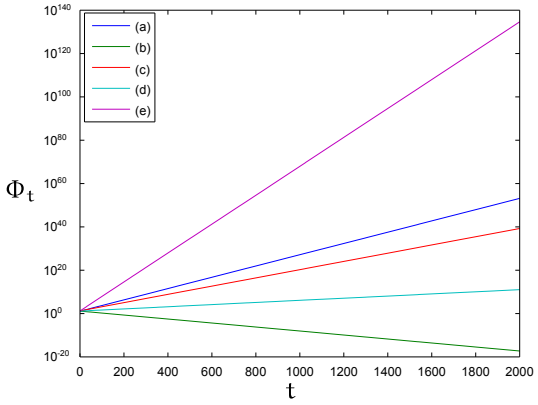



**Figure 4.7:** Course of potential during 2000 Iterations for different parameter sets.
(a) $\chi = 0.72984$, $c_1 = c_2 = 1.49617$, $N = 2$ [CK02]
(b) $\chi = 0.72984$, $c_1 = 2.04355$, $c_2 = 0.94879$, $N = 2$ [CD01]
(c) $\chi = 0.72984$, $c_1 = 2.04355$, $c_2 = 0.94879$, $N = 3$ [CD01]
(d) $\chi = 0.6$, $c_1 = c_2 = 1.7$, $N = 2$ [Tre03]
(e) $\chi = 0.6$, $c_1 = c_2 = 1.7$, $N = 3$ [Tre03]

the expected behavior as an exponential increase of the potential. Cases (b) and (d) are not covered by the analysis, here the number of particles is below the bound that is proved to be sufficient. However, case (d) still works, even though the increase of the potential is much smaller than in the provably good cases (a), (c) and (e). Only in case (b) where only two particles are involved, we can see the potential decreasing exponentially because the number of particles is presumably too small. In this case, the swarm will eventually stop, i. e., stagnate. But we also see in case (c) that using one additional particle and not changing the remaining parameters enables the swarm to keep its motion.

In all cases, for the small swarm size of $N \geq 3$, the common parameter choices avoid the problem mentioned in [LW11].

## 4.2 Proof of Linear Convergence Time

After having the proof that in the 1-dimensional case PSO finds the local optimum, it follows in particular convergence towards the global optimum if the objective function $f$ is unimodal. At that point, the question of the runtime arises.

In this section, a method for formally proving runtime results is introduced. This method is based on drift theory and will make use of Theorem 3.7, the multiplicative drift theorem for continuous search spaces we proposed in the previous chapter. In order to apply it, a measure $\Psi$ is needed that measures the progress of the particle swarm and the distance to the desired state of the swarm, i. e., the smaller $\Psi$ is, the closer is the swarm to the state when all particles stand on the optimum and all velocities are $0$. The crucial part of the proposed technique is the proper choice of $\Psi$. The distance measure $\Psi$ must not only depend of the global attractor alone, because it will turn out that the swarm can encounter situations when the chance to improve the global attractor within the near future is very small. Therefore, the analysis needs to identify such barriers, i. e., situations which do not allow sufficient improvements of the global attractor, in the following called bad events, and to take into account the self-healing property of the particle swarm, i. e., its ability to recover from such bad events.

As a result, we obtain that the swarm converges linearly towards the optimum, where linear convergence rate means that the time for halving the distance to the limit is constant or equivalently, the number of digits, in which the current value and the limit agree, increases linearly over time.

For the rest of this section, we consider only 1-dimensional, unimodal functions. Since PSO is invariant under translations, we assume without loss of generality that the unique optimum is at $0$.

### 4.2.1 Measuring the Distance to Optimality

In this section, we introduce the proposed distance measure for analyzing the runtime of the classical, 1-dimensional PSO and establish the necessary drift condition to apply Theorem 3.7.

The measure will be composed of two different kinds of components. There will be a so-called *primary measure* $\Psi_t^{(0)}$, measuring what is recognized as ac-

tual progress of the algorithm. The primary measure should be non-increasing and 0 if and only if the swarm has converged towards an optimal point. As it turns out, the natural candidates for a primary measure, namely $|G_t^1|$ and $f(G_t^1)$, are both bad choices. As for $|G_t^1|$, since $f$ is not assumed to be symmetric, we can not assume that points closer to 0 are always better. The choice $f(G_t^1)$ does not yield the same weakness, i, e., it preserves the order between the points with respect of their quality, but the amount by which $f(G_t^1)$ decreases when the global attractor is updated depends too much on the unknown objective function $f$. So, the measure $f(G_t^1)$ is very hard to handle. These considerations show that measures in terms of the size of the area that yields an improvement, are better choices.

**Definition 4.4.** For a fixed search space $S$ and a point $x \in S$, $A(x)$ denotes the area inside $S$ of search points that are at least as good as $x$ with respect to the objective function $f$. In terms:

$$A(x) := \{y \in S \mid f(y) \leq f(x)\}.$$

A possible candidate for the primary measure $\Psi_t^{(0)}$ is $|A(G_t^1)|$, but for technical reasons, we measure not only the quality of the global attractor, but also the quality of every single local attractor. The exact choice of our primary measure will be stated in Definition 4.5 below.

However, for the primary measure alone, we cannot expect any drift condition to hold. In order to see this, consider, e. g., the following pathological situations: If the global attractor is already close to the optimum, but the squared potential of the swarm is orders of magnitudes higher than the remaining distance to the optimum, then the probability for an update of the global attractor is (arbitrarily) close to 0. Therefore, we cannot hope for the drift condition (3.11) of Theorem 3.7 with a constant $\delta > 0$ to hold. Another such pathological situation occurs when the swarm potential is much smaller than the difference between global attractor and optimum. In that situation, updates will happen frequently but the updated position will be very close to the original one, so the overall progress is still small.

In order to handle situations that prevent the swarm from improving its primary measure, additional measures called *secondary measures* $\Psi_t^{(i)}$ are used, measuring the badness of a configuration, where "bad" means that the particles are prevented from performing significant improvements of their primary measure. In order to prove linear convergence time, the first step is to find secondary measures $\Psi_t^{(i)}$ and events $B_t^{(i)}$, the bad events, with $B_t^{(i)} \cap B_t^{(j)} = \emptyset$ for $i \neq j$, such that the following conditions hold:

1. The secondary measures do not worsen unboundedly, i. e., $E[\sum_i \Psi_{t+1}^{(i)} \mid \mathcal{F}_t] \leq C \cdot (\Psi_t^{(0)} + \Psi_t^{(i)})$ with a constant $C > 0$,
2. In the situation $B_t^{(i)}$, the particle swarm can heal itself, i. e., $E[\Psi_{t+1}^{(i)} \mid \mathcal{F}_t, B_t^{(i)}] \leq (1 - \delta_i) \cdot \Psi_t^{(i)}$ with a constant $\delta_i > 0$,
3. If $B_t^{(i)}$ holds, the other secondary measures are much smaller than $\Psi_t^{(i)}$, i. e., $B_t^{(i)} \Rightarrow \forall j \neq i : \Psi_t^{(j)} \ll \Psi_t^{(i)}$,
4. If $B_t^{(i)}$ holds, $\Psi_t^{(i)}$ has at least a constant fraction of the primary measure $\Psi_t^{(0)}$, i. e., $B_t^{(i)} \Rightarrow \Psi_t^{(i)} \geq \Psi_t^{(0)}/d_i$,
5. If none of the bad events holds, the primary measure fulfills the drift condition, i. e., $\bigcap_i \overline{B}_t^{(i)} \Rightarrow E[\Psi_{t+1}^{(0)} \mid \mathcal{F}_t, B_t^{(0)}] \leq (1 - \delta_0) \cdot \Psi_t^{(0)}$ with a constant $\delta_0 > 0$,
6. If none of the bad events holds, the primary measure has at least a constant fraction of every secondary measure, i. e., $\bigcap_i \overline{B}_t^{(i)} \Rightarrow \forall i : \Psi_t^{(i)} \leq D_i \cdot \Psi_t^{(0)}$,
7. At the beginning, the secondary measures have a finite expectation, i. e., the particles are initialized such that $E[\Psi_0^{(i)}] < \infty$,
8. A swarm converging towards the optimum implies that every secondary measure converges towards 0.

If we can find a set of secondary measures and corresponding bad events, such that the conditions mentioned above are satisfied, the actual optimality measure results as a weighted sum of the primary and all secondary measures and has the form

$$\Psi_t := C_\Psi \cdot \Psi_t^{(0)} + \sum_i \Psi_t^{(i)}$$

with some constant $C_\Psi > 0$ that emphasizes the influence of the primary measure. If at some time $t$ the event $B_t^{(i)}$ holds, 2. guarantees an expected decrease of $\Psi_t^{(i)}$. At the same time, the other secondary measures might increase, but because of 1. their increase is bounded by a constant times their own value plus the value of the primary measure. Because of 3. and with $C_\Psi$ sufficiently large, this value is insignificant compared to $\Psi_t$, so the sum over

all secondary measures still decreases. With 4 and the fact that $\Psi^{(0)}$ is non-increasing, it follows that the decrease of the secondary measures leads to a noticeable decrease of $\Psi_t$.

If at time $t$ no $B_t^{(i)}$ holds, 5. implies that the primary measure decreases by a constant factor. However, the secondary measures might be by at most a constant factor larger than $\Psi_t^{(0)}$ and they might increase by at most a constant factor. In order to avoid an increase of $\Psi$, the influence of the primary measure is strengthened by multiplying it with a constant $C_\Psi$, sufficiently large to guarantee that the decrease of $C_\Psi \cdot \Psi_t^{(0)}$ is noticeable larger than the largest possible increase of $\sum_i \Psi_t^{(i)}$.

Note that instead of the drift conditions in 2. and 5., weaker requirements are still sufficient, i. e., instead of insisting on an expected decrease within the next step, an expected decrease within a properly chosen, $\mathcal{F}_t$-measurable time $\sigma(t)$ is sufficient. Condition 7. is typically easy to verify and 8. follows from the others and is only stated as an additional hint when searching for a good set of secondary measures. But verifying 1.-6. requires more effort and a lot of technical calculations.

For the 1-dimensional setting, we specify the primary and secondary measures as follows:

**Definition 4.5** (Distance measure)**.** For some constant $C_\Psi$ to be fixed later, the optimality measure $\Psi = (\Psi_0, \Psi_1, \dots)$ is defined as

$$\Psi_t := C_\Psi \cdot \sum_{n=1}^{N} \sqrt{|A(L_t^n)|} + \sum_{n=1}^{N} Y_t^n + \frac{|A(G_t^1)|}{\Phi_{t+1}^1},$$

with

$$Y_t^n = \sqrt{|V_t^n|} + \sqrt{|G_t^1 - X_t^n|}$$

and

$$\Phi_{t+1}^1 = \sqrt{\sum_{n=1}^{N} \left( a \cdot |V_t^n| + |G_t^1 - X_t^n| \right)}$$

as already defined in Definition 3.9. I. e., the primary measure is

$$\Psi_t^{(0)} := \sum_{n=1}^{N} \sqrt{|A(L_t^n)|}$$

and the secondary measures are

$$\Psi_t^{(1)} := \Psi_t^H := \sum_{n=1}^{N} Y_t^n$$

and

$$\Psi_t^{(2)} := \Psi_t^L := \frac{|A(G_t^1)|}{\Phi_{t+1}^1}.$$

The associated bad events are

$$B_t^{(1)} := B_t^H := \left\{ \sum_{n=1}^{N} Y_t^n \geq c_H \cdot \sum_{n=1}^{N} \sqrt{|A(L_t^n)|} \right\}$$

and

$$B_t^{(2)} := B_t^L := \left\{ \frac{|A(G_t^1)|}{\sum_{n=1}^{N} Y_t^n} \geq c_L \cdot \sum_{n=1}^{N} (Y_t^n + \sqrt{|A(L_t^n)|}) \right\}$$

with two constants $c_H$ and $c_L$ to be fixed later.

The occurring square roots have rather technical reasons. Informally, we think of $B_t^H$ as the event that at time $t$ the potential is too high, so the probability for improving the primary measure at all is small. Again due to technical reasons, $\Psi_t^H$ measures the potential in a different way than just $\Psi_{t+1}^1$. Similarly, $B_t^L$ describes the event of a too small potential at time $t$, so the updates cannot reduce the primary measure significantly. The parameters $c_H$ and $c_L$ quantify the "badness" of the respective situation and are used as control parameters in 3., i. e., as will become clear soon, the larger $c_H$ is chosen, the smaller will $\Psi_t^L$ be whenever $B_t^H$ holds. Similarly, the larger $c_L$ is chosen, the smaller will $\Psi_t^H$ be whenever $B_t^L$ holds.

In the following section, we will verify that the optimality measure, as defined in Definition 4.5, indeed satisfies all the requirements and therefore is the tool of choice to prove the expected linear convergence time of the PSO algorithm.

For the rest of this chapter, we will use the convention to write const for a positive constant, depending only on the swarm parameters $\chi$, $c_1$, $c_2$ and $N$ and sometimes on the potential parameter $a$, where any two occurrences of const do not necessarily refer to the same constant. Additionally, if we want to emphasize the dependencies of a const, we write, e. g., $\text{const}(\chi, c_1, c_2)$.

### 4.2.2 Lower Bounds for the Decrease of the Distance Measure

We need to examine the three measures $\Psi_t^H$, $\Psi_t^L$ and $\Psi_t^{(0)}$ in order to verify that they decrease sufficiently fast whenever the according event occurs and that they never increase too fast. While the general bounds on the increase of the according measures are in most cases intuitively clear, it is not obvious that, e. g., in case of a very high potential, $\Psi_t^H$ reduces sufficiently fast. Therefore, before we present the formal proofs, we perform experiments to illustrate that the stated measures are indeed a good choice and yield the desired behavior.

#### High Potential

The first case is that the swarm potential is too large, i. e., much larger than the areas that yield an improvement for either the global or the local attractor. In Section 3.5.2, a convergence analysis was presented, showing that under the assumption of constant and identical attractors, the potential of the swarm converges with linear convergence speed towards 0. To verify that this result is stable, i. e., that the potential still decreases sufficiently fast if the attractors are not constant but can move only inside an area which is small compared to the swarm potential, we perform an experiment to see how the swarm behaves when its attractors are very close to the optimum while the potential is comparatively high.

In order to study the behavior of a swarm with a too high potential, we choose two objective functions, namely Sphere, defined as

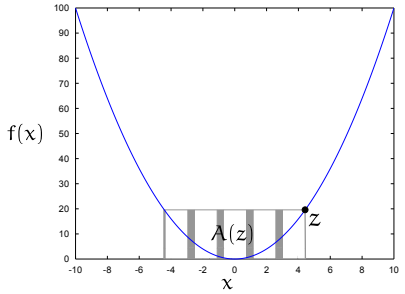
$$\text{Sphere}(x) = x^2,$$

and Sphere$^+$ ([LW11]), defined as

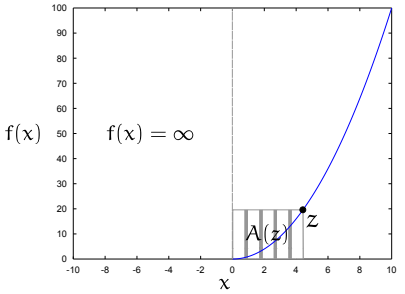
$$\text{Sphere}^+(x) = \begin{cases} \text{Sphere}(x), & \text{if } x \geq 0, \\ \infty, & \text{otherwise.} \end{cases}$$

The functions can be seen in Figure 4.8a and Figure 4.8b.

In case of Sphere, $A(z)$ is for every $z \in \mathbb{R}$ a symmetric interval around the optimum 0, while for the function Sphere$^+$, the optimum is at the upper bound of the interval $A(z)$ for every $z \in \mathbb{R}, z \geq 0$.




**(a)** 1-dimensional function Sphere.



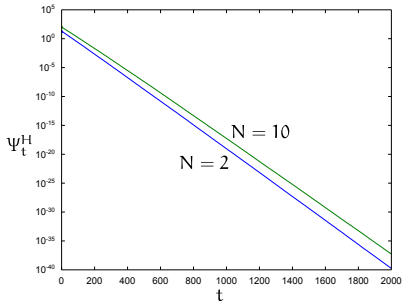

**(b)** 1-dimensional function Sphere$^+$.

**Figure 4.8:** The symmetric function Sphere and the asymmetric function Sphere$^+$.

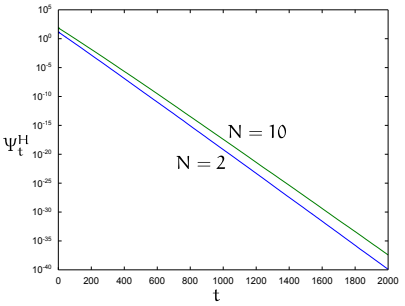
**Experiment 4.2.** For each of the two objective functions, we use the swarm sizes $N = 2$ and $N = 10$. We initialize the velocities uniformly at random over $[-100, 100]$ and choose the particles' starting positions from the interval $[-10^{-100}, 10^{-100}]$ in case of Sphere and $[0, 10^{-100}]$ in case of Sphere$^+$. I. e., the distance to the optimum is smaller than $10^{-100}$, while the velocities are of order 100 and therefore the potential is comparatively high. Figure 4.9 shows the obtained values for $\Psi_t^H$. We can see that indeed the potential of the swarm increases with a speed neither depending much on the objective function nor on the number of particles. Since the updates of attractors are the only way the swarm reacts on the objective function, and since in the configuration with a too high potential attractor updates happen too seldom to have high influence on the swarm's movement, the behavior is the same for both (and any other) objective functions.

Regarding the influence of the swarm size, the swarm with only two particles apparently decreases its potential slightly faster than the swarm with 10 particles. The reason for this is that the measure $\Psi_t^H$ sums up all the potentials of the single particles and is therefore only a constant factor smaller than the maximum potential over all particles. Since the particles move (almost) independently of each other, $\Psi_t^H$ behaves similar to the maximum of N independent random variables and therefore increases with N even if the distribution of the particles' potentials itself remains unchanged.

Indeed, the analysis from Section 3.5.2 can be extended to cover the case where the attractors are not constant but can move not further than a distance that is small in comparison to the current potential. We formally prove




**(a)** Particles processing SPHERE after being initialized with a too high potential.




**(b)** Particles processing SPHERE$^+$ after being initialized with a too high potential.

**Figure 4.9:** Particle swarm suffering from too high potential while processing 10-dimensional objective functions SPHERE and SPHERE$^+$ with $N = 2$ or $N = 10$ particles, initialized with 1 or $D/2$ dimensions with too high potential.

the observation from Experiment 4.2 in the following lemma, which verifies 1.-4. for $B_t^H$.

**Lemma 4.2.** There are constants $t'_H \in \mathbb{N}$, $c'_H > 0$, $C_H > 0$, $\delta_H \in (0,1)$ and $\text{const}_H$, depending only on $c_1$, $c_2$, $\chi$ and $N$, such that for every $t_0 \in \mathbb{N}$, every $t_H \geq t_0 + t'_H$ and every $c_H > c'_H$, we have

$$\mathrm{E}\left[\sum_{n=1}^{N} Y_{t_H}^n \;\middle|\; \mathcal{F}_{t_0}\right] \leq C_H \cdot \left(\sum_{n=1}^{N} Y_{t_0}^n + \sum_{n=1}^{N} \sqrt{|A(L_{t_0}^n)|}\right), \tag{4.2}$$

$$\mathrm{E}\left[\sum_{n=1}^{N} Y_{t_H}^n \;\middle|\; \mathcal{F}_{t_0}, B_{t_0}^H\right] \leq (1-\delta_H) \cdot \sum_{n=1}^{N} Y_{t_0}^n, \tag{4.3}$$

$$B_{t_0}^H \text{ implies } |A(G_{t_0+1}^1)|/\Phi_{t_0+1}^1 \leq \text{const}_H \cdot \Psi_{t_0}/c_H, \tag{4.4}$$

$$B_{t_0}^H \text{ implies } \sum_{n=1}^{N} Y_{t_0}^n \geq \Psi_{t_0} \Big/ \left(1 + \frac{C_\Psi + \text{const}_H}{c_H}\right). \tag{4.5}$$

*Proof.* To shorten notation, we write $\mathrm{E}_{t_0}[\circ]$ is $\mathrm{E}[\circ \mid \mathcal{F}_{t_0}]$, analogous $\mathrm{Var}_{t_0}[\circ]$ and $\mathrm{Cov}_{t_0}[\circ]$. In order to use the results from Section 3.5.2, we first concen-

trate only on a single particle $n$ and express $E_{t_0}[Y^n_{t_H}]$ in terms of expectation and variance without absolute value inside:

$$\begin{aligned} & E_{t_0}\left[\sqrt{|V^n_{t_H}|} + \sqrt{|G^1_{t_H} - X^n_{t_H}|}\right] \\ \leq\ & \sqrt{E_{t_0}[|V^n_{t_H}|]} + \sqrt{E_{t_0}[|G^1_{t_H} - X^n_{t_H}|]} \\ \leq\ & \sqrt{|E_{t_0}[V^n_{t_H}]|} + \sqrt{\mathrm{Var}_{t_0}[V^n_{t_H}]} \\ & + \sqrt{|E_{t_0}[G^1_{t_H+1} - X^n_{t_H}]|} + \sqrt{\mathrm{Var}_{t_0}[G^1_{t_H} - X^n_{t_H}]}, \end{aligned} \tag{4.6}$$

where for the first inequality we applied Jensen's inequality and the second inequality follows from

$$\begin{aligned} E[|Z|] &\leq E[|Z - E[Z]| + |E[Z]|] \\ &\leq \sqrt{E[(Z - |E[Z]|)^2]} + |E[Z]| = \sqrt{\mathrm{Var}[Z]} + |E[Z]|, \end{aligned}$$

which follows from the triangle inequality and the generalized mean inequality.

It remains to bound the expectation and the variance of the velocity and the distance to the global attractor. The same task was done by Jiang et al. in [JLY07a] under the assumption of constant local and global attractors. In this situation here, the attractors are not constant but their movement is limited: The local attractor of particle $n$ can not leave $A(L^n_t)$ and the global attractor is as least as good as any local, so it will also stay inside every $A(L^n_t)$. With this observation, we can modify the analysis from Section 3.5.2 as follows.

From the movement equations, we obtain for every $t \in \mathbb{N}$.

$$\begin{pmatrix} X^n_{t+1} \\ X^n_t \end{pmatrix} = \underbrace{\begin{pmatrix} 1 + \chi - \frac{c_1+c_2}{2} & -\chi \\ 1 & 0 \end{pmatrix}}_{=:A} \cdot \begin{pmatrix} X^n_t \\ X^n_{t-1} \end{pmatrix} + \begin{pmatrix} c_1 \cdot r^n_t \cdot L^n_t + c_2 \cdot s^n_t \cdot G^n_t \\ 0 \end{pmatrix}$$

By iterating, we get

$$\begin{pmatrix} X^n_{t_H} \\ X^n_{t_H-1} \end{pmatrix} = A^{t_H - t_0} \cdot \begin{pmatrix} X^n_{t_0} \\ X^n_{t_0-1} \end{pmatrix} + \sum_{t=0}^{t_H-t_0-1} A^i \cdot \begin{pmatrix} c_1 \cdot r^n_{t_0+t} \cdot L^n_{t_0+t} + c_2 \cdot s^n_{t_0+t} \cdot G^n_{t_0+t} \\ 0 \end{pmatrix}$$

and therefore

$$\begin{aligned}
&\left\| \begin{pmatrix} E_{t_0}[X^n_{t_H}] \\ E_{t_0}[X^n_{t_H-1}] \end{pmatrix} \right\|_1 \\
&= \left\| A^{t_H-t_0} \cdot \begin{pmatrix} X^n_{t_0} \\ X^n_{t_0-1} \end{pmatrix} + \frac{1}{2} \cdot \sum_{t=0}^{t_h-t_0-1} A^t \cdot \begin{pmatrix} E_{t_0}[c_1 \cdot L^n_{t_0+t} + c_2 \cdot G^n_{t_0+t}] \\ 0 \end{pmatrix} \right\|_1 \\
&\leq \|Q\|_1 \cdot |\lambda_{max}|^{t_h-t_0} \cdot \|Q^{-1}\|_1 \left\| \begin{pmatrix} X^n_t \\ X^n_{t_0-1} \end{pmatrix} \right\|_1 \\
&\quad + \frac{1}{2} \cdot \|Q\|_1 \cdot \frac{1}{1-|\lambda_{max}|} \cdot \|Q^{-1}\|_1 \cdot (c_1+c_2) \cdot |A(L^n_{t_0})|,
\end{aligned}$$

where $Q$ is an invertible matrix such that $A = Q \cdot D \cdot Q^{-1}$ for some diagonal matrix $D$ and $\lambda_{max}$ is an eigenvalue of $A$ with largest absolute value. Note that the parameter guidelines for $\chi$, $c_1$ and $c_2$ suggested in [JLY07a] guarantee $|\lambda_{max}| < 1$. Since $X^n_{t-1} = X^n_t - V^n_t$, we have

$$\begin{aligned}
&|E_{t_0}[X^n_{t_H}]| + |E_{t_0}[V^n_{t_H}]| \\
&\leq 2 \cdot (|E_{t_0}[X^n_{t_H}]| + |E_{t_0}[X^n_{t_H} - V^n_{t_H}]|) \\
&\leq \frac{1}{2} \cdot c(\chi, c_1, c_2) \cdot (|\lambda_{max}|^{t_H-t_0} \cdot (|X^n_{t_0}| + |X^n_{t_0} - V^n_{t_0}|) + |A(L^n_{t_0})|) \\
&\leq c(\chi, c_1, c_2) \cdot (|\lambda_{max}|^{t_H-t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|) \qquad (4.7)
\end{aligned}$$

with a constant $c(\chi, c_1, c_2)$, depending only on $\chi$, $c_1$ and $c_2$.

Analogous calculations show that also for the variance

$$\begin{aligned}
&\sqrt{Var_{t_0}[X^n_{t_H}]} + \sqrt{Var_{t_0}[V^n_{t_H}]} \\
&\quad \leq c'(\chi, c_1, c_2) \cdot (|\lambda'_{max}|^{t_H-t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|)
\end{aligned} \qquad (4.8)$$

holds for some $\lambda'_{max}$ with $|\lambda'_{max}| < 1$ and some constant $c'(\chi, c_1, c_2)$. This will turn out to yield a sufficient bound for the first square root in Equation (4.6).

Now, we are going to bound the expected difference between the position and the global attractor at time $t_H$:

$$\begin{aligned}
&|E_{t_0}[G^1_{t_H} - X^n_{t_H}]| \leq |A(L^n_{t_0})| + |E_{t_0}[X^n_{t_H}]| \\
&\leq |A(L^n_{t_0})| + c(\chi, c_1, c_2) \cdot (|\lambda_{max}|^{t_H-t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|) \\
&\leq c''(\chi, c_1, c_2) \cdot (|\lambda_{max}|^{t_H-t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|) \qquad (4.9)
\end{aligned}$$

Finally, $\mathrm{Var}_{t_0}[G^1_{t_H} - X^n_{t_H}]$ is bounded as follows with the Cauchy-Schwarz inequality in the second "$\leq$":

$$\begin{aligned}
\mathrm{Var}_{t_0}[G^1_{t_H} - X^n_{t_H}] &\leq \mathrm{Var}_{t_0}[G^1_{t_H}] + \mathrm{Var}_{t_0}[X^n_{t_H}] + 2 \cdot |\mathrm{Cov}_{t_0}[G^1_{t_H}, X^n_{t_H}]| \\
&\leq \mathrm{Var}_{t_0}[G^1_{t_H}] + \mathrm{Var}_{t_0}[X^n_{t_H}] + 2 \cdot \sqrt{\mathrm{Var}_{t_0}[G^1_{t_H}]} \cdot \sqrt{\mathrm{Var}_{t_0}[X^n_{t_H}]} \\
&\leq |A(L^n_{t_0})|^2 + \mathrm{Var}_{t_0}[X^n_{t_H}] + 2 \cdot |A(L^n_{t_0})| \cdot \sqrt{\mathrm{Var}_{t_0}[X^n_{t_H}]} \\
&\leq |A(L^n_{t_0})|^2 + (c'(\chi, c_1, c_2) \cdot (|\lambda'_{\max}|^{t_H - t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|))^2 \\
&\quad + 2 \cdot |A(L^n_{t_0})|^2 \cdot c'(\chi, c_1, c_2) \cdot (|\lambda'_{\max}|^{t_H - t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|) \\
&\leq c'''(\chi, c_1, c_2) \cdot (|\lambda'_{\max}|^{t_H - t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|)^2 \qquad (4.10)
\end{aligned}$$

With (4.7), (4.8), (4.9) and (4.10), we can bound the right side of (4.6) as follows:

$$\begin{aligned}
&\mathrm{E}_{t_0}\left[\sqrt{|V^n_{t_H}|} + \sqrt{|G^1_{t_H+1} - X^n_{t_H}|}\,\right] \\
&\leq \underbrace{(\sqrt{c + c'} + \sqrt{c'' + c'''})}_{=:\mathfrak{C}_H} \cdot \sqrt{\lambda^{t_H - t_0} \cdot (|X^n_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|} \\
&\leq \mathfrak{C}_H \cdot \sqrt{\lambda^{t_H - t_0} \cdot (|G^1_{t_0} - X^n_{t_0}| + |G^1_{t_0}| + |V^n_{t_0}|) + |A(L^n_{t_0})|} \\
&\leq \mathfrak{C}_H \cdot \sqrt{\lambda^{t_H - t_0} \cdot (|G^1_{t_0} - X^n_{t_0}| + |V^n_{t_0}|) + (1 + \lambda^{t_H - t_0}) \cdot |A(L^n_{t_0})|} \\
&\leq \mathfrak{C}_H \cdot \left(\lambda^{(t_H - t_0)/2} \cdot Y^n_{t_0} + \sqrt{2 \cdot |A(L^n_{t_0})|}\right)
\end{aligned}$$

where $\lambda$ denotes $\max\{|\lambda_{\max}|, |\lambda'_{\max}|\}$. The statement (4.2) follows by summing up over all particles and setting $C_H := \mathfrak{C}_H \cdot \sqrt{2}$.

Since

$$\sum_{n=1}^{N} Y^n_{t_0} \geq c_H \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_{t_0})|}$$

is equivalent to

$$\sum_{n=1}^{N} \sqrt{|A(L^n_{t_0})|} \leq 1/c_H \cdot \sum_{n=1}^{N} Y^n_{t_0},$$

we have

$$\begin{aligned} & E_{t_0}\left[\sum_{n=1}^{N}\sqrt{|V_{t_H}^n|}+\sqrt{|G_{t_H}^1-X_{t_H}^n|}\mid\sum_{n=1}^{N}Y_{t_0}^n\geq c_H\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}\right] \\ \leq\ & C_H\cdot\left(\lambda^{(t_H-t_0)/2}+\frac{1}{c_H}\right)\cdot\sum_{n=1}^{N}Y_{t_0}^n\leq(1-\delta_H)\cdot\sum_{n=1}^{N}Y_{t_0}^n \end{aligned}$$

with $\delta_H := 1 - C_H\cdot\left(\lambda^{(t_H-t_0)/2}+\frac{1}{c_H}\right)$. If $c_H \geq 4\cdot C_H =: c'_H$ and $t_H - t_0 \geq \lceil 2\cdot\log(4\cdot C_H)/\log(1/\lambda)\rceil =: t'_H$, then $\delta_H \geq 1/2$. That proves statement (4.3).

Additionally, from $\sum_{n=1}^{N} Y_{t_0}^n \geq c_H\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}$, it follows that

$$\begin{aligned} |A(G_{t_0}^1)|/\Phi_{t_0+1}^1 &= |A(G_{t_0+1}^1)|/\sqrt{\sum_{n=1}^{N}(a\cdot|V_{t_0}^n|+|G_{t_0}^1-X_{t_0}^n|)} \\ &\leq \sqrt{2\cdot N\cdot\max\{a,1\}}\cdot|A(G_{t_0}^1)|/\sum_{n=1}^{N}\left(\sqrt{|V_{t_0}^n|}+\sqrt{|G_{t_0}^1-X_{t_0}^n|}\right) \\ &\leq \sqrt{2\cdot N\cdot\max\{a,1\}}\cdot|A(G_{t_0}^1)|/\left(c_H\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}\right) \\ &\leq \frac{\sqrt{2\cdot\max\{a,1\}}}{c_H}\cdot\sum_{n=1}^{N}\sqrt{|A(L_t^n)|}\leq \mathrm{const}_H\cdot\Psi_t/c_H, \end{aligned}$$

where $\mathrm{const}_H := \sqrt{2\cdot\max\{a,1\}}$. That finishes the proof of statement (4.4).

Finally, we have that

$$\begin{aligned} \Psi_{t_0} &= C_\Psi\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}+\sum_{n=1}^{N}Y_{t_0}^n+\frac{|A(G_{t_0}^1)|}{\Phi_{t_0+1}^1} \\ &\leq \frac{C_\Psi}{c_H}\cdot\sum_{n=1}^{N}Y_{t_0}^n+\sum_{n=1}^{N}Y_{t_0}^n+\frac{\mathrm{const}}{c_H}\cdot\sum_{n=1}^{N}Y_{t_0}^n, \end{aligned}$$

i. e., $\sum_{n=1}^{N} Y_{t_0}^n \geq \Psi_{t_0}/(1+\frac{C_\Psi+\mathrm{const}_H}{c_H})$. That finishes the proof of statement (4.5). □

### Low Potential

Our next goal is to examine the case when the potential is too small, such that updates of the attractors happen frequently but the movement and therefore the decrease of $|A(G_t^n)|$ is insignificant. This happens if every particle has velocity and distance to both attractors much smaller than the distance to the optimum, i. e., the whole swarm is gathered close to a non-optimal point. In Section 4.1.1, we have proved that this situation is not stable, i. e., the swarm enters a state in which every velocity points towards the optimum and every particle updates its local attractor in every step. We called this state *running* and we have shown that for an appropriate parameter choice, during this state the potential increases exponentially.

For different parameter sets, we have provided experimental evidence in Experiment 4.1 that even for a small number of particles, the swarm accelerates. Here, we repeat a similar experiment. But instead of comparing different parameter sets, we keep the standard parameters of this thesis ($\chi = 0.72984$, $c_1 = c_2 = 1.496172$) while the number of particles varies.

**Experiment 4.3.** We initialize the particles uniformly over $[-100, 100]$ and use the objective function $f(x) = -x$, which obviously has no local optimum and is therefore useful for simulating the situation of a local optimum which is far out of reach. Figure 4.10 shows the measured courses of the potentials for swarm sizes from $N = 2$ to $N = 10$. We can see that the potentials indeed increase exponentially. In contrast to the case of a high potential, the number of particles indeed matters. Figure 4.10 clearly shows that with a larger swarm size, the swarm charges potential faster and is therefore able to heal itself faster from encountering the bad event $B_t^L$.

By reusing the result of Section 4.1.1, we can formally show that in presence of a too small potential, the term $\Psi_t^L$ satisfies the desired condition, i. e., it decreases sufficiently fast. The following lemma verifies 1.-4. for $B_t^L$.

**Lemma 4.3.** There are constants $t_L' \in \mathbb{N}$, $c_L' > 0$, $C_L > 0$, $\delta_L \in (0, 1)$ and $\text{const}_L$, depending only on $c_1$, $c_2$, $\chi$ and $N$, such that there is for every $t_0 \in \mathbb{N}$

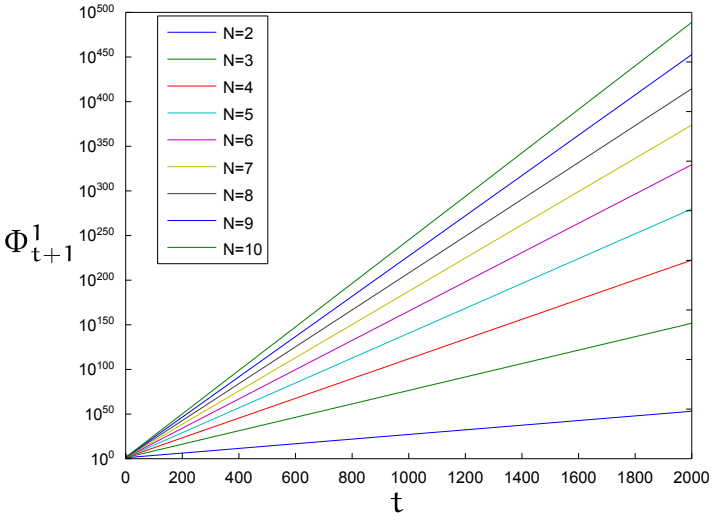



**Figure 4.10:** Particle swarm suffering from too low potentials while processing the 1-dimensional objective function $f(x) = -x$ with swarm sizes between $N = 2$ and $N = 10$ particles.

a $\mathcal{F}_{t_0}$-measurable time $\sigma_L$ with $t_0 \leq \sigma_L \leq t_0 + t'_L$ almost surely, such that for every $c_L > c'_L$ and every $t_L \geq t_0$, the following statements are fulfilled.

$$E\left[\frac{|A(G^1_{t_L})|}{\Phi^1_{t_L+1}} \mid \mathcal{F}_{t_0}\right] \leq C_L^{t_L - t_0} \cdot \frac{|A(G^1_{t_0+1})|}{\Phi^1_{t_0}}, \tag{4.11}$$

$$E\left[\frac{|A(G^1_{\sigma_L})|}{\Phi^1_{\sigma_L+1}} \mid \mathcal{F}_{t_0}, B^L_t\right] \leq (1 - \delta_L) \cdot \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}}, \tag{4.12}$$

$$B^L_t \text{ implies } \sum_{n=1}^{N} Y^n_{t_0} \leq \text{const}_L \cdot \Psi_{t_0}/c_L, \tag{4.13}$$

$$B^L_t \text{ implies } \frac{|A(G^1_{t_0+1})|}{\Phi^1_{t_0+1}} \geq \frac{\Psi_{t_0}}{1 + \text{const}_L \cdot C_\Psi/c_L}. \tag{4.14}$$

*Proof.* First, we establish the bound of the expected increase of $|A(G^n_t)|/\Phi^1_t$ without any further knowledge in order to prove statement (4.11). Since $|A(G^n_t)|$ is non-increasing over time, we only need to bound $E_{t_0}[1/\Phi^n_{t_H+1}]$. First, the focus lies only on a single step of just one particle of the swarm.

In order to simplify notation, we consider without loss of generality particle 1 as the one particle that is moved. The step of particle 1 can effect the contributions of the other particles, if it updates the global attractor and brings it closer to their positions. Therefore, a distinction between two cases is made: The first case is that the contribution of particle 1 to the potential is insignificant, so its step can only change the global attractor a little. In the second case, particle 1 has a high contribution to the swarm potential. In that case, its step might decrease the contributions of the other particles by an arbitrary amount, but its own contribution decreases only within certain bounds, therefore the whole swarm has a constant fraction of its potential left. Let $t \in \mathbb{N}$ be an arbitrary point in time.

Case 1: The swarm potential is much larger than the potential of particle 1. More precisely:

$$\sum_{n=1}^{N} |G_{t-1}^1 - X_{t-1}^n| \geq 2 \cdot N \cdot (\chi \cdot |V_{t-1}^1| + c_1 \cdot |L_{t-1}^1 - X_{t-1}^1| + (c_2+1) \cdot |G_{t-1}^1 - X_{t-1}^1|).$$

Since

$$\begin{aligned}
|G_{t-1}^1 - X_{t-1}^n| &\geq |G_{t-1}^1 - X_{t-1}^n| - |G_{t-1}^1 - G_{t-1}^1| \\
&\geq |G_{t-1}^1 - X_{t-1}^n| - |G_{t-1}^1 - X_t^1| \\
&\geq |G_{t-1}^1 - X_{t-1}^n| - |G_{t-1}^1 - X_{t-1}^1| - |X_t^1 - X_{t-1}^1| \\
&= |G_{t-1}^1 - X_{t-1}^n| - |G_{t-1}^1 - X_{t-1}^1| - |V_t^1| \\
&\geq |G_{t-1}^1 - X_{t-1}^n| - \chi \cdot |V_{t-1}^1| - c_1 \cdot |L_{t-1}^1 - X_{t-1}^1| \\
&\quad -(c_2+1) \cdot |G_{t-1}^1 - X_{t-1}^1|,
\end{aligned}$$

we have

$$\begin{aligned}
&\sum_{n=1}^{N} |G_{t-1}^1 - X_{t-1}^n| \\
&\geq \sum_{n=1}^{N} \Big( |G_{t-1}^1 - X_{t-1}^n| - \chi \cdot |V_{t-1}^1| - c_1 \cdot |L_{t-1}^1 - X_{t-1}^1| \\
&\quad - (c_2+1) \cdot |G_{t-1}^1 - X_{t-1}^1| \Big)
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2} \cdot \sum_{n=1}^{N} |G_{t-1}^1 - X_{t-1}^n| + \frac{1}{2} \cdot \sum_{n=1}^{N} |G_{t-1}^1 - X_{t-1}^n| - N \cdot \chi \cdot |V_{t-1}^1| \\
&\quad - N \cdot c_1 \cdot |L_{t-1}^1 - X_{t-1}^1| - N \cdot (c_2 + 1) \cdot |G_{t-1}^1 - X_{t-1}^1| \\
&\geq \frac{1}{2} \cdot \sum_{n=1}^{N} |G_{t-1}^1 - X_{t-1}^n|
\end{aligned}
$$

This observation leads to

$$
\begin{aligned}
\Phi_t^1 &= \sqrt{a|V_t^1| + |G_{t-1}^1 - X_t^1| + \sum_{n=2}^{N} \left(a|V_{t-1}^n| + |G_{t-1}^1 - X_{t-1}^n|\right)} \\
&\geq \sqrt{\frac{1}{2}|G_{t-1}^1 - X_t^1| + \sum_{n=2}^{N} \left(a|V_{t-1}^n| + \frac{1}{2}|G_{t-1}^1 - X_{t-1}^n|\right)} \\
&\geq \sqrt{\frac{1}{2}} \cdot \sqrt{|G_{t-1}^1 - X_t^1| + \sum_{n=2}^{N} \left(a|V_{t-1}^n| + |G_{t-1}^1 - X_{t-1}^n|\right)}.
\end{aligned}
$$

On the other hand, we have

$$
\begin{aligned}
\Phi_t^1 &= \sqrt{\sum_{n=1}^{N} a|V_{t-1}^n| + |G_{t-1}^1 - X_{t-1}^n|} \\
&\leq \sqrt{a|V_{t-1}^1| + |G_{t-1}^1 - X_{t-1}^1| + \sum_{n=2}^{N} \left(a|V_{t-1}^n| + |G_{t-1}^1 - X_{t-1}^n|\right)} \\
&\leq \sqrt{\left(1 + \frac{a}{2N\chi}\right)|G_{t-1}^1 - X_{t-1}^1| + \sum_{n=2}^{N} \left(a|V_{t-1}^n| + \frac{a}{2N\chi}|G_{t-1}^1 - X_{t-1}^n|\right)} \\
&\leq \sqrt{\left(1 + \frac{a}{2N\chi}\right)} \cdot \sqrt{|G_{t-1}^1 - X_{t-1}^1| + \sum_{n=2}^{N} \left(a|V_{t-1}^n| + |G_{t-1}^1 - X_{t-1}^n|\right)}.
\end{aligned}
$$

It follows that

$$\Phi_t^1 \geq \Phi_t^1 / \sqrt{2 \cdot (1 + a/(2 \cdot N \cdot \chi))}$$

and therefore

$$E_{t-1}[1/\Phi_t^1] \leq \sqrt{2 \cdot (1 + a/(2 \cdot N \cdot \chi))}/\Phi_t^1.$$

Case 2: Particle 1 contributes significantly to the swarm potential. More precisely:

$$\sum_{n=1}^{N} |G_{t-1}^1 - X_{t-1}^n| < 2 \cdot N \cdot (\chi \cdot |V_{t-1}^1| + c_1 \cdot |L_{t-1}^1 - X_{t-1}^1| + (c_2+1) \cdot |G_{t-1}^1 - X_{t-1}^1|).$$

Since particle 1 might be able to bring the global attractor closer to the other particles, we cannot rely on their old distances to the global attractor as a lower bound for the swarm potential after the step of particle 1. However, the velocities of the remaining swarm members will stay the same. For the potential after the step of particle 1, that means

$$\begin{aligned} \Phi_t^1 &= a|V_t^1| + |\mathcal{G}_{t-1}^1 - X_t^1| + \sum_{n=2}^{N} \left( a|V_{t-1}^n| + |\mathcal{G}_{t-1}^1 - X_{t-1}^n| \right) \\ &\geq a|V_t^1| + |\mathcal{G}_{t-1}^1 - X_t^1| + \sum_{n=2}^{N} a|V_{t-1}^n|. \end{aligned}$$

On the other hand, the portion of the potential that comes from the distances of the particles $2, \ldots, N$ to the global attractor, i. e., the part that can actually vanish is exactly the part that is bounded in terms of the contribution of particle 1 to the potential:

$$\begin{aligned} \Phi_t^1 &= \sum_{n=1}^{N} a|V_{t-1}^n| + |G_{t-1}^1 - X_{t-1}^n| \\ &\leq 2 \cdot N \cdot \left( \chi \cdot |V_{t-1}^1| + c_1 \cdot |L_{t-1}^1 - X_{t-1}^1| + (c_2+1) \cdot |G_{t-1}^1 - X_{t-1}^1| \right) \\ &\quad + a|V_{t-1}^1| + \sum_{n=2}^{N} a|V_{t-1}^n|. \end{aligned}$$

By setting

$$\begin{aligned} b_1 &:= 2 \cdot N \cdot (\chi \cdot |V_{t-1}^1| + c_1 \cdot |L_{t-1}^1 - X_{t-1}^1| + (c_2+1) \cdot |G_{t-1}^1 - X_{t-1}^1|) \\ &\quad + a \cdot |V_{t-1}^1| + \sum_{n=2}^{N} a \cdot |V_{t-1}^n|, \\ b_2 &:= a|V_t^1| + |\mathcal{G}_{t-1}^1 - X_t^1|, \\ c &:= \sum_{n=2}^{N} a|V_{t-1}^n|, \end{aligned}$$

and with the generalized weighted mean inequality, we obtain

$$
\begin{aligned}
&\mathrm{E}_{t-1}[\Phi_t^1/\Phi_t^1] \\
&\quad\leq\ \mathrm{E}_{t-1}\left[\sqrt{\frac{2N(\chi|V_{t-1}^1|+c_1|L_{t-1}^1-X_{t-1}^1|+(c_2+1)|G_{t-1}^1-X_{t-1}^1|)}{a|V_t^1|+|G_{t-1}^1-X_t^1|+\sum_{n=2}^N a|V_{t-1}^n|}}\right. \\
&\qquad\left.\overline{+\frac{a|V_{t-1}^1|+\sum_{n=2}^N a|V_{t-1}^n|}{a|V_t^1|+|G_{t-1}^1-X_t^1|+\sum_{n=2}^N a|V_{t-1}^n|}}\right] \\
&\quad=\ \mathrm{E}_{t-1}\left[\sqrt{\frac{b_1+a}{b_2+a}}\right] \\
&\quad\leq\ \mathrm{E}_{t-1}\left[w\cdot\sqrt{\frac{b_1}{b_2}}+(1-w)\cdot\sqrt{\frac{a}{a}}\right]=w\cdot\mathrm{E}_{t-1}\left[\sqrt{\frac{b_1}{b_2}}\right]+(1-w)
\end{aligned}
$$

for $w = b_1/(b_1+a) \leq 1$. Note that $w$ is $\mathcal{F}_{t-1}$-measurable. The term $\mathrm{E}_{t-1}[\sqrt{b_1/b_2}]$ is bounded in the following way:

$$
\begin{aligned}
&\mathrm{E}_{t-1}\left[\sqrt{\frac{b_1}{b_2}}\right] \\
&=\mathrm{E}_{t-1}\left[\sqrt{\frac{2\cdot N\cdot(\chi\cdot|V_{t-1}^1|+c_1\cdot|L_{t-1}^1-X_{t-1}^1|+(c_2+1)\cdot|G_{t-1}^1-X_{t-1}^1|)}{a\cdot|V_t^1|+|G_{t-1}^1-X_t^1|}}\right. \\
&\qquad\left.\overline{+\frac{a|V_{t-1}^1|}{a\cdot|V_t^1|+|G_{t-1}^1-X_t^1|}}\right] \\
&\leq\mathrm{E}_{t-1}\left[\sqrt{\frac{2\cdot N\cdot(\chi\cdot|V_{t-1}^1|+c_1\cdot|L_{t-1}^1-X_{t-1}^1|+(c_2+1)\cdot|G_{t-1}^1-X_{t-1}^1|)}{a\cdot|V_t^1|}}\right. \\
&\qquad\left.\overline{+\frac{a\cdot|V_{t-1}^1|}{a\cdot|V_t^1|}}\right] \\
&=\mathrm{E}_{t-1}\left[\sqrt{\frac{2\cdot N\cdot(\chi\cdot|V_{t-1}^1|+c_1\cdot|L_{t-1}^1-X_{t-1}^1|+(c_2+1)\cdot|G_{t-1}^1-X_{t-1}^1|)}{a\cdot|\chi\cdot|V_{t-1}^1|+c_1\cdot r_t^1\cdot(L_{t-1}^1-X_{t-1}^1)+c_2\cdot s_t^1\cdot(G_{t-1}^1-X_{t-1}^1)|}}\right. \\
&\qquad\left.\overline{+\frac{a\cdot|V_{t-1}^1|}{a\cdot|\chi\cdot|V_{t-1}^1|+c_1\cdot r_t^1\cdot(L_{t-1}^1-X_{t-1}^1)+c_2\cdot s_t^1\cdot(G_{t-1}^1-X_{t-1}^1)|}}\right]
\end{aligned}
$$

$$\leq \sqrt{4 \cdot N/a + 1/\chi} \cdot \\ \cdot E_{t-1}\left[\sqrt{\frac{\chi \cdot |V^1_{t-1}| + c_1 \cdot |L^1_{t-1} - X^1_{t-1}| + c_2 \cdot |G^1_{t-1} - X^1_{t-1}|}{|\chi \cdot |V^1_{t-1}| + c_1 \cdot r^1_t \cdot (L^1_{t-1} - X^1_{t-1}) + c_2 \cdot s^1_t \cdot (G^1_{t-1} - X^1_{t-1})|}}\right].$$

The remaining expectation leads to an expression of the form

$$\int_0^1 \int_0^1 \sqrt{\frac{1}{|x + r \cdot y + s \cdot z|}} dr ds$$

with $|x|+|y|+|z| = 1$, which is, as straight-forward calculations show, bounded by some constant.

Both cases together show that on expectation the reciprocal of the swarm potential increases by at most a constant factor during a single step. Therefore, it also increases on expectation by at most a constant factor during a complete iteration. That finishes the proof of statement (4.11).

The next goal is to verify the improved bound for the case of a very small swarm potential. For simplicity, we use the notation

$$E_L[\circ] := E_{t_0}\left[\circ \;\middle|\; |A(G^1_{t_0})| \Big/ \sum_{n=1}^N Y^n_{t_0} \geq c_L \cdot \sum_{n=1}^N \left(Y^n_{t_0} + \sqrt{|A(L^n_{t_0})|}\right)\right],$$

$P_L(\circ)$ is used analogously. Note that

$$|A(G^1_{t_0})| / \sum_{n=1}^N Y^n_{t_0} \geq c_L \cdot \sum_{n=1}^N \left(Y^n_{t_0} + \sqrt{|A(L^n_{t_0})|}\right)$$

implies

$$|A(G^1_{t_0})| / \sum_{n=1}^N Y^n_{t_0} \geq c_L \cdot \sum_{n=1}^N Y^n_{t_0}$$

and therefore

$$|A(G^1_{t_0})| \geq c_L \cdot \Big(\sum_{n=1}^N Y^n_{t_0}\Big)^2 \geq c_L \cdot \sum_{n=1}^N (|V^n_{t_0}| + |G^1_{t_0} - X^n_{t_0}|).$$

In particular, for $c_L > 1$, the condition implies that $\operatorname{sign}(G^1_{t_0}) = \operatorname{sign}(X^n_t)$ for every particle $n$. Without loss of generality, we can assume that for every $n$

$X^n_{t_0} < G^1_{t_0} < 0$. First, we check that a large distance between a particle and its local attractor increases the particles $Y$-value on expectation even within a single step. Consider the case of $|L^{n_0}_{t_0} - X^{n_0}_{t_0}| > 9 \cdot N/c_1 \cdot |A(G^1_{t_0})|/c_L$ for some particle $n_0$. Then, we have

$$
\begin{aligned}
&E_L\left[\frac{\Phi^1_{t_0+1}}{\Phi^1_{t_0+2}} \;\middle|\; |L^n_{t_0} - X^n_{t_0}| \geq 9 \cdot N/c_1 \cdot |A(G^1_{t_0})|/c_L\right] \\
&\leq E_L\left[\frac{\sqrt{2 \cdot N} \cdot \sum_{n=1}^N Y^n_{t_0}}{|V^{n_0}_{t+1}|} \;\middle|\; |L^n_{t_0} - X^n_{t_0}| \geq 9 \cdot N/c_1 \cdot |A(G^1_{t_0})|/c_L\right] \\
&\leq \sup_{s^{n_0}_{t_0} \in [0,1]} \int_0^1 \frac{\sqrt{2 \cdot N} \cdot \sum_{n=1}^N Y^n_{t_0}}{|\chi \cdot V^{n_0}_{t_0} + c_1 \cdot r^{n_0}_{t_0} \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0}) + c_2 \cdot s^{n_0}_{t_0} \cdot (G^{n_0}_{t_0} - X^{n_0}_{t_0})|} dr^{n_0}_{t_0+1} \\
&\leq \sup_{s^{n_0}_{t_0} \in [0,1]} \left(\sqrt{2 \cdot N} \cdot \sum_{n=1}^N Y^n_{t_0}\right) \cdot 2/(c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0})) \cdot \\
&\quad \cdot \left(\sqrt{|c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0}) - \chi \cdot V^{n_0}_{t_0} - c_2 \cdot s^{n_0}_{t_0} \cdot (G^{n_0}_{t_0} - X^{n_0}_{t_0})|}\right. \\
&\qquad \left. - \sqrt{|\chi \cdot V^{n_0}_{t_0} + c_2 \cdot s^{n_0}_{t_0} \cdot (G^{n_0}_{t_0} - X^{n_0}_{t_0})|}\right) \\
&\leq \left(\sqrt{8 \cdot N} \cdot \sum_{n=1}^N Y^n_{t_0}\right) \cdot \frac{\sqrt{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0}) + \chi \cdot |V^{n_0}_{t_0}|}}{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0})} \\
&\leq \left(\sqrt{8 \cdot N} \cdot \sqrt{\frac{|A(G^1_{t_0})|}{c_L}}\right) \cdot \frac{\sqrt{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0}) + \chi \cdot \left(\sum_{n=1}^N Y^n_{t_0}\right)^2}}{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0})} \\
&\leq \left(\sqrt{8 \cdot N} \cdot \sqrt{\frac{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0})}{9 \cdot N}}\right) \cdot \frac{\sqrt{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0}) + \chi \cdot |A(G^1_{t_0})|/c_L}}{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0})} \\
&\leq \left(\sqrt{8/9} \cdot \sqrt{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0})}\right) \cdot \frac{\sqrt{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0}) \cdot (1 + 1/(9 \cdot N))}}{c_1 \cdot (L^{n_0}_{t_0} - X^{n_0}_{t_0})} \\
&= \sqrt{8/9 \cdot 1 + 1/(9 \cdot N)} < 1,
\end{aligned}
$$

i. e., in this situation we can set $\sigma_L := 1$. For the rest of the proof, we assume that $|L^{n_0}_{t_0} - X^{n_0}_{t_0}| \leq 9 \cdot N/c_1 \cdot |A(G^1_{t_0})|/c_L$. In particular, for a sufficiently large $c_L$ this guarantees that the local attractors also have the same sign as the global

attractor, i. e., without loss of generality, we can assume that $X^n_{t_0} < L^n_{t_0} < G^1_{t_0} < 0$ for every $n$.

There are two essentially different cases, depending on the shape of the objective function $f$: Either, $A(G^1_{t_0})$ has size about $2 \cdot |G^1_{t_0}|$, as it is the case when processing, e. g., objective function Sphere. Or $|A(G^1_{t_0})| \gg |G^1_{t_0}|$. In the first case, the particles have a large area on which the function is monotone to accelerate. In the second case, the monotone area might be too small for a decent acceleration of the particles, but after overcoming the optimum at 0, the particles improve the global attractor significantly.

In the proof of Lemma 4.1 in Section 4.1.1, we have shown that on a monotone area the particles do actually accelerate, as soon as they have entered a certain constellation. More precise, we have shown that for proper choices of the swarm parameters $\chi$, $c_1$, $c_2$, $N$ and of the parameter $a$ of the potential and every $t \in \mathbb{N}$,

$$E_t[\Phi^n_t/\Phi^n_{t+1} \mid \mathcal{F}_t, \exists \varepsilon \in \{-1, 1\}, \forall n : \operatorname{sign}(V^n_t) = \operatorname{sign}(G^1_t - X^n_t) = \varepsilon] \leq q < 1$$

holds, where $q$ depends only on $\chi$, $c_1$, $c_2$, $N$ and $a$.

The state "running" describes the situation when on a monotone area the velocities all point towards the direction in which the function is falling. Note that under such circumstances, the monotonicity of $f$ implies that $\operatorname{sign}(V^n_t) = \operatorname{sign}(G^1_t - X^n_t) = \operatorname{sign}(L^n_t - X^n_t)$ for every particle $n$.

Since the improvements of $|A(G)|/\Phi$ start as soon as either the swarm gets running or it surpasses 0, the strategy for the proof of statement (4.12) is to separate the expectation into two conditioned expectations, one that assumes that the swarm either started running or reached a positive position not later than at a certain iteration $t_0 + t'_0$ and one that assumes that the swarm did neither of both until time $t_0 + t'_0$. Let the stopping time $\tau$ be the first time $\geq t$ when at least one of the two events happens, namely when either the swarm is running or it reaches a position $> 0$. We have

$$\begin{aligned}
&E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1}] \\
&= P_L(\tau \leq t_0 + t'_0) \cdot E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau \leq t_0 + t'_0] \\
&\quad + P_L(\tau > t_0 + t'_0) \cdot E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau > t_0 + t'_0] \\
&\leq E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau \leq t_0 + t'_0] \\
&\quad + P_L(\tau > t_0 + t'_0) \cdot E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau > t_0 + t'_0]
\end{aligned} \tag{4.15}$$

In the following, we will bound each of the three remaining expressions, starting with $E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau > t_0 + t'_0]$.

Since the expectation $E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau > t_0 + t'_0]$ is conditioned, we cannot apply (4.11) for an upper bound. Instead, we prove a uniform bound on the future swarm potential. This is possible because in the current situation, no particle can bring the global attractor closer to a different particle, so each one can only damage its own contribution to the potential. The following calculations show that every iteration preserves at least a constant fraction of the swarm potential. The updated values of particle $n$ are

$$V^n_{t_0+1} = \chi \cdot V^n_{t_0} + c_1 \cdot r^n_{t_0+1} \cdot (L^n_{t_0} - X^n_{t_0}) + c_2 \cdot s^n_{t_0+1} \cdot (G^n_{t_0} - X^n_{t_0}),$$
$$X^n_{t_0+1} = X^n_{t_0} + \chi \cdot V^n_{t_0} + c_1 \cdot r^n_{t_0+1} \cdot (L^n_{t_0} - X_{t_0} t^n) + c_2 \cdot s^n_{t_0+1} \cdot (G^n_{t_0} - X^n_{t_0}).$$

If $V^n_{t_0}$ is positive, it follows that

$$a \cdot |V^n_{t_0+1}| + |G^1_{t_0+1} - X^n_{t_0+1}| \geq a \cdot |V^n_{t_0+1}| \geq a \cdot \chi \cdot |V^n_{t_0}|$$

and additionally

$$\begin{aligned} a \cdot |V^n_{t_0+1}| + |G^1_{t_0+1} - X^n_{t_0+1}| &= a \cdot V^n_{t_0+1} + G^1_{t_0+1} - X^n_{t_0+1} \\ &\geq \min\{a, 1\} \cdot (V^n_{t_0+1} + G^n_{t_0} - X^n_{t_0+1}) \\ &= \min\{a, 1\} \cdot |G^n_{t_0} - X^n_{t_0}|. \end{aligned}$$

Since both bounds hold, any weighted mean of them is also a valid bound. It follows

$$a \cdot |V^n_{t_0+1}| + |G^1_{t_0+1} - X^n_{t_0+1}| \geq \frac{\min\{a, 1\} \cdot \chi}{\min\{a, 1\} + \chi} \cdot (a \cdot |V^n_{t_0}| + |G^n_{t_0} - X^n_{t_0}|).$$

If on the other hand $V^n_{t_0}$ is negative, two subcases need to be considered. If $-\chi \cdot V^n_{t_0} \geq (c_1 + c_2) \cdot (G^n_{t_0} - X^n_{t_0})$, we have

$$\begin{aligned} -V^n_{t_0+1} &\geq -\chi \cdot V^n_{t_0} - c_1 \cdot r^n_{t_0+1} \cdot (L^n_{t_0} - X^n_{t_0}) - c_2 \cdot s^n_{t_0+1} \cdot (G^n_{t_0} - X^n_{t_0}) \\ &\geq -\chi \cdot V^n_{t_0} - (c_1 + c_2) \cdot (G^n_{t_0} - X^n_{t_0}) \geq 0, \\ G^1_{t_0+1} - X^n_{t_0+1} &\geq G^n_{t_0} - X^n_{t_0} - V^n_{t_0+1}. \end{aligned}$$

Consequently, for the contribution to the potential, we have

$$\begin{aligned} &a \cdot |V^n_{t_0+1}| + |G^1_{t_0+1} - X^n_{t_0+1}| = -a \cdot V^n_{t_0+1} + G^1_{t_0+1} - X^n_{t_0+1} \\ &= -(a+1) \cdot V^n_{t_0+1} + G^n_{t_0} - X^n_{t_0} \\ &\geq -(a+1) \cdot V^n_{t_0+1} + 1/2 \cdot (G^n_{t_0} - X^n_{t_0}) + 1/2 \cdot (V^n_{t_0+1} - \chi \cdot V^n_{t_0})/(c_1 + c_2) \\ &\geq 1/2 \cdot (G^n_{t_0} - X^n_{t_0}) - \chi/2 \cdot 1/(c_1 + c_2) \cdot V^n_{t_0} \\ &\geq \chi/2 \cdot 1/(c_1 + c_2) \cdot \min\{1, 1/a\} \cdot (G^n_{t_0} - X^n_{t_0} - a \cdot V^n_{t_0}) \\ &= \chi/2 \cdot 1/(c_1 + c_2) \cdot \min\{1, 1/a\} \cdot (a \cdot |V^n_{t_0}| + |G^1_{t_0} - X^n_{t_0}|). \end{aligned}$$

If $-\chi \cdot V^n_{t_0} < (c_1 + c_2) \cdot (G^n_{t_0} - X^n_{t_0})$, we obtain

$$\begin{aligned}
&a \cdot |V^n_{t_0+1}| + |G^1_{t_0+1} - X^n_{t_0+1}| \\
&\geq \min\{a, 1\} \cdot (V^n_{t_0+1} + G^n_{t_0+1} - X^n_{t_0+1}) \\
&= \min\{a, 1\} \cdot (G^n_{t_0+1} - X^n_{t_0}) \\
&\geq \min\{a, 1\} \cdot (G^n_{t_0} - X^n_{t_0}) \\
&\geq \min\{a, 1\} \cdot \frac{\chi}{\chi + a \cdot (c_1 + c_2)} (G^n_{t_0} - X^n_{t_0} - a \cdot V^n_{t_0}) \\
&= \min\{a, 1\} \cdot \frac{\chi}{\chi + a \cdot (c_1 + c_2)} (a \cdot |V^n_{t_0}| + |G^n_{t_0} - X^n_{t_0}|).
\end{aligned}$$

Putting both cases together and building the minimum over all appearing constants leads to

$$\Phi^1_{t_0+1} \geq \text{const} \cdot \Phi^1_{t_0},$$

where $\text{const} = \text{const}(\chi, c_1, c_2, a)$. This implies

$$E_L[1/\Phi^1_{\sigma_L+1} \mid \tau > t_0 + t'_0] \leq \text{const}^{t'_0} \cdot C_L^{\sigma_L - t_0 - t'_0} \cdot 1/\Phi^1_{t_0+1}.$$

Now, we are going to bound the next term of (4.15), namely

$$E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau \leq t_0 + t'_0].$$

Since the condition

$$\sqrt{|A(G^1_{t_0})|} \geq c_L \cdot \sum_{n=1}^{N} \left( \sqrt{|V^n_{t_0}|} + \sqrt{G^1_{t_0} - X^n_{t_0}} \right)$$

does not imply any lower bound on the number of iterations the swarm can keep running, we first restrict the considerations to the case when

$$|V^n_{t_0}| + |G^1_{t_0} - X^n_{t_0}| + |L^n_{t_0} - X^n_{t_0}| \leq |G^1_{t_0}|/\sqrt{c_L}$$

for every $n$. Since

$$\begin{aligned}
|V^n_{t_0+1}| &\leq \chi \cdot |V^n_{t_0}| + c_1 \cdot |L^n_{t_0} - X_{t_0}| + c_2 \cdot |G^1_{t_0} - X^n_{t_0}|, \\
|L^n_{t_0+1} - X^n_{t_0+1}| &\leq |L^n_{t_0} - X^n_{t_0}| + |V^n_{t_0+1}|, \\
|G^1_{t_0+1} - X^n_{t_0+1}| &\leq |G^1_{t_0} - X^n_{t_0}| + \sum_{n'=1}^{N} |V^{n'}_{t_0+1}|,
\end{aligned}$$

during one iteration $\sum_{n=1}^{N} |V_{t_0}^n| + |L_{t_0}^n - X_{t_0}^n| + |G_{t_0}^1 - X_{t_0}^n|$ increases by at most a factor const that depends only on $c_1$, $c_2$, $\chi$ and $N$. Therefore, we have for every particle $n$:

$$\begin{aligned} |X_{t_0+t_0'}^n - G_{t_0}^1| &\leq |G_{t_0}^1 - X_{t_0}^n| + \sum_{l=1}^{t_0'-1} |V_{t_0+l}^n| \\ &\leq |G_{t_0}^1| \cdot (1/\sqrt{c_L} + \sum_{i=1}^{t_0'-1} \text{const}^i/\sqrt{c_L}) < |G_{t_0}^1| \end{aligned}$$

for $t_0' \leq \frac{\log c_L}{\log \text{const}}$. This implies that the swarm will for $t_0' \leq \frac{\log c_L}{\log \text{const}}$ reach no positive position at time $t_0 + t_0'$ and therefore still be running.

In combination with the results from Lemma 4.1 of Section 4.1.1, this leads to

$$E_L[|A(G_{\sigma_L}^1)|/\Phi_{\sigma_L+1}^1 \mid \tau \leq t_0 + t_0'] \leq \text{const}^{t_0'} \cdot q^{\sigma_L - t_0 - t_0'} \cdot |A(G_{t_0}^1)|/\Phi_{t_0+1}^1$$

for $t_0' \leq \frac{\log c_L}{\log \text{const}}$.

It remains the case when

$$|V_{t_0}^n| + |G_{t_0}^n - X_{t_0}^n| + |L_{t_0}^n - X_{t_0}^n| \geq |G_{t_0}^1|/\sqrt{c_L}$$

for some $n$. If that is the case, we cannot expect a large increase of $\Phi$. Instead, we can expect a large decrease of $|A(G)|$ since after the swarm stops running, $|A(G)| \leq \text{const} \cdot |G_{t_0}^1|$ will hold. Note that

$$|A(G_{t_0}^1)|/c_L \geq \sum_{n'=1}^{N} |V_{t_0}^{n'}| + |G_{t_0}^{n'} - X_{t_0}^{n'}| + |L_{t_0}^{n'} - X_{t_0}^{n'}|$$

implies here

$$|A(G_{t_0}^1)| \geq \sqrt{c_L} \cdot |G_{t_0}^1|.$$

For our calculations regarding the decrease of $|A(G_{t_0}^1)|$, we introduce a sequence of random variables $(A_t')_{t \geq t_0}$, which has the constant value $|A(G_{t_0}^1)|$ until the first particle surpasses the optimum at 0. From that point in time on, during each step of a particle $A_t'$ decreases by a factor of $q/C_L$, i. e., $A_t'$ pays for the possible decrease of $\Phi$ and additionally it decreases by a factor of $q$. For the formal analysis, $A_t'$ is defined in the following way:

$$\begin{aligned} A_{t_0}' &:= |A(G_{t_0}^1)| \\ A_{t+1}' &:= \begin{cases} A_t', & \text{if } \forall s, t_0 \leq s \leq t, \forall n : \text{sign}(X_s^n) = \text{sign}(X_{t_0}^1), \\ A_t' \cdot (q/C_L)^N, & \text{otherwise.} \end{cases} \end{aligned}$$

Similar to the previous case, we have

$$E_L[A'_{\sigma_L}/\Phi^1_{\sigma_L+1} \mid \tau \le t_0 + t'_0] \le \text{const}^{t'_0} \cdot q^{\sigma_L - t_0 - t'_0} \cdot |A(G^1_{t_0})|/\Phi^1_{t_0+1}.$$

Now, we have to show that

$$|A(G^1_{t_0+t})| \le A'_{t_0+t}$$

for every $t \le \text{const}(\chi, c_1, c_2, N) \cdot \log c_L$. Let $t_0 + \tau_0$ denote the first time when some particle surpasses 0. For $t < \tau_0$, $|A(G^1_{t_0+t})| \le A'_{t_0+t}$ follows directly from the definition of $A'$. For $t \ge \tau_0$, let $n_0$ be the first particle that overcame the optimum at 0. We have

$$|A(G^1_{t_0+t+1})| \le |A(G^{n_0}_{t_0+\tau_0+1})| \le |X^{n_0}_{t_0+\tau_0} - X^{n_0}_{t_0+\tau_0-1}| = |V^{n_0}_{t_0+\tau_0}|. \tag{4.16}$$

As a bound for the velocity when overcoming 0, we have

$$\begin{aligned} |V^{n_0}_{t_0+\tau_0}| \le \chi \cdot |X^{n_0}_{t_0+\tau_0-1} - X^{n_0}_{t_0+\tau_0-2}| + c_1 \cdot (L^{n_0}_{t_0+\tau_0-1} - X_{t_0+\tau_0-1}) \\ + c_2 \cdot (G^{n_0}_{t_0+\tau_0-1} - X^{n_0}_{t_0+\tau_0-1}). \end{aligned} \tag{4.17}$$

Before the time $t_0 + \tau_0$, the largest possible distance between the position of some particle $n$ and 0 is bounded from above by

$$\begin{aligned} |X^n_{t_0}| + \sum_{t=0}^{\infty} \chi^t \cdot |V^n_{t_0}| &\le |G^1_{t_0}| + |X^n_{t_0} - G^1_{t_0}| + 1/(1-\chi) \cdot |V^n_{t_0}| \\ &\le |G^1_{t_0}| + 1/(1-\chi) \cdot |A(G^1_{t_0})|/c_L \\ &\le \text{const} \cdot |A(G^1_{t_0})|/\sqrt{c_L} \end{aligned}$$

for $c_L > 1$. Therefore, we have for every $n \le N$ and every $s \le \tau_0$

$$|X^n_{t_0+s}| \le \text{const} \cdot |A(G^1_{t_0})|/\sqrt{c_L}.$$

Since the global and local attractors are also former positions of some particles, this, together with (4.16) and (4.17), implies

$$|A(G^1_{t_0+t})| \le \text{const} \cdot |A(G^1_{t_0})|/\sqrt{c_L}.$$

From the definition of $A'$, it follows that

$$A'_{t_0+t} \ge |A(G^1_{t_0})| \cdot (q/C_L)^{N \cdot t}.$$

For

$$t \leq \log(\sqrt{c_L}/\text{const})/(N \cdot \log(C_L/q)) = \text{const} \cdot \log(c_L)$$

and the choice

$$\sigma_L := t_0 + \text{const} \cdot \log(c_L),$$

it follows that

$$\begin{aligned} |A(G^1_{t_0+t})| &\leq \text{const} \cdot |A(G^1_{t_0})|/\sqrt{c_L} \\ &\leq \text{const} \cdot A'_{t_0+t} \cdot (C_L/q)^{N \cdot t}/\sqrt{c_L} \\ &\leq A'_{t_0+t}. \end{aligned}$$

Therefore, we obtain

$$\begin{aligned} E_L[|A(G^1_{\sigma_L})|/\Phi^1_{\sigma_L+1} \mid \tau \leq t_0 + t'_0] &\leq E_L[A'_{\sigma_L}/\Phi^1_{\sigma_L+1} \mid \tau \leq t_0 + t'_0] \\ &\leq \text{const}^{t'_0} \cdot q^{\sigma_L - t_0 - t'_0} \cdot |A(G^1_{t_0})|/\Phi^1_{t_0+1}. \end{aligned}$$

Finally, we examine the last term of (4.15), namely the probability for the swarm to become running or overcoming 0 until time $t_0 + t'_0$. Once a particle has a velocity pointing towards the optimum, it will not change its sign before passing it. So, if the swarm is neither running at time $t_0 + t'_0$ nor has jumped over 0, there is at least one particle $n$ with a velocity pointing away from 0. For this particle and every $0 < t < t'_0$, we have

$$\begin{aligned} |V^n_{t_0+t+1}| &\leq \chi \cdot |V^n_{t_0+t}| - c_1 \cdot r^n_{t_0+t+1} \cdot |L^n_{t_0+t} - X^n_{t_0+t}| \\ &\quad - c_2 \cdot s^n_{t_0+t+1} \cdot |G^n_{t_0+t} - X^n_{t_0+t}| \\ &\leq |V^n_{t_0+t+1}| \leq \chi \cdot |V^n_{t_0+t}| - c_1 \cdot r^n_{t_0+t+1} \cdot (L^n_{t_0+1} - X^n_{t_0+1}) \\ &\leq |V^n_{t_0+t+1}| \leq \chi \cdot |V^n_{t_0+t}| - c_1 \cdot r^n_{t_0+t+1} \cdot \chi \cdot |V^n_{t_0}|. \end{aligned}$$

It follows, that

$$|V^n_{t_0+t'_0}| \leq \chi^{t'_0} \cdot |V^n_{t_0}| - c_1 \cdot \chi \cdot |V^n_{t_0}| \cdot \sum_{t=1}^{t'_0-1} \chi^{t'_0-t-1} \cdot r^n_{t_0+t+1}$$

and therefore for every $t$

$$r^n_{t_0+t+1} \leq \frac{\chi^t}{c_1}.$$

Since the $r^n_{t_0+t+1}$ are independent, we have

$$P_L(\tau > t_0 + t'_0) \leq N \cdot \prod_{t=1}^{t'_0-1} \frac{\chi^t}{c_1} = N \cdot \frac{\chi^{(t'_0)^2/2}}{(\sqrt{\chi} \cdot c_1)^{t'_0}}.$$

Putting things together, we obtain

$$\begin{aligned}
&E_L[|A(G^1_{\sigma_L})|/\Phi^n_{\sigma_L}] \le E_L[|A(G^1_{\sigma_L})|/\Phi^n_{\sigma_L} \mid \tau \le t_0 + t_0'] \\
&\quad + P_L(\tau > t_0 + t_0') \cdot E_L[|A(G^1_{\sigma_L})|/\Phi^n_{\sigma_L} \mid \tau > t_0 + t_0'] \\
&\le \mathrm{const}^{t_0'} \cdot q^{\sigma_L - t_0 - t_0'} \cdot \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}} + \frac{\chi^{(t_0')^2/2}}{(\sqrt{\chi}\cdot c_1)^{t_0'}} \cdot \mathrm{const}^{t_0'} \cdot \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}} \\
&= \left( \mathrm{const}^{t_0'} \cdot q^{\sigma_L - t_0 - t_0'} + \frac{\chi^{(t_0')^2/2}}{(\sqrt{\chi}\cdot c_1)^{t_0'}} \cdot \mathrm{const}^{t_0'} \right) \cdot \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}}.
\end{aligned}$$

for every $t_0' \le \frac{\log c_L}{\log \mathrm{const}}$. To finish the proof of statement (4.12), we first choose $t_0'$ sufficiently large, such that

$$\frac{\chi^{(t_0')^2/2}}{(\sqrt{\chi}\cdot c_1)^{t_0'}} \cdot \mathrm{const}^{t_0'} < 1/3.$$

Then, we choose $\sigma_L$ sufficiently large to obtain $\mathrm{const}^{t_0'} \cdot q^{\sigma_L - t_0 - t_0'} < 1/3$. The value $t'$ is set accordingly. Finally, we choose $c_L$ sufficiently large to guarantee $t_0' \le \frac{\log c_L}{\log \mathrm{const}}$. Note that all the choices only depend on the fixed parameters of the swarm, i. e., on $\chi$, $c_1$, $c_2$, $N$, $a$ and $q$ which itself depends only on the other swarm parameters. With the choice of $\delta_L := 1/3$, the proof of statement (4.12) is finished.

From $\frac{|A(G^1_{t_0})|}{\sum_{n=1}^N Y^n_{t_0}} \ge c_L \cdot \sum_{n=1}^N (Y^n_{t_0} + \sqrt{|A(L^n_{t_0})|})$, it follows that

$$\begin{aligned}
\sum_{n=1}^N Y^n_{t_0} &\le \frac{|A(G^1_{t_0})|}{c_L \cdot \sum_{n=1}^N Y^n_{t_0}} \\
&= \frac{|A(G^1_{t_0})|}{c_L \cdot \sum_{n=1}^N (\sqrt{|V^n_t|} + \sqrt{|G^1_t - X^n_t|})} \\
&\le \frac{|A(G^1_{t_0})|}{c_L \cdot \sqrt{\sum_{n=1}^N (|V^n_t| + |G^1_t - X^n_t|)}} \\
&\le \frac{\max\{\sqrt{a}, 1\} \cdot |A(G^1_{t_0})|}{c_L \cdot \Phi^1_{t+1}} \le \max\{\sqrt{a}, 1\} \cdot \Psi_{t_0}/c_L.
\end{aligned}$$

That proves statement (4.13).

Finally, $\frac{|A(G^1_{t_0})|}{\sum_{n=1}^N Y^n_{t_0}} \geq c_L \cdot \sum_{n=1}^N (Y^n_{t_0} + \sqrt{|A(L^n_{t_0})|})$ implies

$$\begin{aligned}
\Psi_{t_0} &= C_\Psi \cdot \sum_{n=1}^N \sqrt{|A(L^n_{t_0})|} + \sum_{n=1}^N Y^n_{t_0} + \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}} \\
&\leq (C_\Psi + 1)/c_L \cdot \frac{|A(G^1_{t_0})|}{\sum_{n=1}^N Y^n_{t_0}} + \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}} \\
&\leq \max\{\sqrt{a}, 1\} \cdot (C_\Psi + 1)/c_L \cdot \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}} + \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}} \\
&= (\max\{\sqrt{a}, 1\} \cdot (C_\Psi + 1)/c_L + 1) \cdot \frac{|A(G^1_{t_0})|}{\Phi^1_{t_0+1}}.
\end{aligned}$$

That finishes the proof of statement (4.14).

□

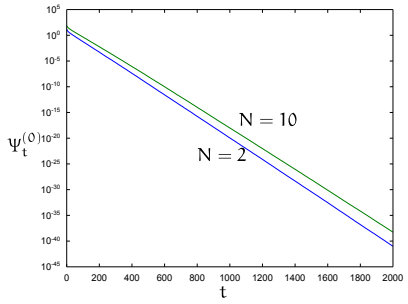
**The Right Amount of Potential**

The next step is to analyze the primary measure $\Psi^{(0)}_t$. The goal is to show that when the potential is neither too high nor too low, the primary measure indeed decreases on expectation by a factor of $1 - \delta_R$. This is not surprising and it can be clearly seen in experiments.

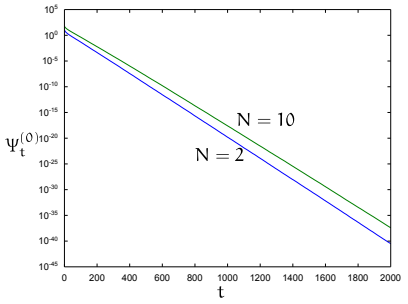
**Experiment 4.4.** We measure $\Psi^{(0)}_t$ for $N = 10$ particles processing the objective functions Sphere and Sphere$^+$ after being initialized with positions chosen randomly from $[-100, 100]$ for Sphere and from $[0, 100]$ for Sphere$^+$ and with swarm sizes $N = 2$ and $N = 10$. The results can be seen in Figure 4.11. Indeed, the primary measure decreases exponentially over time. Similar to the case of a too high potential, we can see that in case of the larger swarm size $\Psi^{(0)}_t$ decreases slightly slower.

While the claimed decrease of $\Psi^{(0)}_t$ is intuitively clear since it only says that the swarm indeed optimizes if it is not in a condition that prevents it from optimizing, the formal proof turns out to yield some peculiarities and a number of case distinctions is necessary.

Before we can verify 5. and 6., we state a rather technical lemma that shows how a particle swarm with an appropriate potential level can hit its goal. Its




**(a)** Particles processing objective function Sphere




**(b)** Particles processing objective function Sphere$^+$

**Figure 4.11:** Primary measure of a particle swarm processing the 1-dimensional objective functions Sphere and Sphere$^+$ with swarm sizes $N = 2$ and $N = 10$.

proof consists of the explicit construction of sets of sequences of steps, that have a sufficiently high probability and bring the particles into the area that is substantially better than the current attractors.

**Lemma 4.4.** Let $A = [a, b]$ be an interval containing 0, i. e., $a \leq 0 \leq b$. Define

$$d_L := \frac{1}{\chi} \cdot \frac{\max\left\{\frac{2\cdot(\chi+1)}{\chi\cdot(c_1-1)}, \frac{4\cdot\chi+2\cdot c_2-2}{\chi\cdot(c_2-1)}\right\}}{\min\{\chi, 1/3, (c_2-1)/6\}}.$$

Assuming that

$$d_L \cdot \sup_{x\in A(G_t^n)} \operatorname{dist}(x, A) \leq |V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \leq d_H \cdot |A|$$

for some $d_H > 0$, the probability for particle $n$ to hit $A$ within the next 3 steps is bounded from below by a constant $\operatorname{const}(d_H, c_1, c_2, \chi)$.

*Proof.* We divide the proof into three parts, depending on the number of steps necessary to hit $A$. First, we cover configurations that allow the swarm to hit $A$ within a single step, even with some less restrictive bounds for $|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n|$:

**Claim 4.1.** Given that

- $|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \leq \frac{5}{3} \cdot ((3 \cdot \max\{c_1, c_2\} + 1) \cdot d_H + 1) \cdot |A|$,

- $|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \geq \max\left\{\frac{2(\chi+1)}{\chi(c_1-1)}, \frac{4\chi+2c_2-2}{\chi(c_2-1)}\right\} \cdot \sup_{x \in A(G_t^n)} \operatorname{dist}(x, A)$ and
- $X_t^n \leq 0$ and $-(L_t^n - X_t^n) - \frac{c_2-1}{2} \cdot (G_t^n - X_t^n) \leq \chi \cdot V_t^n \leq (a - X_t^n) + |A|/2$ or $X_t^n \geq 0$ and $-(L_t^n - X_t^n) - \frac{c_2-1}{2} \cdot (G_t^n - X_t^n) \geq \chi \cdot V_t^n \geq (b - X_t^n) - |A|/2$,

the probability for particle $n$ hitting $A$ within the next step is bounded from below by a constant $\operatorname{const}(d_H, c_1, c_2, \chi)$.

*Proof.* Due to symmetry reasons, we can without loss of generality assume that $X_t^n \leq 0$ and $-(L_t^n - X_t^n) - \frac{c_2-1}{2} \cdot (G_t^n - X_t^n) \leq \chi \cdot V_t^n \leq (a - X_t^n) + |A|/2$. By straight-forward calculation, we can show that for any interval

$$I \subset [X_t^n + \chi \cdot V_t^n, X_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n)],$$

we have

$$P_t(X_{t+1}^n \in I) \geq \frac{|I|^2}{2 \cdot (c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n))^2}. \tag{4.18}$$

Therefore, all that remains to do is to find a sufficiently large $I$. Note that due to to unimodality of the objective function, the position of a particle can never be strictly between its local and the global attractor, i. e., for $X_t^n \leq 0$, the signs of $L_t^n - X_t^n$ and $G_t^n - X_t^n$ are both non-negative. Therefore, the area that can actually be reached by particle $n$ within its next step is

$$R := [X_t^n + \chi \cdot V_t^n, X_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n)],$$

so we need to bound $|R \cap A|$. First, we consider the case $V_t^n < 0$. In that case, the highest point reachable by particle $n$ is the interval

$$\begin{aligned}
&X_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n) \\
&= G_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + (c_2 - 1) \cdot (G_t^n - X_t^n) \\
&\geq G_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + (c_2 - 1) \cdot (G_t^n - X_t^n) \\
&\quad - \min\left\{\frac{\chi + c_1}{\chi + 1}, \frac{\chi + c_2 - 1}{\chi + (c_2 - 1)/2}\right\} \cdot \\
&\quad \cdot \Bigg(\underbrace{\chi \cdot V_t^n + (L_t^n - X_t^n) + \frac{c_2 - 1}{2} \cdot (G_t^n - X_t^n)}_{\geq 0 \text{ by assumption.}}\Bigg)
\end{aligned}$$

$$
\begin{aligned}
&\geq G_t^n + \chi \cdot \min\left\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\right\} \cdot |V_t^n| + \chi \cdot \frac{c_1-1}{\chi+1} \cdot (L_t^n - X_t^n) \\
&\quad + \chi \cdot \frac{c_2-1}{2\cdot\chi+c_2-1} \cdot (G_t^n - X_t^n) \\
&\geq G_t^n + \chi \cdot \min\left\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\right\} \cdot \\
&\quad \cdot (|V_t^n| + (L_t^n - X_t^n) + (G_t^n - X_t^n)) \\
&\geq G_t^n + \operatorname{dist}(G_t^n, A) + \\
&\quad + \chi/2 \cdot \min\left\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\right\} \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n)) \\
&= a + \chi/2 \cdot \min\left\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\right\} \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n)).
\end{aligned}
$$

This results in

$$
\begin{aligned}
A^* := &\left[a, a + \chi/2 \cdot \min\left\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\right\} \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n))\right] \\
&\cap [a, b].
\end{aligned}
$$

Note that $A^* \subset R \cap A$. Since

$$
|A^*| = \min\left\{|A|, \chi/2 \cdot \min\left\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\right\} \cdot (L_t^n + G_t^n - 2X_t^n)\right\},
$$

(4.18) implies that particle $n$ hits $A$ with probability at least

$$
\begin{aligned}
P_t(X_{t+1}^n \in A^*) &\geq \frac{|A^*|^2}{2\cdot(c_1\cdot(L_t^n - X_t^n) + c_2\cdot(G_t^n - X_t^n))^2} \\
&\geq \min\left\{\frac{|A|^2}{2\cdot(c_1\cdot(L_t^n - X_t^n) + c_2\cdot(G_t^n - X_t^n))^2}, \right. \\
&\qquad \left. \frac{(\chi/2 \cdot \min\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\} \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n)))^2}{2\cdot(c_1\cdot(L_t^n - X_t^n) + c_2\cdot(G_t^n - X_t^n))^2}\right\} \\
&\geq \min\left\{\frac{1}{\operatorname{const}(d_H, c_1, c_2, \chi)}, \frac{(\chi/2 \cdot \min\{\frac{c_1-1}{\chi+1}, \frac{c_2-1}{2\cdot\chi+c_2-1}\})^2}{2\cdot(c_1+c_2)^2}\right\}.
\end{aligned}
$$

That finishes the case $V_t^n < 0$. Next, we examine the case $V_t^n \geq 0$. We have

$$\begin{aligned}
|R \cap A| &= |[\max\{a, X_t^n + \chi \cdot V_t^n\}, \\
&\quad \min\{X_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n), b\}]| \\
&\geq \min\{|A|, |A|/2, X_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n) - a, \\
&\quad c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n)\}.
\end{aligned}$$

Since

$$\begin{aligned}
&X_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n) \\
&= G_t^n + \chi \cdot V_t^n + c_1 \cdot (L_t^n - X_t^n) + (c_2 - 1) \cdot (G_t^n - X_t^n) \\
&\geq G_t^n + \min\{\chi, c_1, (c_2 - 1)\} \cdot (V_t^n + (L_t^n - X_t^n) + (G_t^n - X_t^n)) \\
&\geq G_t^n + \min\{\chi, c_1, (c_2 - 1)\} \cdot (1/2 \cdot \max\left\{\frac{2 \cdot (\chi + 1)}{\chi \cdot (c_1 - 1)}, \frac{4 \cdot \chi + 2 \cdot c_2 - 2}{\chi \cdot (c_2 - 1)}\right\} \\
&\quad \cdot \operatorname{dist}(G_t^n, A) + 1/2 \cdot (V_t^n + (L_t^n - X_t^n) + (G_t^n - X_t^n))) \\
&\geq G_t^n + \min\{\chi, c_1, (c_2 - 1)\} \cdot (1/2 \cdot \max\{2/\chi, 2/c_1, 2/(c_2 - 1)\} \\
&\quad \cdot \operatorname{dist}(G_t^n, A) + 1/2 \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n))) \\
&\geq a + \min\{\chi, c_1, c_2 - 1\} \cdot 1/2 \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n)),
\end{aligned}$$

it follows that

$$\begin{aligned}
|R \cap A| \geq \min\{&|A|, |A|/2, \min\{\chi, c_1, c_2 - 1\}/2 \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n)), \\
&c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n)\}
\end{aligned}$$

and therefore, again using (4.18) and ignoring the first and the last term inside the minimum since the second, respectively the third, is clearly smaller, we obtain

$$\begin{aligned}
P_t(X_{t+1}^n \in A) &\geq \frac{(\min\{|A|/2, \min\{\chi, c_1, (c_2 - 1)\} \cdot 1/2 \cdot \cdots}{2 \cdot (c_1 \cdot (L_t^n - X_t^n) + \cdots} \\
&\quad \frac{\cdots \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n))\})^2}{\cdots + c_2 \cdot (G_t^n - X_t^n))^2} \\
&= \min\left\{\frac{(|A|/2)^2}{2 \cdot (c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n))^2}, \right. \\
&\quad \left. \frac{(\min\{\chi, c_1, (c_2 - 1)\} \cdot 1/2 \cdot ((L_t^n - X_t^n) + (G_t^n - X_t^n)))^2}{2 \cdot (c_1 \cdot (L_t^n - X_t^n) + c_2 \cdot (G_t^n - X_t^n))^2}\right\}
\end{aligned}$$

$$\geq \min\left\{ \frac{1/4}{(\frac{10}{3}\cdot((3\cdot\max\{c_1,c_2\}+1)\cdot d_H+1)\cdot\max\{c_1,c_2\})^2}, \right.$$
$$\left. \frac{(\min\{\chi, c_1, (c_2-1)\})^2}{4\cdot(c_1+c_2)^2} \right\}$$
$$=: \mathrm{const}(d_H, c_1, c_2, \chi).$$

This finishes the proof of the claim. □

Next, we consider the case of a velocity with large absolute value and pointing away from $A$. As it will turn out, with a constant probability the distances to the attractors will within one step increase sufficiently to yield a configuration satisfying the requirements of Claim 4.1.

**Claim 4.2.** Given that

- $|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \leq (3\cdot\max\{c_1,c_2\}+1)\cdot d_H\cdot|A|$,
- $|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \geq \dfrac{\max\left\{\frac{2\cdot(\chi+1)}{\chi\cdot(c_1-1)}, \frac{4\cdot\chi+2\cdot c_2-2}{\chi\cdot(c_2-1)}\right\}}{\min\{\chi, 1/3, (c_2-1)/6\}}\cdot$
  $\cdot \sup_{x\in A(G_t^n)} \mathrm{dist}(x, A)$ and
- $X_t^n \leq 0$ and $-(L_t^n - X_t^n) - \frac{c_2-1}{2}\cdot(G_t^n - X_t^n) \geq \chi\cdot V_t^n$ or $X_t^n \geq 0$ and $-(L_t^n - X_t^n) - \frac{c_2-1}{2}\cdot(G_t^n - X_t^n) \leq \chi\cdot V_t^n$,

the probability for particle $n$ hitting $A$ within the next 2 steps is bounded from below by a constant $\mathrm{const}(d_H, c_1, c_2, \chi)$.

*Proof.* Due to symmetry reasons, we can without loss of generality assume that $X_t^n \leq 0$ and $-(L_t^n - X_t^n) - \frac{c_2-1}{2}\cdot(G_t^n - X_t^n) \geq \chi\cdot V_t^n$. With probability $(c_2-1)/(18\cdot c_1\cdot c_2)$, the random values $r_{t+1}^n$ and $s_{t+1}^n$ satisfy $r_{t+1}^n \leq 1/(3\cdot c_1)$ and $s_{t+1}^n \leq (c_2-1)/(6\cdot c_2)$. Every such choice leads to

$$\begin{aligned}
|V_{t+1}^n| &\geq \chi\cdot|V_t^n| - |c_1\cdot r_{t+1}^n\cdot(L_t^n - X_t^n) + c_2\cdot s_{t+1}^n\cdot(G_t^n - X_t^n)| \\
&\geq \chi\cdot|V_t^n| + (L_t^n - X_t^n)/3 + (c_2-1)/6\cdot(G_t^n - X_t^n) \\
&\geq \min\{\chi, 1/3, (c_2-1)/6\}\cdot(|V_t^n| + (L_t^n - X_t^n) + (G_t^n - X_t^n)) \\
&\geq \max\left\{\frac{2\cdot(\chi+1)}{\chi\cdot(c_1-1)}, \frac{4\cdot\chi+2\cdot c_2-2}{\chi\cdot(c_2-1)}\right\}\cdot \sup_{x\in A(G_t^n)} \mathrm{dist}(x, A),
\end{aligned}$$

i. e., a configuration that satisfies the second condition of Claim 4.1. Additionally, since $0 \in A$ and $X_t^n < 0$ implies $L_{t+1}^n \geq X_t^n$, for every such choice of $r_{t+1}^n$ and $s_{t+1}^n$, it follows that $X_t^n \leq 0$, $V_t^n \leq 0$ and

$$\begin{aligned} -(L_{t+1}^n - X_{t+1}^n) - \frac{c_2 - 1}{2} \cdot (G_{t+1}^n - X_{t+1}^n) &\leq -(L_{t+1}^n - X_{t+1}^n) \\ &\leq -(X_t^n - X_{t+1}^n) = V_{t+1}^n \\ &\leq \chi \cdot V_{t+1}^n, \end{aligned}$$

which is sufficient for the third precondition of Claim 4.1. To finish the proof of the claim, it only remains to verify the first condition. From $L_{t+1}^n = L_t^n$ and

$$\begin{aligned} |G_{t+1}^n - X_{t+1}^n| &\leq |G_{t+1}^n - G_{t+1}^n| + |G_t^n - X_t^n| + |X_t^n - X_{t+1}^n| \\ &\leq |A(G_1^n)| + |G_1^n - X_t^n| + |V_{t+1}^n| \\ &\leq |A| + 2 \cdot \sup_{x \in A(G_t^n)} \operatorname{dist}(x, A) + |G_t^n - X_t^n| + |V_{t+1}^n| \\ &\leq |A| + 2 \cdot \frac{\min\{\chi, 1/3, (c_2 - 1)/6\}}{\max\left\{\frac{2 \cdot (\chi+1)}{\chi \cdot (c_1 - 1)}, \frac{4 \cdot \chi + 2 \cdot c_2 - 2}{\chi \cdot (c_2 - 1)}\right\}} \cdot (|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n|) \\ &\quad + |G_t^n - X_t^n| + |V_{t+1}^n| \\ &\leq |A| + \frac{2}{3} \cdot |V_t^n| + \frac{2}{3} \cdot |G_t^n - X_t^n| + \frac{2}{3} \cdot |L_t^n - X_t^n| + |V_{t+1}^n|, \end{aligned}$$

it follows that

$$\begin{aligned} &|V_{t+1}^n| + |G_{t+1}^n - X_{t+1}^n| + |L_{t+1}^n - X_{t+1}^n| \\ &\quad \leq |V_{t+1}^n| + |A| + \frac{2}{3} \cdot |V_t^n| + \frac{2}{3} \cdot |G_t^n - X_t^n| + \frac{2}{3} \cdot |L_t^n - X_t^n| \\ &\qquad + |V_{t+1}^n| + |L_t^n - X_t^n| + |V_{t+1}^n| \\ &\quad = |A| + \frac{2}{3} \cdot |V_t^n| + 3 \cdot |V_{t+1}^n| + \frac{2}{3} \cdot |G_t^n - X_t^n| + \frac{5}{3} \cdot |L_t^n - X_t^n| \\ &\quad \leq |A| + \frac{2 + 9 \cdot \chi}{3} \cdot |V_t^n| + \frac{2}{3} \cdot |G_t^n - X_t^n| + \frac{5}{3} \cdot |L_t^n - X_t^n| \\ &\quad \leq |A| + \frac{5}{3}(|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n|) \\ &\quad \leq \frac{5}{3} \cdot ((3 \cdot \max\{c_1, c_2\} + 1) \cdot d_H + 1) \cdot |A| \end{aligned}$$

Claim 4.1 finishes the proof. □

All that remains is the case of a velocity pointing towards 0 and (possibly) too high to hit $A$ directly. In that case, either the particle still hits $A$ or it jumps over $A$, resulting in the configuration of either Claim 4.1 or Claim 4.2.

**Claim 4.3.** Given that

- $|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \leq d_H \cdot |A|$,
- $|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \geq \dfrac{\max\left\{\frac{2\cdot(\chi+1)}{\chi\cdot(c_1-1)}, \frac{4\cdot\chi+2\cdot c_2-2}{\chi\cdot(c_2-1)}\right\}}{\chi \cdot \min\{\chi, 1/3, (c_2-1)/6\}}$ $\cdot \sup_{x\in A(G_{t+1}^n)} \operatorname{dist}(x, A)$ and
- $X_t^n \leq 0$ and $\chi \cdot V_t^n \leq (a - X_t^n) + |A|/2$ or $X_t^n \geq 0$ and $\chi \cdot V_t^n \geq (b - X_t^n) - |A|/2$,

the probability for particle $n$ hitting $A$ within the next 3 steps is bounded from below by a constant $\operatorname{const}(d_H, c_1, c_2, \chi)$.

*Proof.* Due to symmetry reasons, we can without loss of generality assume that $X_t^n \leq 0$ and $\chi \cdot V_t^n \geq (a - X_t^n) + |A|/2 \geq 0$, implying $X_{t+1}^n \geq a$. With probability $\frac{c_1-1}{c_1} \cdot \frac{c_2-1}{c_2}$, the random values $r_t^n$ and $s_t^n$ satisfy $r_t^n \geq 1/c_1$ and $s_t^n \geq 1/c_2$. Every such choice leads to

$$\begin{aligned}
&|V_{t+1}^n| + |G_{t+1}^n - X_{t+1}^n| + |L_{t+1}^n - X_{t+1}^n| \geq V_{t+1}^n \\
&\quad\geq\ \chi \cdot V_t^n + (G_t^n - X_t^n) + (L_t^n - X_t^n) \\
&\quad\geq\ \chi \cdot (V_t^n + (G_t^n - X_t^n) + (L_t^n - X_t^n)) \\
&\quad\geq\ \frac{\max\left\{\frac{2\cdot(\chi+1)}{\chi\cdot(c_1-1)}, \frac{4\cdot\chi+2\cdot c_2-2}{\chi\cdot(c_2-1)}\right\}}{\min\{\chi, 1/3, (c_2-1)/6\}} \cdot \sup_{x\in A(G_t^n)} \operatorname{dist}(x, A),
\end{aligned}$$

i. e., the second conditions of each, Claim 4.1 and Claim 4.2, are satisfied. As for the upper bound of the potential after the next step, note that both attractors are not further away from $X_t^n$ than their old positions. Consequently, we have

$$\begin{aligned}
&|V_{t+1}^n| + |G_{t+1}^n - X_{t+1}^n| + |L_{t+1}^n - X_{t+1}^n| \\
&\quad\leq\ |V_{t+1}^n| + |G_t^n - X_{t+1}^n| + |L_t^n - X_{t+1}^n| \\
&\quad\leq\ 3 \cdot |V_{t+1}^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n| \\
&\quad\leq\ 3 \cdot \chi |V_t^n| + (3 \cdot c_2 + 1) \cdot |G_t^n - X_t^n| + (3 \cdot c_1 + 1) \cdot |L_t^n - X_t^n| \\
&\quad\leq\ (3 \cdot \max\{c_1, c_2\} + 1) \cdot (|V_t^n| + |G_t^n - X_t^n| + |L_t^n - X_t^n|) \\
&\quad\leq\ (3 \cdot \max\{c_1, c_2\} + 1) \cdot d_H \cdot |A|.
\end{aligned}$$

Altogether, for $r_t^n \geq 1/c_1$ and $s_t^n \geq 1/c_2$, only one of two things can happen. Either $X_{t+1} \in A$, then the proof is finished. Or $X_{t+1}^n > b \geq 0$ and $V_{t+1}^n > 0$. In that case, depending on the exact value of $V_{t+1}^n$, the configuration satisfies the conditions of either Claim 4.1 or Claim 4.2. The results of these two claims finish the proof. □

Since every configuration satisfying the conditions of the Lemma also satisfies the conditions of one of the three claims, the proof of the lemma is finished.

□

Now that the technical prerequisites are prepared, it is time to verify 5. and 6. by applying Lemma 4.4 for a certain interval $A$ and verifying that the situation of $\overline{B}_{t_0}^H \cap \overline{B}_{t_0}^L$ guarantees that the conditions for Lemma 4.4 are fulfilled.

**Lemma 4.5.** There are constants $t_R' \in \mathbb{N}$ such that for every $t_0 \in \mathbb{N}$, every $t_R \geq t_0 + t_R'$ and every $c_L, c_H > 0$, there is a constant $\delta_R \in (0,1)$, depending only on $c_L$, $c_H$, $c_1$, $c_2$, $\chi$ and $N$, such that

$$E\left[\sum_{n=1}^{N} \sqrt{|A(L_{t_R}^n)|} \;\middle|\; \mathcal{F}_{t_0}, \overline{B}^H, \overline{B}^L\right] \leq (1-\delta_R) \cdot \sum_{n=1}^{N} \sqrt{|A(L_{t_0}^n)|}. \tag{4.19}$$

Furthermore, $\overline{B}^H$ and $\overline{B}^L$ together imply that

$$|A(G_{t_0}^1)|/\Phi_{t_0+1}^1 \leq \sqrt{N} \cdot (2 \cdot c_L + 1) \cdot \sum_{n=1}^{N} \sqrt{|A(L_{t_0}^n)|}, \tag{4.20}$$

$$\sum_{n=1}^{N} \sqrt{|A(L_{t_0}^n)|} \geq \Psi_{t_0}/(C_\Psi + c_H + \sqrt{N} \cdot (2 \cdot c_L + 1)). \tag{4.21}$$

*Proof.* For simplicity, we use the following notation:

$$E_R[\,\circ\,] := E\left[\,\circ \;\middle|\; \mathcal{F}_t, \overline{B}^H, \overline{B}^L\right],$$

analogous $P_R(\,\circ\,)$. Since $|A(L_t^n)|$ is non-decreasing in $t$, all needed to do is to prove that with constant probability $p_0$, the largest of the $|A(L_{t_0}^n)|$ decreases

by at least a constant factor of $1-\delta_0$ with $\delta_0 \in (0,1)$ within $t'_R$ steps. From there, it follows that

$$\begin{aligned}E_R\left[\sum_{n=1}^{N}\sqrt{|A(L^n_{t_R})|}\right] &\leq \sum_{n=1}^{N}\sqrt{|A(L^n_{t_0})|} - p_0\cdot(1-\sqrt{1-\delta_0})\cdot\max_n\sqrt{|A(L^n_{t_0})|}\\ &\leq (1-p_0\cdot(1-\sqrt{1-\delta_0})/N)\cdot\sum_{n=1}^{N}|A(L^n_{t_0})|.\end{aligned}$$

Since there are many substantially different configurations of the particles and the attractors, a lot of cases need to be considered. We divide the proof into two parts.

First, we consider the case when at least some local attractors are much worse than the global attractor, i. e.,

$$|A(G^1_{t_0})| < 1/(2\cdot N)\cdot\sum_{n=1}^{N}|A(L^n_{t_0})|.$$

Let $n_0$ be in $\operatorname{argmax}\{|A(L^n_{t_0})| \mid n \leq N\}$. It follows that $|A(G^1_{t_0})| < |A(L^{n_0}_{t_0})|/2$. The goal is to prove that with a constant probability, particle $n_0$ can halve its $|A(L^{n_0}_t)|$ within a constant number of steps. Let $A^*$ be the interval of size $|A(L^{n_0}_{t_0})|/2$, such that $|A(z)| \leq |A(L^{n_0}_{t_0})|/2$ for every $z \in A^*$, i. e., $A^*$ consists of the better half of $|A(L^{n_0}_t)|$. Since $|A(G^1_{t_0})| < |A(L^{n_0}_{t_0})|/2$, we have $A(G^1_{t_0}) \subset A^*$ and therefore $A(G^n_{t_0}) \subset A^*$, which leads to $\sup_{x\in A(G^n_{t_0})}\operatorname{dist}(x,A^*) = 0$ and no lower bound on the potential is needed to apply Lemma 4.4. As an upper bound, we obtain

$$\begin{aligned}&|V^n_{t_0}| + |G^n_{t_0} - X^n_{t_0}| + |L^n_{t_0} - X^n_{t_0}|\\ &\leq\ |V^n_{t_0}| + |G^n_{t_0} - G^1_{t_0}| + |G^1_{t_0} - X^n_{t_0}| + |L^n_{t_0} - G^1_{t_0}| + |G^1_{t_0} - X^n_{t_0}|\\ &\leq\ 2\cdot(|V^n_{t_0}| + |G^1_{t_0} - X^n_{t_0}|) + 2\cdot|A(L^n_{t_0})|\\ &\leq\ 2\cdot(\sqrt{|V^n_{t_0}|} + \sqrt{|G^1_{t_0} - X^n_{t_0}|})^2 + 2\cdot|A(L^n_{t_0})|\\ &\leq\ 2\cdot(c_H+1)\cdot|A(L^n_{t_0})| = 4\cdot(c_H+1)\cdot|A^*|.\end{aligned}$$

With Lemma 4.4, we obtain a lower bound of $\text{const}(c_H, c_1, c_2, \chi)$ for hitting $A^*$ within the following $t'_R := 3$ steps. That finishes the first case.

In the second case, we consider the situation when

$$|A(G^1_{t_0})| \geq 1/(2\cdot N)\cdot\sum_{n=1}^{N}|A(L^n_{t_0})|.$$

In that situation, it will turn out that the particles improve the global attractor, such that $|A(G)|$ decreases by at least a constant factor. This implies that also $\sum_{n=1}^{N} \sqrt{|A(L^n)|}$ decreases by a constant factor. While the upper bound on the potential implies a constant probability for improving the attractors at all, an additional lower bound on the potential is necessary to ensure that the improvement is significant. If $\sum_{n=1}^{N} Y_{t_0}^n \leq \sqrt{N \cdot |A(G_{t_0}^n)|}$, it follows that

$$\frac{|A(G_{t_0}^1)|}{\sum_{n=1}^{N} Y_{t_0}^n} \leq c_L \cdot \sum_{n=1}^{N} (Y_{t_0}^n + \sqrt{|A(L_{t_0}^n)|}) \leq c_L \cdot (\sqrt{N} + \sqrt{2 \cdot N}) \cdot \sqrt{|A(G_{t_0}^1)|}$$

and therefore

$$\sum_{n=1}^{N} (Y_{t_0}^n)^2 \geq \frac{1}{N} \cdot \left( \sum_{n=1}^{N} Y_{t_0}^n \right)^2 \geq \frac{|A(G_{t_0}^1)|}{c_L^2 \cdot (1+\sqrt{2})^2 \cdot N^2}.$$

Altogether, it follows that there is at least one particle $n$ with

$$|V_{t_0}^n| + |G_{t_0}^1 - X_{t_0}^n| \geq (Y_{t_0}^n)^2 \geq \frac{1}{N} \cdot \sum_{n=1}^{N} (Y_{t_0}^n)^2 \geq \frac{|A(G_{t_0}^1)|}{\max\{1, c_L^2 \cdot (1+\sqrt{2})^2\} \cdot N^2}.$$

Let $n_0$ be the first particle with

$$|V_{t_0}^{n_0}| + |G_{t_0}^1 - X_{t_0}^{n_0}| + |L_{t_0}^{n_0} - X_{t_0}^{n_0}| \geq \frac{|A(G_{t_0}^1)|}{\max\{N^2, c_L^2 \cdot (1+\sqrt{2})^2 \cdot N^2, 2(c_1+c_2)\}}.$$

Before examining the possible improvements made by particle $n_0$, we need to check by which amount its potential could be decreased during the steps of the first $n_0 - 1$ particles. Imagine that particle $n_0$ has a local attractor equal to its current position and a velocity close to 0, so all of its potential comes from the distance to the global attractor, which can be decreased during the steps of the other particles. I. e., if $G_{t_0}^{n_0} \approx X_{t_0}^{n_0}$, then the potential of particle $n_0$ is gone and there is no hope for a significant improvement during its step. However, if $G_{t_0}^1$ and $X_{t_0}^{n_0}$ have the same sign, then $G_{t_0}^1$ is already the closest point to $X_{t_0}^{n_0}$ from the whole interval $A(G_{t_0}^1)$, so any update of the global attractor increases the distance. If on the other hand $G_{t_0}^1$ and $X_{t_0}^{n_0}$ have different signs, any improvement of the global attractor does indeed decrease the distance $|G_{t_0}^1 - X_{t_0}^{n_0}|$ but, since for every particle $n < n_0$ we have

$$|V_{t_0}^n| + |G_{t_0}^1 - X_{t_0}^n| + |L_{t_0}^n - X_{t_0}^n| < |A(G_{t_0}^1)|/(2 \cdot c_1 + 2 \cdot c_2),$$

decreasing the distance too much is not possible without intermediate updates of the global attractor which reduce $|A(G)|$ considerably. More precisely: If $\operatorname{sign}(G_{t_0}^1) = -\operatorname{sign}(X_{t_0}^{n_0})$, then for any particle $n < n_0$, we have $\operatorname{sign}(X_{t_0}^n) = -\operatorname{sign}(X_{t_0}^{n_0})$, i. e., $G_{t_0}^1$ is between $X_{t_0}^n$ and $X_{t_0}^{n_0}$, because otherwise, we would have

$$|V_{t_0}^n| + |G_{t_0}^1 - X_{t_0}^n| + |L_{t_0}^n - X_{t_0}^n| \geq |G_{t_0}^1 - X_{t_0}^n| \geq |A(G_{t_0}^1)|,$$

a contradiction.

Additionally, if

$$|A(G_{t_0}^{n_0})| \geq |A(G_{t_0}^1)| \cdot (1 - 1/(4c_1 + 4c_2)),$$

then there is an interval $I$ of length

$$|A(G_{t_0}^{n_0})| \geq |A(G_{t_0}^1)| \cdot (1 - 1/(2c_1 + 2c_2))$$

between particle $n_0$ and all previous particles, that is not visited by any particle $n < n_0$. Now let $n$ be the first particle to jump over $I$, i. e.,

$$|G_{t_0}^n - G_{t_0}^1| \leq |A(G_{t_0}^1)|/(2c_1 + 2c_2)$$

and

$$|G_{t_0}^{n+1} - G_{t_0}^n| \geq |A(G_{t_0}^1)| \cdot (1 - 1/(2c_1 + 2c_2)).$$

For the maximal improvement $|G_{t_0}^{n+1} - G_{t_0}^n|$ of the global attractor during the step of particle $n$, that means

$$\begin{aligned}
&|G_{t_0}^{n+1} - G_{t_0}^n| \leq \chi \cdot |V_{t_0}^n| + c_1 \cdot |L_{t_0}^n - X_{t_0}^n| + (c_2 - 1) \cdot |G_{t_0}^n - X_{t_0}^n| \\
&\leq \chi \cdot |V_{t_0}^n| + c_1 \cdot |L_{t_0}^n - X_{t_0}^n| + (c_2 - 1) \cdot |G_{t_0}^1 - X_{t_0}^n| + (c_2 - 1) \cdot |G_{t_0}^n - G_{t_0}^1| \\
&\leq (c_1 + c_2) \cdot (|V_{t_0}^n| + |G_{t_0}^1 - X_{t_0}^n| + |L_{t_0}^n - X_{t_0}^n|) + (c_2 - 1) \cdot |G_{t_0}^n - G_{t_0}^1| \\
&\leq (c_1 + c_2) \cdot |A(G_{t_0}^1)|/(4c_1 + 4c_2) + (c_2 - 1) \cdot |A(G_{t_0}^1)|/(2c_1 + 2c_2) \\
&\leq 3/4 \cdot |A(G_{t_0}^1)| \\
&< |A(G_{t_0}^1)| \cdot (1 - 1/(2c_1 + 2c_2)).
\end{aligned}$$

That concludes the contradiction. In summary, we have shown that one of the following cases holds:

- $|A(G_{t_0}^{n_0})| < |A(G_{t_0}^1)| \cdot (1 - 1/(2c_1 + 2c_2))$ (before particle $n_0$'s step, some other particle has hit interval I),

- $|G_{t_0}^{n_0} - X_{t_0}^{n_0}| \geq |A(G_{t_0}^1)| \cdot (1 - 1/(2c_1 + 2c_2))$ (right before the step of particle $n_0$, I is still between its position and the global attractor),

- $|G_{t_0}^{n_0} - X_{t_0}^{n_0}| \geq |G_{t_0}^1 - X_{t_0}^{n_0}|$ (if $\operatorname{sign}(G_{t_0}^1) = \operatorname{sign}(X_{t_0}^{n_0})$).

In the first case, there is nothing left to prove, and in the second and the third case, we have

$$\begin{aligned} &|V_{t_0}^{n_0}| + |G_{t_0}^{n_0} - X_{t_0}^{n_0}| + |L_{t_0}^{n_0} - X_{t_0}^{n_0}| \\ &\geq \frac{|A(G_{t_0}^1)|}{\max\{N^2, c_L^2 \cdot (1+\sqrt{2})^2 \cdot N^2, 2 \cdot c_1 + 2 \cdot c_2\}} = \frac{|A(G_{t_0}^1)|}{\text{const}} \end{aligned}$$

For $A(G_{t_0}^1) =: [a, b]$, choose

$$A^* := \left[ \min\left\{ a + \frac{|A(G_{t_0}^1)|}{\text{const} \cdot d_L}, 0 \right\}, \max\left\{ b - \frac{|A(G_{t_0}^1)|}{\text{const} \cdot d_L}, 0 \right\} \right],$$

where $d_L$ denotes the constant from Lemma 4.4. I. e., $A^*$ is obtained from $A(G_{t_0}^1)$ by leaving out an area of size $|A(G_{t_0}^1)|/(\text{const} \cdot d_L)$ at each boundary. Since that way 0 might be excluded from $A^*$, 0 and the points in between are added again. Note that the construction makes sure that after hitting $A^*$, the area that can be entered by the global attractor has size at most $|A(G_{t_0}^1)| \cdot (1 - 1/(\text{const} \cdot d_L))$ and that the lower bound as demanded by Lemma 4.4 holds.

Since $|G_{t_0}^{n_0} - G_{t_0}^1| \leq |A(G_{t_0}^1)|$, the upper bound of

$$\begin{aligned} &|V_{t_0}^{n_0}| + |G_{t_0}^{n_0} - X_{t_0}^{n_0}| + |L_{t_0}^{n_0} - X_{t_0}^{n_0}| \leq c_H \cdot |A(L_{t_0}^n)| + |A(G_{t_0}^1)| \\ &\leq (4 \cdot N \cdot c_H + 1) \cdot |A(G_{t_0}^1)| \leq \frac{\text{const} \cdot (4 \cdot N \cdot c_H + 1)}{\text{const} - 2} \cdot |A^*|. \end{aligned}$$

holds. Lemma 4.4 guarantees a hitting probability, which is bounded from below by a constant $\text{const}(c_H, N, c_1, c_2, \chi)$. That finishes the second case and therefore the proof of (4.19).

For the proof of (4.20), we use that $\sum_{n=1}^{N} Y_{t_0}^n \geq \sum_{n=1}^{N} \sqrt{|A(L_{t_0}^n)|}$ implies that

$$\begin{aligned}\frac{|A(G_{t_0}^1)|}{\Phi_{t_0+1}^1} &\leq \frac{\sqrt{N}\cdot|A(G_{t_0}^1)|}{\sum_{n=1}^{N} Y_{t_0}^n} \\ &\leq \frac{\sqrt{N}\cdot\left(\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}\right)^2}{\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}} \\ &= \sqrt{N}\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}.\end{aligned}$$

On the other hand, if $\sum_{n=1}^{N} Y_{t_0}^n \leq \sum_{n=1}^{N} \sqrt{|A(L_{t_0}^n)|}$, we have

$$\begin{aligned}\frac{|A(G_{t_0}^1)|}{\Phi_{t_0+1}^1} &\leq \frac{\sqrt{N}\cdot|A(G_{t_0}^1)|}{\sum_{n=1}^{N} Y_{t_0}^n} \\ &\leq \sqrt{N}\cdot c_L\cdot\sum_{n=1}^{N}(Y_{t_0}^n + \sqrt{|A(L_{t_0}^n)|}) \\ &\leq 2\cdot\sqrt{N}\cdot c_L\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}.\end{aligned}$$

That proves (4.20). Finally, we have that

$$\begin{aligned}\Psi_{t_0} &= C_\Psi\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|} + \sum_{n=1}^{N}\left(\sqrt{|V_{t_0}^n|} + \sqrt{|G_{t_0}^1 - X_t^n|}\right) + \frac{|A(G_{t_0}^1)|}{\Phi_{t_0+1}^1} \\ &\leq C_\Psi\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|} + c_H\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|} \\ &\quad + \sqrt{N}\cdot(2\cdot c_L + 1)\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|} \\ &\leq (C_\Psi + c_H + \sqrt{N}\cdot(2\cdot c_L + 1))\cdot\sum_{n=1}^{N}\sqrt{|A(L_{t_0}^n)|}.\end{aligned}$$

Statement (4.21) follows immediately.

□

### 4.2.3 Putting things together

In this section, we combine the technical results from the previous sections to verify the drift condition of Theorem 3.7 for the concrete choice of a distance measure $\Psi_t$ as stated in Definition 4.5.

**Lemma 4.6.** There are constants $t_{\max} \in \mathbb{N}$ and $\delta \in (0,1)$, such that for every $t \in \mathbb{N}$, there is an $\mathcal{F}_t$-measurable, $\mathbb{N}$-valued random variable $\sigma(t)$ with $t < \sigma(t) < t + t_{\max}$ almost surely, such that

$$\mathbf{1}_{\{\Psi_t > 0\}} \cdot \mathrm{E}[\Psi_{\sigma(t)} \mid \mathcal{F}_t] \leq \mathbf{1}_{\{\Psi_t > 0\}} \cdot \Psi_t \cdot (1-\delta).$$

*Proof.* For $t \in \mathbb{N}$, we consider three cases. The first case is the one of Lemma 4.2, i. e., when

$$\sum_{n=1}^{N} Y_t^n \geq c_H \sum_{n=1}^{N} \sqrt{|A(L_t^n)|}$$

for some $c_H \geq c'_H$ to be fixed later. In that case, we set $\sigma(t) := t + t'_H$ with $t'_H$ as in Lemma 4.2 and obtain

$$\begin{aligned}
\mathrm{E}[\Psi_{\sigma(t)} \mid \mathcal{F}_t] &= C_\Psi \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_{t+t'_H})|} + \sum_{n=1}^{N} Y^n_{t+t'_H} + \frac{|A(G^1_{t+t'_H})|}{\Phi^1_{t+t'_H+1}} \\
&\overset{(4.3),(4.11)}{\leq} C_\Psi \cdot \sum_{n=1}^{N} \sqrt{|A(L_t^n)|} + (1-\delta_H) \cdot \sum_{n=1}^{N} Y_t^n + C_L^{t'_H} \cdot \frac{|A(G_t^1)|}{\Phi^1_{t+1}} \\
&= \Psi_t - \delta_H \cdot \sum_{n=1}^{N} Y_t^n + (C_L^{t'_H} - 1) \cdot \frac{|A(G_t^1)|}{\Phi^1_{t+1}} \\
&\overset{(4.4),(4.5)}{\leq} \Psi_t - \frac{\delta_H \cdot \Psi_t}{1 + (C_\Psi + \mathrm{const}_H)/c_H} + (C_L^{t'_H} - 1) \cdot \frac{\mathrm{const}_H \cdot \Psi_t}{c_H} \\
&= \left(1 - \frac{c_H \cdot \delta_H}{c_H + C_\Psi + \mathrm{const}_H} + (C_L^{t'_H} - 1) \cdot \frac{\mathrm{const}_H}{c_H}\right) \cdot \Psi_t. \qquad (4.22)
\end{aligned}$$

In the second case, i. e., in the situation of Lemma 4.3, when

$$|A(G_t^1)| / \Big( \sum_{n=1}^{N} Y_t^n \Big) \geq c_L \cdot \sum_{n=1}^{N} (Y_t^n + \sqrt{|A(L_t^n)|}),$$

we set $\sigma(t) := \sigma_L$ with $\sigma_L$ as in Lemma 4.3 and obtain

$$\begin{aligned}
E[\Psi_{\sigma(t)} \mid \mathcal{F}_t] &= C_\Psi \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_{\sigma(t)})|} \\
&\quad + \sum_{n=1}^{N} \Big( \sqrt{|V^n_{\sigma(t)}|} + \sqrt{|G^1_{\sigma(t)} - X^n_{\sigma(t)}|} \Big) + \frac{|A(G^1_{\sigma(t)})|}{\Phi^1_{\sigma(t)}} \\
&\overset{(4.2),(4.12)}{\leq} C_\Psi \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} + C_H \cdot \sum_{n=1}^{N} (Y^n_t + \sqrt{|A(L^n_t)|}) \\
&\quad + (1 - \delta_L) \cdot \frac{|A(G^1_t)|}{\Phi^1_{t+1}} \\
&= \Psi_t + (C_H - 1) \cdot \sum_{n=1}^{N} (Y^n_t + \sqrt{|A(L^n_t)|}) - \delta_L \cdot \frac{|A(G^1_t)|}{\Phi^1_{t+1}} \\
&\overset{(4.13),(4.14)}{\leq} \Psi_t + (C_H - 1) \cdot \frac{const_L}{c_L} \cdot \Psi_t + \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} \\
&\quad - \frac{\delta_L \cdot \Psi_t}{1 + const_L \cdot C_\Psi / c_L} \\
&\leq \Psi_t + (C_H - 1) \cdot \frac{const_L}{c_L} \cdot \Psi_t + \frac{\Psi_t}{C_\Psi} - \frac{\delta_L \cdot \Psi_t}{1 + const_L \cdot C_\Psi / c_L} \\
&= \left( 1 + (C_H - 1) \cdot \frac{const_L}{c_L} + \frac{1}{C_\Psi} \right. \\
&\quad \left. - \frac{c_L \cdot \delta_L}{c_L + const_L \cdot C_\Psi} \right) \cdot \Psi_t .
\end{aligned} \tag{4.23}$$

Finally, in the situation of Lemma 4.5, when

$$\sum_{n=1}^{N} Y^n_t < c_H \sum_{n=1}^{N} \sqrt{|A(L^n_t)|}$$

and

$$|A(G^1_t)| / \Big( \sum_{n=1}^{N} Y^n_t \Big) < c_L \cdot \sum_{n=1}^{N} \Big( Y^n_t + \sqrt{|A(L^n_t)|} \Big),$$

we set $\sigma(t) := t + t'_R$ with $t'_R$ as in Lemma 4.5 and obtain

$$\begin{aligned}
\mathrm{E}[\Psi_{\sigma(t)} \mid \mathcal{F}_t] &= C_\Psi \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_{\sigma(t)})|} + \sum_{n=1}^{N} \left( \sqrt{|V^n_{\sigma(t)}|} + \sqrt{|G^1_{\sigma(t)} - X^n_{\sigma(t)}|} \right) \\
&\quad + \frac{|A(G^1_{\sigma(t)})|}{\Phi^1_{\sigma(t)}} \\
&\overset{(4.2),(4.11),(4.19)}{\leq} C_\Psi \cdot (1 - \delta_R) \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} + C_H \cdot \sum_{n=1}^{N} \left( Y^n_t + \sqrt{|A(L^n_t)|} \right) \\
&\quad + C_L^{t'_R} \cdot \frac{|A(G^1_t)|}{\Phi^1_{t+1}} \\
&= \Psi_t - (C_\Psi \cdot \delta_R - C_H) \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} + (C_H - 1) \cdot \sum_{n=1}^{N} Y^n_t \\
&\quad + (C_L^{t'_R} - 1) \cdot \frac{|A(G^1_t)|}{\Phi^1_{t+1}} \\
&\leq \Psi_t - (C_\Psi \cdot \delta_R - C_H) \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} \\
&\quad + (C_H - 1) \cdot c_H \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} + (C_L^{t'_R} - 1) \cdot \frac{|A(G^1_t)|}{\Phi^1_{t+1}} \\
&\overset{(4.20)}{\leq} \Psi_t - (C_\Psi \cdot \delta_R - C_H) \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} \\
&\quad + (C_H - 1) \cdot c_H \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} \\
&\quad + (C_L^{t'_R} - 1) \cdot \sqrt{N} \cdot (2 \cdot c_L + 1) \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|} \\
&= \Psi_t + \big( (C_H - 1) \cdot c_H + (C_L^{t'_R} - 1) \cdot \sqrt{N} \cdot (2 \cdot c_L + 1) \\
&\quad - (C_\Psi \cdot \delta_R - C_H) \big) \cdot \sum_{n=1}^{N} \sqrt{|A(L^n_t)|}.
\end{aligned}$$

For any choice of the occurring constants that satisfies

$$(C_H - 1) \cdot c_H + (C_L^{t'_R} - 1) \cdot \sqrt{N} \cdot (2 \cdot c_L + 1) \leq (C_\Psi \cdot \delta_R - C_H),$$

it follows by (4.21), that

$$\begin{aligned} \mathrm{E}[\Psi_{\sigma(t)} \mid \mathcal{F}_t] &\leq 1 + \cdots \\ &\cdots + \frac{(C_H - 1) \cdot c_H + (C_L^{t'_R} - 1) \cdot \sqrt{N} \cdot (2 \cdot c_L + 1) - (C_\Psi \cdot \delta_R - C_H)}{C_\Psi + c_H + \sqrt{N} \cdot (2 \cdot c_L + 1)} \cdot \Psi_t \end{aligned} \tag{4.24}$$

All that is left to do is to fix the constants $c_H$, $c_L$ and $C_\Psi$ in order to ensure that the factors before the $\Psi_t$ in (4.22), (4.23) and (4.24) are less than 1. Therefore, we make the choices

$$\begin{aligned} c_H := c_L := \max \Bigg\{ & c'_H, \mathrm{const}_H, 2 \cdot \frac{\mathrm{const}_H \cdot (2 + \hat{C}_\Psi) \cdot (C_L^{t'_H} - 1)}{\delta_H}, \\ & c'_L, 2 \cdot \frac{((C_H - 1) \cdot \mathrm{const}_L + 1/\hat{C}_\Psi) \cdot (1 + \mathrm{const}_L \cdot \hat{C}_\Psi)}{\delta_L}, 1, C_H \Bigg\}, \end{aligned}$$

and $C_\Psi := \hat{C}_\Psi \cdot c_H$ with a constant $\hat{C}_\Psi$ that we will fix later. For the first case, (4.22) leads to

$$\begin{aligned} \mathrm{E}[\Psi_{\sigma(t)} \mid \mathcal{F}_t] &\leq \left(1 - \frac{c_H \cdot \delta_H}{c_H + C_\Psi + \mathrm{const}_H} + (C_L^{t'_H} - 1) \cdot \frac{\mathrm{const}_H}{c_H}\right) \cdot \Psi_t \\ &= \left(1 - \frac{\delta_H}{1 + \hat{C}_\Psi + \mathrm{const}_H / c_H} + (C_L^{t'_H} - 1) \cdot \frac{\mathrm{const}_H}{c_H}\right) \cdot \Psi_t \\ &\leq \left(1 - \frac{\delta_H}{2 + \hat{C}_\Psi} + (C_L^{t'_H} - 1) \cdot \frac{\mathrm{const}_H}{c_H}\right) \cdot \Psi_t \\ &\leq \left(1 - \frac{\delta_H}{2 + \hat{C}_\Psi} + \frac{\delta_H}{2 \cdot (2 + \hat{C}_\Psi)}\right) \cdot \Psi_t \\ &\leq \left(1 - \underbrace{\frac{\delta_H}{2 \cdot (2 + \hat{C}_\Psi)}}_{=:\delta_1}\right) \cdot \Psi_t. \end{aligned}$$

In a similar way, for the second case (4.23) leads to

$$\begin{aligned}
\mathrm{E}[\Psi_{\sigma(t)} \mid \mathcal{F}_t] &\leq \left(1 + (C_H - 1) \cdot \frac{\mathrm{const}_L}{c_L} + \frac{1}{C_\Psi} - \frac{c_L \cdot (\delta_L)}{c_L + \mathrm{const}_L \cdot C_\Psi}\right) \cdot \Psi_t \\
&= \left(1 + \frac{(C_H - 1) \cdot \mathrm{const}_L + 1/\hat{C}_\Psi}{c_L} - \frac{\delta_L}{1 + \mathrm{const}_L \cdot \hat{C}_\Psi}\right) \cdot \Psi_t \\
&\leq \left(1 + \frac{\delta_L}{2 \cdot (1 + \mathrm{const}_L \cdot \hat{C}_\Psi)} - \frac{\delta_L}{1 + \mathrm{const}_L \cdot \hat{C}_\Psi}\right) \cdot \Psi_t \\
&= \left(1 - \underbrace{\frac{\delta_L}{2 \cdot (1 + \mathrm{const}_L \cdot \hat{C}_\Psi)}}_{=:\delta_2}\right) \cdot \Psi_t.
\end{aligned}$$

Finally, we set

$$\hat{C}_\Psi := 2 \cdot \frac{C_H + 3 \cdot (C_L^{t'_R} - 1) \cdot \sqrt{N}}{\delta_R}$$

and obtain for the third case from (4.24)

$$\begin{aligned}
&\mathrm{E}[\Psi_{\sigma(t)} \mid \mathcal{F}_t] \leq \Psi_t \cdot \Bigg(1 + \cdots \\
&\quad \cdots + \frac{(C_H - 1) \cdot c_H + (C_L^{t'_R} - 1) \cdot \sqrt{N} \cdot (2 \cdot c_L + 1) - (C_\Psi \cdot \delta_R - C_H)}{C_\Psi + c_H + \sqrt{N} \cdot (2 \cdot c_L + 1)}\Bigg) \\
&= \Psi_t \cdot \Bigg(1 + \cdots \\
&\quad \cdots \frac{(C_H - 1) \cdot c_H + (C_L^{t'_R} - 1) \cdot \sqrt{N} \cdot (2 \cdot c_H + 1) - (\hat{C}_\Psi \cdot c_H \cdot \delta_R - C_H)}{\hat{C}_\Psi \cdot c_H + c_H + \sqrt{N} \cdot (2 \cdot c_H + 1)}\Bigg) \\
&\leq \Psi_t \cdot \left(1 + \frac{(C_H - 1) + 3 \cdot (C_L^{t'_R} - 1) \cdot \sqrt{N} - \hat{C}_\Psi \cdot \delta_R + 1}{\hat{C}_\Psi + 1 + 2 \cdot \sqrt{N}}\right) \\
&= \Psi_t \cdot \left(1 + \frac{C_H + 3 \cdot (C_L^{t'_H} - 1) \cdot \sqrt{N} - 2 \cdot (C_H + 3 \cdot (C_L^{t'_R} - 1) \cdot \sqrt{N})}{\hat{C}_\Psi + 1 + 3 \cdot \sqrt{N}}\right) \\
&= \Psi_t \cdot \left(1 - \underbrace{\frac{C_H + 3 \cdot (C_L^{t'_R} - 1) \cdot \sqrt{N}}{2 \cdot (C_H + 3 \cdot (C_L^{t'_R} - 1) \cdot \sqrt{N})/\delta_R + 1 + 3 \cdot \sqrt{N}}}_{=:\delta_3}\right).
\end{aligned}$$

With $\delta := \min\{\delta_1, \delta_2, \delta_3\}$ and $t_{max} := \max\{t'_H, t'_L, t'_R\}$ the proof is finished. □

Now, everything is prepared to apply our drift theorem, Theorem 3.7.

**Theorem 4.2.** Let $b$ be the diameter of the search space and assume that the particles are initialized such that $E[\sum_{n=1}^{N} Y_t^n] \leq C_Y \cdot \sqrt{b}$ and $E[1/\Phi_{t+1}^1] \leq C_\Phi$ for two constants $C_Y, C_\Phi > 0$. This is the case, e. g., when the particles' positions are initialized independently and uniformly over $[-b, b]$ and if the velocities have finite expectation. Then there is a constant $c$, depending only on the swarm parameters $\chi$, $c_1$, $c_2$ and $N$, such that the following holds: Let $\tau := \min\{t \geq 0 \mid \Psi_t \leq 2^{-k}\}$. Then we have

$$E[\tau] \leq \text{const}(\chi, c_1, c_2, N, a, C_Y, C_\Phi) \cdot (k + \log(b+1) + 1).$$

*Proof.* We define $Z_t := 2^k \cdot \Psi_t$. Then, Lemma 4.6 implies that for every $t \in \mathbb{N}$

$$\mathbf{1}_{\{Z_t > 1\}} \cdot E[Z_{\sigma(t)} \mid \mathcal{F}_t] \leq \mathbf{1}_{\{Z_t > 1\}} \cdot Z_t \cdot (1 - \delta)$$

for some $\delta \in (0, 1)$ and some $\mathcal{F}_t$-measurable $\sigma(t)$ with $t \leq \sigma(t) \leq t + t_{max} \in \mathbb{N}$. Note that $Z_t \leq 1 \Leftrightarrow \Psi_t \leq 2^{-k}$. By Theorem 3.7, it follows that

$$E[\tau \mid \mathcal{F}_0] \leq t_{max} \cdot \mathbf{1}_{\{Z_0 > 1\}} \cdot (\log(Z_0) + 2)/\delta$$

and therefore

$$E[\tau] = E[E[\tau \mid \mathcal{F}_0]] \leq E[t_{max} \cdot \mathbf{1}_{\{Z_0>1\}} \cdot (\ln(Z_0)+2)/\delta] \leq t_{max} \cdot (k \cdot \log(E[\Psi_0]) + 2)/\delta.$$

As for the expectation of $\Psi$ right after the initialization, we have

$$\begin{aligned} E[\Psi_0] &= E\left[C_\Psi \cdot \sum_{n=1}^{N} \sqrt{|A(L_t^n)|} + \sum_{n=1}^{N} Y_t^n + \frac{|A(G_t^1)|}{\Phi_{t+1}^1}\right] \\ &\leq C_\Psi \cdot N \cdot \sqrt{b} + C_Y \cdot \sqrt{b} + C_\Phi \cdot b \\ &\leq (C_\Psi \cdot N + C_Y + C_\Phi) \cdot \max\{b, \sqrt{b}\} \end{aligned}$$

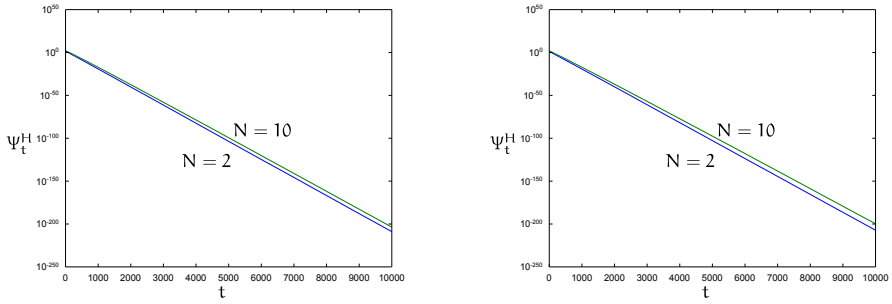
and therefore

$$\begin{aligned} E[\tau] &\leq t_{max} \cdot (\ln(2) \cdot k + \log(C_\Psi \cdot N + C_Y + C_\Phi) + \log(\max\{b, \sqrt{b}\}) + 2)/\delta \\ &\leq \underbrace{t_{max}/\delta \cdot \max\{1, \log(C_\Psi \cdot N + C_Y + C_\Phi) + 2\}}_{=:c} \cdot (k + \log(b+1) + 1) \end{aligned}$$

That finishes the proof. □

In order to provide some experimental insight as well, we calculated the complete measure $\Psi$ with $C_\Psi := 1$ for swarms processing the objective functions Sphere and Sphere$^+$ in the following experiment.

**Experiment 4.5.** As before, the swarm sizes $N = 2$ and $N = 10$ are tested and the particles are initialized over $[-100, 100]$ for processing Sphere and $[0, 100]$ for optimizing Sphere$^+$. The results can be seen in Figure 4.12.




**(a)** Optimizing Sphere. **(b)** Optimizing Sphere$^+$.

**Figure 4.12:** Course of optimality measure $\Psi$ while processing objective function Sphere and Sphere$^+$ for swarm sizes $N = 2$ and $N = 10$.

Actually, increasing the swarm size does in the 1-dimensional situation not increase the speed by which $\Psi$ decreases. The reason for this might be that the run time consists mostly of the phases when the potential needs to be decreased, which has to be done by every particle independently of the others and can therefore not be accelerated by increasing the swarm size. Therefore, the experiments indicate that the smallest $N$, for which the swarm converges towards the optimum at all, might be already the optimal choice for the swarm size.

# 5. Convergence for Multidimensional Problems

In the previous chapter, we have demonstrated the power of the bad-events-technique to analyze the classical particle swarm optimization (PSO) in case of 1-dimensional objective functions. We studied two bad events, namely the event of a potential too high and the event of a potential too low to allow a sufficiently large expected improvement of the global attractor within near future. We have been shown that without any modification, PSO can heal itself from encountering any of the two bad events. This implies that the swarm can find local optima and furthermore, if the objective function is unimodal, we were able to prove a runtime bound.

As it will turn out soon, we cannot achieve the same results in the more general $D$-dimensional case. Here, certain bad events exist from which the swarm might not be able to heal itself. But still, studying the bad events is fruitful because the investigations uncover the possible weaknesses of the swarm and allow for a distinction between ordinary bad events, from which the particle swarm recovers on its own, and "fatal events" that might indeed prevent the swarm from finding a local optimum and therefore make intervention necessary.

In the first part of this chapter, we expose the bad events of the $D$-dimensional case and empirically examine them for if they are fatal or not. Although the complicated interdependencies between different dimensions prevent a formal run time analysis, we suggest appropriate informal secondary measures based on the potential and present experimental investigations of these measures during the respective bad events. Afterwards, we apply a modification to the PSO algorithm, enabling it to heal itself from the former fatal events. We prove that the resulting method almost surely finds local optima, similar to the 1-dimensional situation.

Finally, we present experiments using standard double precision numbers to demonstrate that the examined phenomena are not just artifacts of the implementation with arbitrary precision numbers, but that they show up and effect the quality of the solution found even in standard situations.

## 5.1 Determining Further Bad Events

There are three essentially different bad events, i. e., three different conditions on the configuration of the swarm, preventing major improvements of the attractors. These three bad events lead to a very different behavior of the particle swarm and will be examined in detail during this section.

The first bad event occurs when in at least one dimension $d_0$, the swarm has a too high potential, such that the probability for updating the attractor at all is small. I.e., this happens when in dimension $d_0$ the potential is orders of magnitude larger than the area that would improve the attractors, while the potential in other dimensions, whose corresponding entries of the global attractor could be improved, is much smaller than in dimension $d_0$. Therefore, the overall value of the objective function depends almost only on the entries in dimensions like $d_0$ with too high potential and it is unlikely that the swarm updates the global attractor. If this bad event occurs, which can be seen as the $D$-dimensional generalization of the bad event $B_t^H$ from the previous chapter, the swarm needs to decrease its potential in dimension $d_0$ to heal itself.

Second and similar to the event $B_t^L$ from the previous chapter, configurations could occur in which the whole swarm has in every dimension a potential much smaller than the distance to the next local optimum. In that case, the desired behavior of the swarm is to accelerate and to charge potential.

Finally, there is a bad event which can only occur if $D \geq 2$, namely the case of very imbalanced potentials. If the potential is very imbalanced, i. e., if there is a dimension $d_0$ with a potential much smaller than in the other dimensions, then the $d_0$'th entry of a position has almost no influence on the decision whether an attractor gets updated after the next step or not. If additionally the $d_0$'th entry of the global attractor differs much from the corresponding entry of the next local optimum, the swarm is unlikely to decrease that distance and might behave like in the $D-1$-dimensional situation, optimizing all dimensions except for $d_0$. In order to recover from this imbalanced state, the swarm must "turn around the corner", i. e., the quotient $\Phi_t^{d_0} / \sum_{d=1}^{D} \Phi_t^d$ must increase.

As it turns out, the swarm can heal itself from the first two bad events easily, while the third bad event might become fatal even when PSO processes the very simple function Sphere. Therefore, we study the third event in detail to find a modification that allows the swarm to recover from it.

Throughout this section, we assume that the objective function has the form $f(x) = x^t \cdot A \cdot x$ with a positive-definite matrix $A$. While this may sound very restrictive, every analytic function $g$ can near a simple local optimum, i. e., a local optimum where all second derivatives are non-zero, be approximated by such a function $f$, where $A$ is the Jacobi-matrix of $g$.

### 5.1.1 High Potential in at least one Dimension: A Bad Event

If the potential in a certain dimension $d$ is too large, the behavior of the swarm is very similar to the 1-dimensional case. Let

$$\pi_d(A) := \{x_d \in \mathbb{R} \mid \exists x_1, \dots, x_{d-1}, x_{d+1}, \dots, x_D \in \mathbb{R} : (x_1, \dots, x_D) \in A\}$$

denote the projection of some set $A \subset \mathbb{R}^D$ on the $d$-axis. Then, a necessary condition for an improvement of the local attractor of particle $n$ is that $X_t^{n,d} \in \pi_d(A_{L_t^n})$. Now, we can formulate the bad event of a too high potential as

$$B_t^H := \left\{ \exists d \in \{1, \dots, D\} : \sum_{n=1}^{N} Y_t^{n,d} \geq c_H \cdot \sum_{n=1}^{N} \sqrt{|\pi_d(A_{L_t^n})|} \right\}$$

with

$$Y_t^{n,d} = \sqrt{|V_t^{n,d}|} + \sqrt{|G_t^{l,d} - X_t^{n,d}|}$$

as an alternative measure for the potential, similar to the 1-dimension case (Definition 3.9). A graphical representation of this bad event $B_t^H$ is given in Figure 5.1.

Similar to the 1-dimensional case, the $Y_t^{n,d}$ can serve as a secondary measure for this event, in terms:

$$\Psi_t^H := \sum_{d=1}^{D} \sum_{n=1}^{N} Y_t^{n,d}.$$

In order to examine how the particle swarm performs when confronted with such a configuration, Experiment 4.2 from the previous chapter is repeated with more than 1 dimension. Since in [LW11] the function $\textsc{Sphere}^+$ is only defined as a 1-dimensional function, we define its $D$-dimensional generalization as

$$\textsc{Sphere}^+((x_1, \dots, x_D)) := \begin{cases} \textsc{Sphere}((x_1, \dots, x_D)), & \text{if } \min_{d=1,\dots,D} x_d \geq 0, \\ \infty, & \text{otherwise,} \end{cases}$$

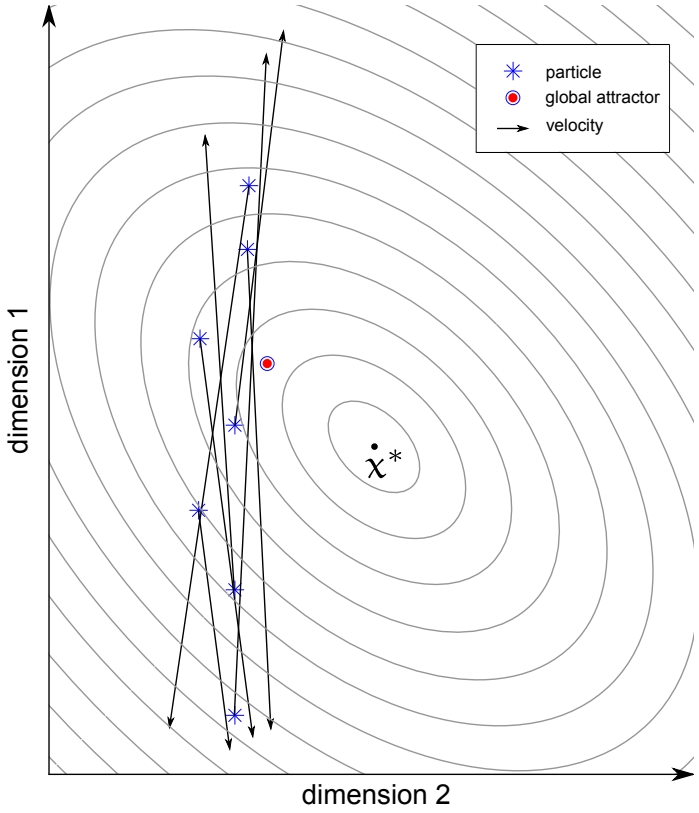
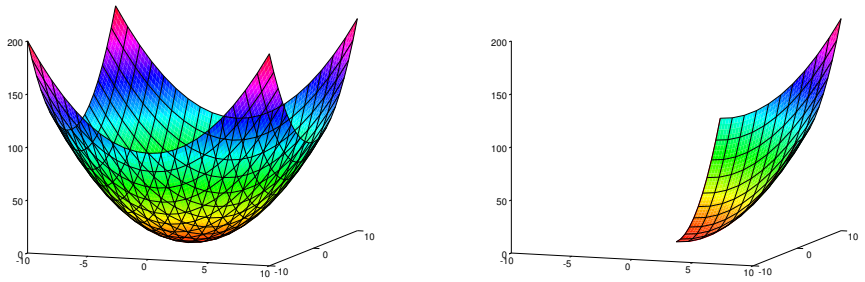



**Figure 5.1:** Dimension 1 has a too high potential and voids possible improvements in dimension 2.

i. e., $\text{SPHERE}^+(x)$ is identical to $\text{SPHERE}(x)$ if all the coordinate entries of $x$ are non-negative. Otherwise, $\text{SPHERE}^+(x)$ is infinite. Figure 5.2 shows the comparison of the functions SPHERE and $\text{SPHERE}^+$. Similar as before, SPHERE represents the type of function which is perfectly symmetric, i. e., the area $A(x)$ of points as least as good as some fixed point $x$ is always a ball around the optimum, while $\text{SPHERE}^+$ is as asymmetric as possible with the optimum in a corner of $A(x)$.

**Experiment 5.1.** For the experiment, we set the search space dimension to 10 and use swarm sizes $N = 2$ and $N = 10$ for each of the two functions. We initialize the particles' positions in $[-10^{-100}, 10^{-100}]$ in order to guarantee that the attractors are close to the optimum. In combination with




**(a)** Function SPHERE. **(b)** Function SPHERE$^+$.

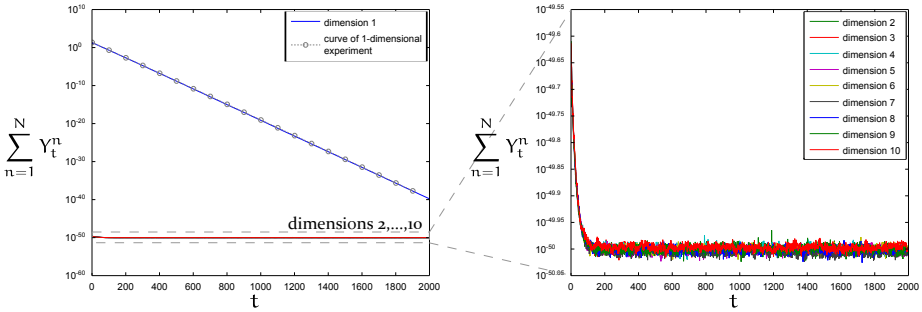
**Figure 5.2:** The symmetric function SPHERE and the asymmetric function SPHERE$^+$.

the different objective functions and swarm sizes, we initialize the velocities uniformly over $[-100, 100] \times [-10^{-100}, 10^{-100}]^{D-1}$ and over $[-100, 100]^{D/2} \times [-10^{-100}, 10^{-100}]^{D/2}$, i. e., we observe the case of one dimension with a too high potential and the case of one half of the dimensions with a too high potential.

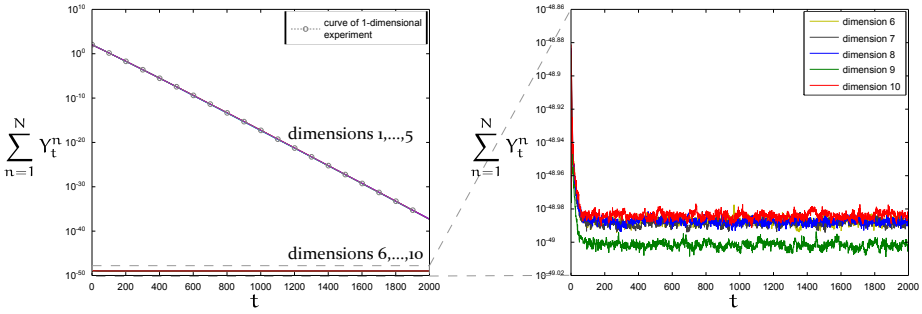
Figure 5.3 shows results about $\sum_{n=1}^{N} Y_t^{n,d}$ for the first 2000 iterations, obtained from optimizing objective function SPHERE. Since the curves regarding SPHERE$^+$ look exactly the same, they are omitted here. Additionally, the circled gray curve shows the values of the potential from the 1-dimensional experiment with a swarm initialized with a too high potential as presented in Figure 4.9, Section 4.2.2, with the respective number of particles. We can see that every dimension with a too high potential (dimension 1 in Figure 5.3a and dimensions $1, \ldots, 5$ in Figure 5.3b) behaves exactly like in the 1-dimensional case and decreases its potential exponentially.

In the other dimensions, in which the potential was not initialized too high, first the potentials decrease as well, but at a certain point, the potentials stagnate and maintain a certain level. The reason for this is that without updates of the attractors, the local and the global attractor do not decrease their difference and this difference leads to a stochastic lower bound on the potential. However, the dimensions, in which the potential do not decrease anymore are the ones that contribute only insignificantly to $\Psi_t^H$. Therefore, the experiments indicate that the secondary measure $\Psi_t^H$ decreases exponentially and the swarm heals itself from this bad event. The speed at which the potential decreases is independent of the objective function and the number

of dimensions and depends only a little on the swarm size. That is because during this bad event, the updates of the local and global attractors are rare and every dimension behaves like an independent copy of the 1-dimensional process in the same situation.




**(a)** Potential of $N = 2$ particles processing Sphere after being initialized with a too high potential in dimension 1.




**(b)** Potential of $N = 10$ particles processing Sphere after being initialized with a too high potential in dimensions $1, \ldots, 5$.

**Figure 5.3:** Particle swarm suffering from too high potentials while processing 10-dimensional objective function Sphere with $N = 2$ or $N = 10$ particles, initialized with 1 or $D/2$ dimensions with too high potential. For comparison, the curve of a 1-dimensional PSO with a too high potential is added.

Indeed, the self-healing capacity of the particle swarm from the bad event $B_t^H$ can be proved, with a proof that is similar to the proof of Lemma 4.2 for 1-dimensional case.

**Lemma 5.1.** There are constants $t'_H \in \mathbb{N}$, $c'_H > 0$, $C_H > 0$, $\delta_H \in (0,1)$ and $\mathrm{const}_H$, depending only on $c_1$, $c_2$, $\chi$ and $N$, such that for every $t_0 \in \mathbb{N}$, every $t_H \geq t_0 + t'_H$ and every $c_H > c'_H$, we have

$$\mathrm{E}\left[\sum_{n=1}^{N} Y_{t_H}^{n,d} \mid \mathcal{F}_{t_0}\right] \leq C_H \cdot \left(\sum_{n=1}^{N} Y_{t_0}^{n,d} + \sum_{n=1}^{N} \sqrt{|A(L_{t_0}^n)|}\right), \tag{5.1}$$

$$\mathrm{E}\left[\sum_{n=1}^{N} Y_{t_H}^{n,d} \mid \mathcal{F}_{t_0}, B_{t_0}^H\right] \leq (1-\delta_H) \cdot \sum_{n=1}^{N} Y_{t_0}^{n,d}, \tag{5.2}$$

$$\tag{5.3}$$

*Proof.* The proof is exactly the same as the proof of (4.2) and (4.3) of Lemma 4.2 in the previous chapter. Here, $\pi_d(A_{L_t^n})$ plays the role of $A_{L_t^n}$ in the 1-dimensional case. Note that the proof of Lemma 4.2 only makes use of the fact that the local attractor is not updated outside of $A_{L_t^n}$, which is also true for its d'th coordinate outside of $\pi_d(A_{L_t^n})$ in the D-dimensional case. Lemma 4.2 did not use the fact that inside of $A_{L_t^n}$, the attractor actually is updated, which was the case in the 1-dimensional situation but is not necessarily true in the D-dimensional case because if $X_t^{n,d} \in \pi_d(A_{L_t^n})$, then it still depends on the other entries of $X_t^n$, whether or not the attractor is updated. □

In particular, the conditions on the swarm parameters for which the particle swarm can heal itself from the bad event $B_t^H$ are the same as in the 1-dimensional case, i. e., the same as specified in Theorem 3.11.

### 5.1.2 Low Potential in every Dimension: A Bad Event

Another bad event occurs when the potential of the swarm is too small in every single dimension, such that only search points very close to the global attractor are visited and therefore every possible improvement is insignificant. In order to recover from such a configuration, the swarm needs to accelerate and to charge potential until its updates become again significant. This event could be formulated as

$$B_t^L := \left\{\forall d \in \{1,\dots,D\} : \Phi_t^d \ll \sqrt{|G_t^{1,d}|}\right\}.$$

Figure 5.4 provides a graphical representation of the bad event $B_t^H$.

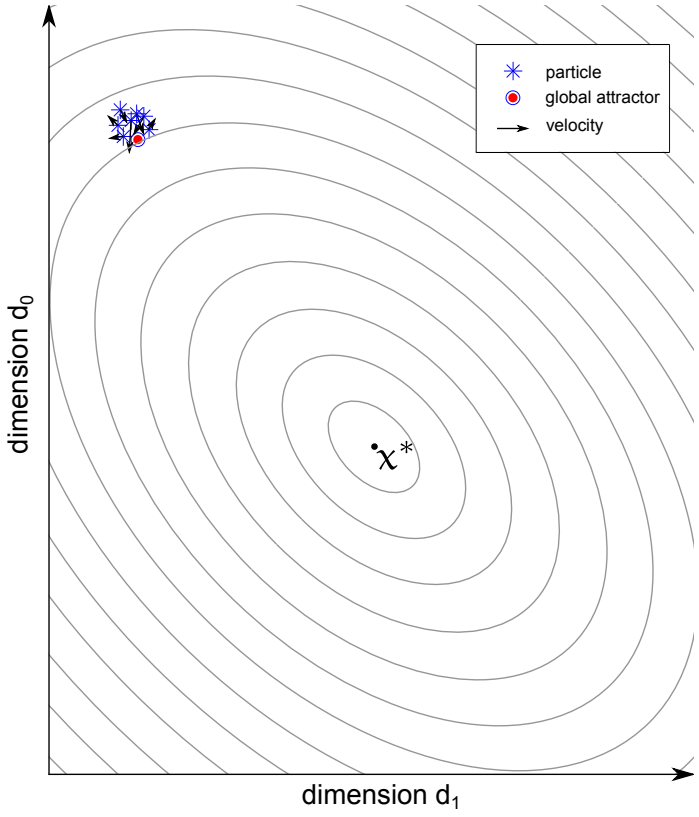



**Figure 5.4:** All dimensions have a potential too small to make significant improvements of the global attractor

A possible secondary measure for this event could be

$$\frac{\sum_{d=1}^{D} |G_t^{1,d}|}{\sum_{d=1}^{D} \Phi_t^d}.$$

To study the behavior of the swarm in such a situation, we use the objective function InclinedPlane, defined as

$$\text{InclinedPlane}(\vec{x}) = -\sum_{d=1}^{D} x_d.$$

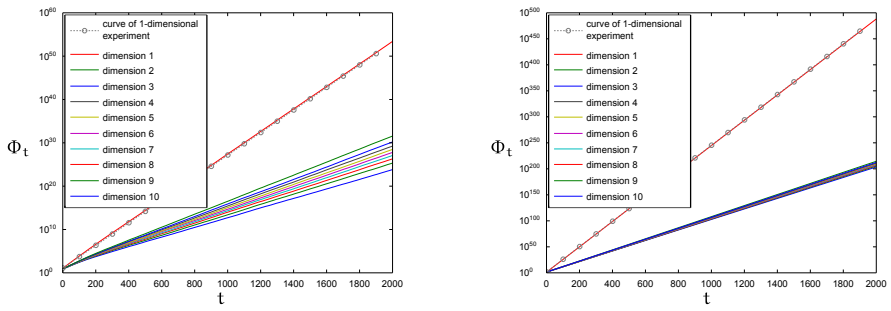
This function is monotonically decreasing in every dimension and therefore suitable to examine the behavior of a particle swarm in a situation where the next local optimum is far out of reach.

**Experiment 5.2.** For this experiment, we set $D$ to 10. Initially, we distribute the particles and the velocities randomly over $[-100; 100]^D$. We set the total number of iterations to 2000. The tests are performed for the swarm sizes $N = 2$ and $N = 10$ and after each iteration $t$, we calculate the potential $\Phi_t$ for each dimension. After every run, we sort the dimensions according to the value of $\Phi_{2000}$, i. e., we switch the indices of the dimensions, such that after the last iteration dimension 1 always has the highest potential, dimension 2 the second highest and so on. The results are stated in Figure 5.5. Additionally, for comparison we added the curve obtained from Experiment 4.3 with a 1-dimensional swarm with the same number of particles processing the objective function $f(x) = -x$.

We can see that the dimension with the largest potential has a potential value far higher than the others, while the other dimensions do not show such a significant difference between each other. That means that the swarm tends to pick one dimension and favor it over all the others. As a consequence, the movement of the swarm becomes more and more parallel to one of the axes. For comparison, the line describing the potential increase in a 1-dimensional situation is added to Figure 5.5. Note that the dimension with the highest potential increases its potential as fast as if it was the only dimension of the swarm.

An explanation for this behavior is the following: Let $d_0$ be the dimension with the largest potential. Further assume that the advance of $d_0$ is large enough, such that for some number of steps the particle with the largest value in dimension $d_0$ is the one that determines the global attractor. Since this only requires $d_0$ to have a potential of a constant factor higher than every other dimension, this will due to the involved randomness eventually happen after sufficiently many iterations. From that moment on, the swarm becomes running in dimension $d_0$.

Figure 5.6 illustrates the mechanism that makes the swarm maintain its own running behavior. Every update of the global attractor increases the potential in $d_0$ considerably, because it increases the distance of every single particle to the global attractor except for the one particle that updated it. In any other dimension $d \neq d_0$, the situation is different. Here, the decision which particle updates the global attractor is stochastically independent of the value $X_t^{n,d}$ in dimension $d$. In other words: If we look only at the dimen-




**(a)** Swarm size $N = 2$ **(b)** Swarm size $N = 10$

**Figure 5.5:** Growth of potential when processing objective function InclinedPlane with swarm size
(a) $N = 2$,
(b) $N = 10$.

sion $d$, the global attractor is chosen uniformly at random from the set of all particles' positions. As a consequence, after some iterations, the $d_0$'th coordinate of the velocity becomes positive for every particle, so the attraction towards the global attractor always goes into the same direction as the velocity, while in the remaining dimensions, the velocities may as well point away from the global attractor, meaning that the particles will be slowed down by the force of attraction.

So, roughly speaking, most of the time the global attractor is somewhere in the middle of the different $X_t^{n,d}$ values for the different particles, giving less potential increase than in dimension $d_0$ where it has a border position. That means that the balanced situation is not stable in a sense that after the imbalance of the potentials in the different dimensions has reached a certain critical value, it will grow unboundedly. From that point on, the decision about attractor updates depends almost only on dimension $d_0$, therefore from the perspective of this dimension, the swarm behaves like in the 1-dimensional case.

These considerations indicate that indeed the swarm heals itself from the bad event $B_t^L$ by choosing one dimension $d_0$ and becoming running in direction $d_0$ until the objective function is no longer decreasing in that direction. In particular, since it is always exactly one dimension which is chosen, the conditions to the swarm parameters, that allow the swarm to actually

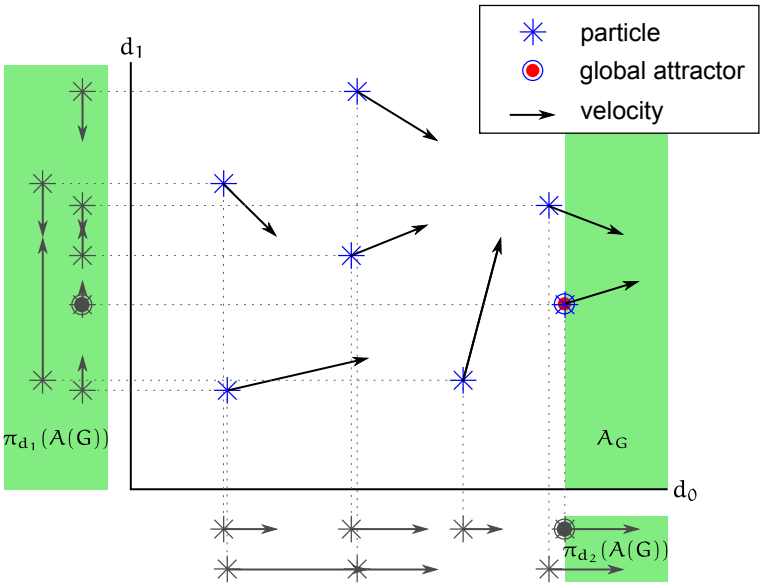



**Figure 5.6:** Particles running in direction $d_0$. In dimension $d_0$, the differences between the coordinate of the particle and the global attractor is on average higher than in dimension $d_1$. The velocities of dimension $d_0$ point in the direction of the global attractor.

increase its potential while running, are again exactly the same as in the 1-dimensional case.

Unfortunately, a new problem arises. When the swarm stops running, the potential in dimension $d_0$ is of the same order as the distance between the swarm and the $d_0$'th coordinate of the optimum, while every other dimension has a potential much smaller than $d_0$. Therefore, the swarm continues improving the global attractor in dimension $d_0$ while ignoring its other components. However, as soon as the room for improvements in dimension $d_0$ is sufficiently much smaller than in some other dimension, possible improvements of the other dimensions are dominated by the "random noise" of dimension $d_0$, i. e., if an attractor is updated or not depends still much more on the respective entries in dimension $d_0$ than on possible improvements of other dimensions. Therefore, the swarm is again not in a situation to make significant progress. This leads to the third bad event, namely the event of imbalanced potentials.

### 5.1.3 Imbalanced Potentials: A Fatal Event

If such a running phase as just described ends, the potentials tend to be very far out of balance, i. e., there is one dimension $d_0$, the one in which the swarm was running, with a potential much larger than the others. Additionally, since the running phase stopped, there is not much improvement left in dimension $d_0$, while the other dimensions still need to be optimized. In general, we can describe the bad event of imbalanced potentials as follows. Let $d_0$ be the dimension with the largest potential and $d_1$ the dimension with the largest distance between the global attractor and the optimum. Then, the bad event of the imbalanced potentials occurs if the potential in dimension $d_0$ is orders of magnitude larger than the potential in dimension $d_1$ while the distance of the global attractor to the optimum is in dimension $d_0$ much smaller than in dimension $d_1$. Additionally, in order to distinct from the event $B_t^H$ of generally too high potential, the potential in dimension $d_0$ is of the same order as the distance to the optimum in the same dimension. In terms (for the case of an optimum at 0):

$$\begin{aligned} d_0 &= \underset{\{d=1,\dots,D\}}{\operatorname{argmax}} \Phi_t^d, d_1 = \underset{\{d=1,\dots,D\}}{\operatorname{argmax}} |G_t^d|, \\ B_t^I &= \{|G_t^{d_1,1}| \gg |G_t^{d_0,1}| \approx Phi_t^{d_0} \gg Phi_t^{d_1}\}. \end{aligned}$$

A visualization of this bad event $B_t^I$ is presented in Figure 5.7.

The first idea is that in such a case, the attractor updates become rare and the swarm behaves similarly to the case of $B_t^H$, i. e., the swarm reduces its potential and converges. However, as we point out in the following, the experiments indicate that at least in the 2-dimensional situation this is not the case.

#### Imbalanced Potentials in 2 Dimensions

In order to examine the behavior of a particle swarm, while it is encountering the bad event of imbalanced potentials, we initialize the particles in an imbalanced way and test experimentally, if the swarm can recover from this imbalanced state.

**Experiment 5.3.** We initialize the positions of the particles uniformly at random over $[-100, 100] \times [-10^{50} - 10^{-100}, -10^{50} + 10^{-100}]$ and the velocities

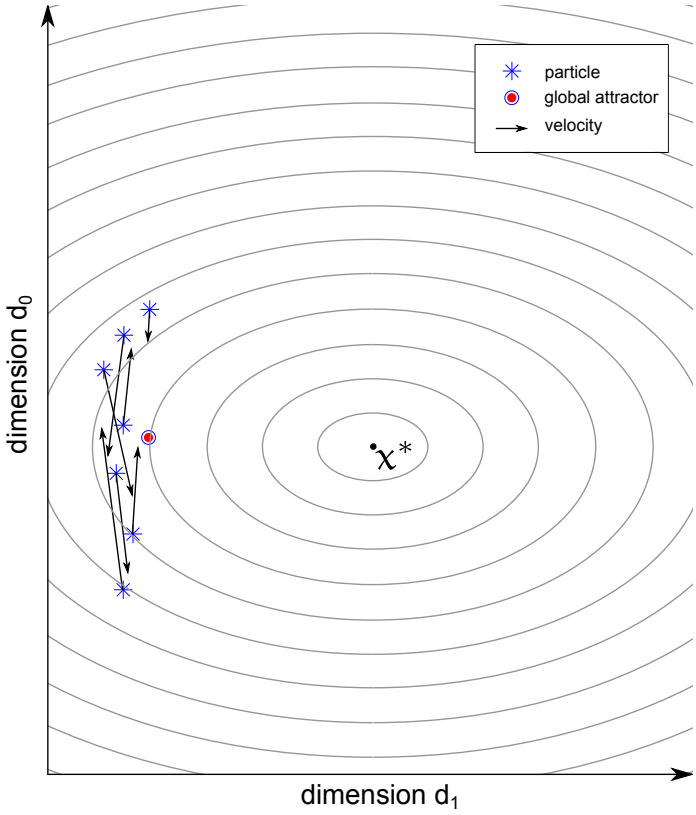



**Figure 5.7:** Imbalanced Potentials: The swarm is in dimension $d_1$ much farther away from the optimum than in dimension $d_0$, while the potential of dimension $d_0$ is much larger than the potential of dimension $d_1$.

over $[-100, 100] \times [-10^{-100}, 10^{-100}]$, such that the second dimension has a much larger distance to the optimum, but a much lower potential than dimension 1. In particular, the potential in dimension 1 is sufficient to reach the optimum, while the potential in dimension 2 is not. We choose two objective functions, namely Sphere (Figure 5.2a) and Schwefel ([SHL+05], Figure 5.8), defined as

$$\text{Schwefel}(\vec{x}) = \sum_{d=1}^{D} \left( \sum_{d'=1}^{d} x_{d'} \right)^2.$$

We test PSO with swarm sizes $N = 2$ and $N = 10$.

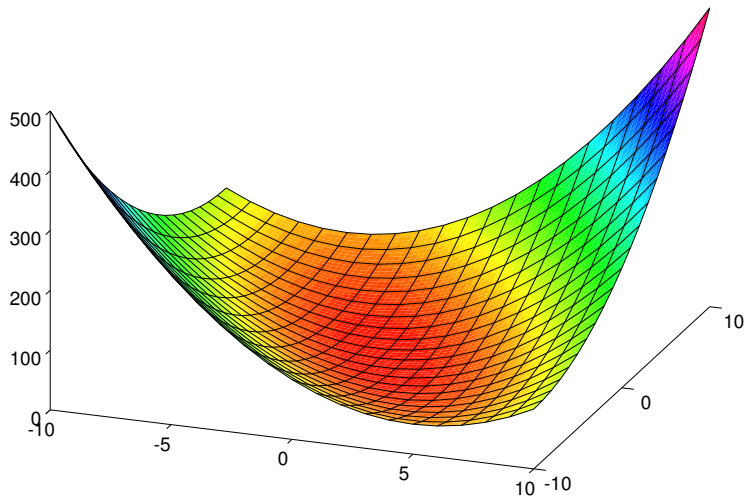
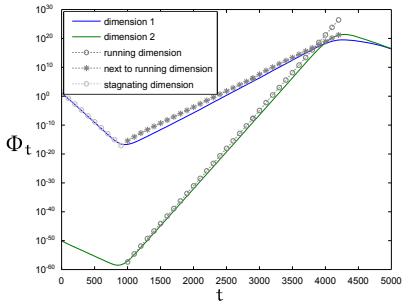


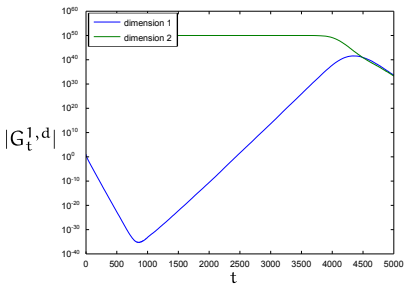

**Figure 5.8:** Objective function SCHWEFEL.

The results regarding objective function SPHERE can be seen in Figure 5.9. Figure 5.9a shows the curve of the potentials in both dimensions for swarm size $N = 2$, while Figure 5.9b shows the courses of the absolute value of the according entry of the global attractors.

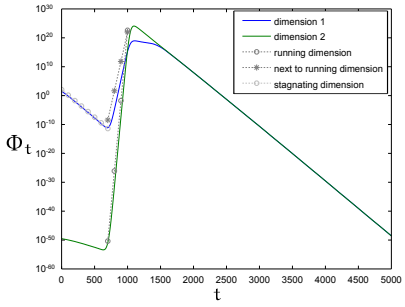
We see that during the first $\approx$ 900 iterations, only the first entry of the global attractor is improved, while the second entry stays about constant. Meanwhile, the potentials of both dimensions decrease. For comparison, a part of the line obtained in Experiment 4.2, where a 1-dimensional swarm was initialized with a too high potential, is added ("stagnating dimension" in Figure 5.9a). Note that here two different measures of potential are compared, but as stated before, they differ by at most a constant factor and are in the current situation both exponentially decreasing, therefore the comparison is still meaningful. We see that dimension 1 decreases with the same speed as the potential in Experiment 4.2, while dimension 2 decreases significantly slower.



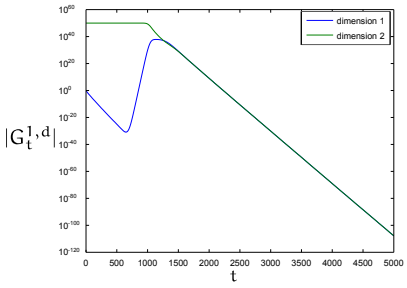

**(a)** Potential curve of $N = 2$ particles initialized with imbalanced potentials.




**(b)** Global attractor getting improved over time.




**(c)** Potential curve of $N = 10$ particles initialized with imbalanced potentials.




**(d)** Global attractor getting improved over time.

**Figure 5.9:** Particle swarm recovering from imbalanced potentials while processing 2-dimensional objective function Sphere with $N = 2$ or $N = 10$ particles, initialized with a too low potential and a very high distance to the optimum in dimension 2.

The reason for this is the following: If after one step of a particle, the new position becomes the new local or even global attractor, depends almost only on the first dimension's entry of the respective positions. Therefore, the first dimension behaves similar to a 1-dimensional PSO and decreases its potential with the same speed as in the 1-dimensional case. However, since the process of optimizing dimension 1 yields a certain update frequency for the attractors, the situation is not the same for dimension 2. In dimension 1, the distance between the old and the new global attractor at each update is limited, i. e., the global attractor can only be updated to points that are close (in relation to $\Phi_t^{d_0}$) to its previous position and therefore tend to be in the

middle of the swarm. On the other hand, since the positions of the particles in dimension 2 have almost no influence on the update probability for the attractors, the difference between the previous and the new position of the global attractor after each update are of order $(\Phi_t^2)^2$. Figure 5.10 illustrates this self-healing effect.

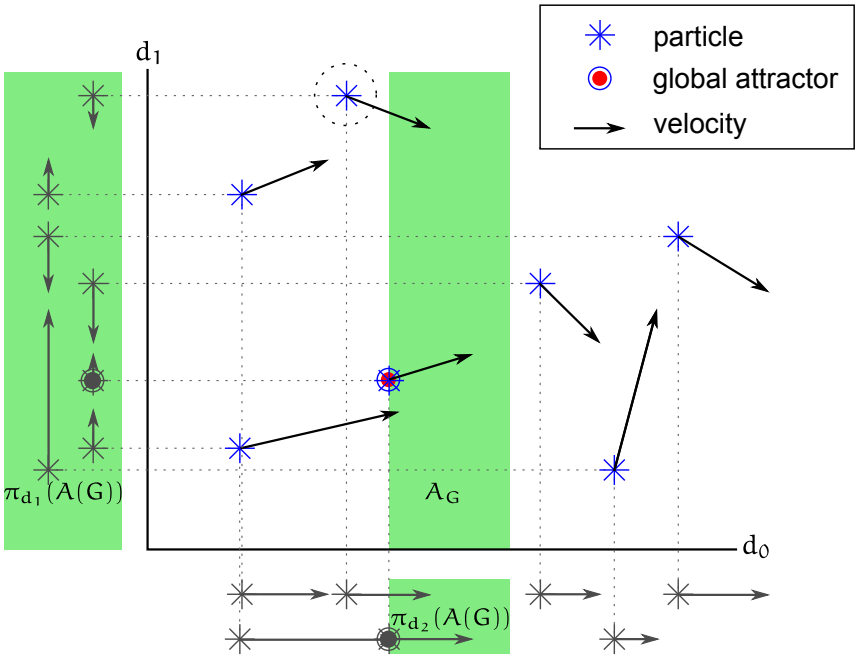



**Figure 5.10:** Particles optimizing dimension $d_0$ while stagnating in dimension $d_1$. Updates of attractors depend mostly on the $d_0$'th entry of the updated position and an attractor update is performed if this entry is comparatively close to the respective entry of the global attractor. Therefore, changes in dimension $d_1$ have only little effect on the updates, but from the perspective of dimension $d_1$ the attractor updates come with a certain frequency and are independent of the positions. If, e. g., the marked particle moves and updates the global attractor, the distance between the old and the updated global attractor has much more impact on dimension $d_1$ than on dimension $d_0$.

In summary, we can see PSO in both dimensions as two processes with different behavior. In dimension 1, a 1-dimensional PSO is run while in di-

mension 2, a PSO-like process is run where the decision if after a step an attractor is updated comes from "outside", namely from dimension 1, which causes attractor updates at a certain frequency. Therefore, if the global attractor is updated, then $|G_t^{n+1,1} - X_{t+1}^{n,1}|$ is much smaller than the (squared) potential of dimension 1, while $|G_t^{n+1,2} - X_{t+1}^{n,2}|$ is of order $(\Psi_t^{n,d})^2$. So, the process describing dimension 2 benefits more from updates than the process of dimension 1. Therefore, the "amount of imbalance" decreases.

Note the similarity of this effect and the situation when the swarm is running. If the swarm is running in a certain dimension $d_0$, then in this dimension the attractor is always at the border of the area populated by the particles, while in every other dimensions it is randomly distributed. Since a global attractor at the border maximizes the sum of the distances between the particles and the global attractor, dimension $d_0$ is able to increase its potential faster. If on the other hand dimension $d_0$ is converging, then the global attractor is in dimension $d_0$ in the middle of the swarm and therefore minimizing the distances between the particles and the global attractors, while in every other dimension it is again randomly distributed. Therefore, the potential of dimension $d_0$ shrinks faster than the potential of the other dimensions.

Although this effect is apparently not strong enough to cause an increase of the potential in dimension 2, at least it slows down the decrease of $\Phi_t^2$ and therefore causes an increase of $\Phi_t^2/\Phi_t^1$. Therefore, after some iterations, the condition $\Phi_t^1 \gg \Phi_t^2$ is violated and the swarm has healed itself from the bad event $B_t^I$.

Finally, at iteration $\approx 900$, the swarm has fully recovered from the imbalanced potentials. Note that this self-healing does not necessarily require the potential in dimension 2 to overcome the potential of dimension 1. Since the gradient of the function is different in different regions, the remaining distance still allows for dimension 2 to have significant impact on the updates of the attractors. In this situation, dimension 1 has a potential too low in comparison with the distance to the optimum. Dimension 2 is generally insignificant because it does not contribute much to the decision about attractor updates. Next, the swarm becomes running in dimension 2, i. e., the potential of dimension 2 increases faster than the potential in dimension 1.

For comparison, shifted (but not scaled) versions of the curves from Experiment 5.2 are added to Figure 5.9a, where the swarm was run on the objective function InclinedPlane in order to simulate the case of a too low potential in every dimension. In Experiment 5.2, one particular dimension, which was afterwards renamed as "dimension 1", was randomly chosen by the process,

in which the potential increased much faster than in the others, while the potentials of the remaining dimensions increased about equally. The curve "running dimension" refers to this special dimension 1 of Experiment 5.2 and the curve "next to running dimension" to dimension 10 of Experiment 5.2. We can see that indeed the increases of the different potentials for both situations behave quite similar.

Note that during this running phase, since the potential in dimension 1 increases and the attractor updates depend almost only on dimension 2, the first coordinate of the global attractor gets worse while its second coordinate improves, such that there is an overall improvement of its objective function value. In Figure 5.9b, we can see $|G_t^{1,1}|$ increasing. For an explanation why the improvement of $|G_t^{1,2}|$ is at the beginning not visible, one has to recall that the axes are logarithmically scaled. The values $|G_t^{1,1}|$ and $|G_t^{1,2}|$ are a *factor* of $\approx 10^{80}$ away from each other, while in the objective function, their squares are summed. Therefore decreasing $|G_t^{1,2}|$ by, e. g., a factor of 0.999, which would not be visible in the figure, improves the global attractor, even if it comes with an $|G_t^{1,1}|$ increased by, e. g., a factor of $10^{40}$, which would indeed be clearly visible.

In Figure 5.9c and Figure 5.9d, we present the respective curves for a swarm with $N = 10$ particles. We see that the behavior is essentially the same as in the case of $N = 2$ particles. During the first phase, the potential of dimension 2 decreases slower than with only $N = 2$ particles, which results in a shorter time necessary for overcoming the imbalance. As seen before, also the increase of potential during the running phase happens faster with more particles, therefore the second phase is again shorter.

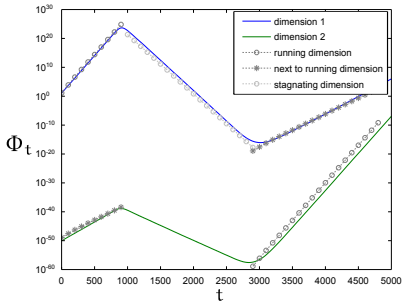
The corresponding results regarding objective function SCHWEFEL are presented in Figure 5.11.

There is a crucial difference between the objective functions SPHERE and SCHWEFEL. The function SPHERE is separable, which in particular means that for every constant $\vec{c} = (c_1, \dots, c_D) \in \mathbb{R}^{D-1}$, we have

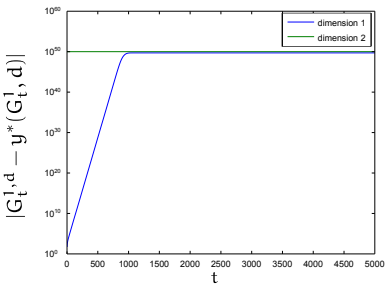
$$\operatorname*{argmin}_{\{x \in \mathbb{R}\}} \text{SPHERE}((x, c_1, \dots, c_D)) = 0.$$

The same is not true for the function SCHWEFEL. Formally, for a function $f(\vec{x}) = \vec{x}^t \cdot A \cdot \vec{x}$ with a positive definite matrix $A \in \mathbb{R}^{D \times D}$, a position $\vec{z} \in \mathbb{R}^D$ and a dimension $d \in \{1, \dots, D\}$, we define

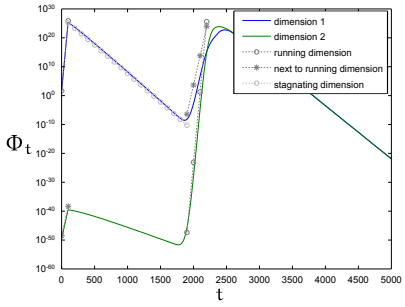
$$y^*(\vec{z}, d) := \operatorname*{argmin}_{\{x \in \mathbb{R}\}} f((z_1, \dots, z_{d-1}, x, z_{d+1}, \dots, z_D)) = 0$$



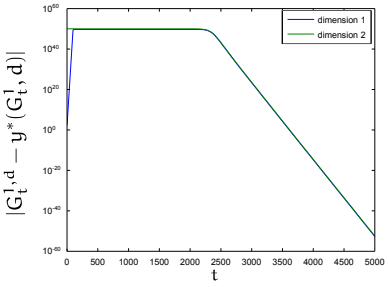

**(a)** Potential curve of $N = 2$ particles initialized with imbalanced potentials.




**(b)** Global attractor getting improved over time.




**(c)** Potential curve of $N = 10$ particles initialized with imbalanced potentials.




**(d)** Global attractor getting improved over time.

**Figure 5.11:** Particle swarm recovering from imbalanced potentials while processing 2-dimensional objective function Schwefel with $N = 2$ or $N = 10$ particles, initialized with a low potential and a very high distance to the optimum in dimension 2.

as the optimum of the 1-dimensional function obtained from fixing all except the $d$'th input of $f$ according to $z$.

When processing objective function Schwefel, the swarm has after initialization in every dimension $d$ a potential too small to reach $y^*(G_0^1, d)$. Therefore, it becomes running - naturally in dimension 1 which has a far larger potential. This can be seen clearly in Figure 5.11a and Figure 5.11c. As soon as dimension 1 is "optimized", i. e., when the swarm approaches $y^*(G_0^1, 1)$, which happens after $\approx$ 1000 iteration for $N = 2$ and $\approx$ 100 iterations for $N = 10$, the bad event of the imbalanced convergence actually

occurs. Similar to the situation when optimizing the function Sphere, the swarm manages to heal itself from this bad event.

We conclude that in the 2-dimensional situation, the method of determining bad events might be the tool to obtain some runtime result, i. e., to verify the linear convergence speed of PSO when optimizing a function of the form $f(x) = x^t \cdot A \cdot x$ for some positive-definite matrix $A$. Although a formal proof for the self-healing capability of the swarm is not found yet, the experiments clearly support the following conjecture.

**Conjecture 5.1** (Linear Convergence Speed for 2-Dimensional Quadratic Objective Functions)**.** Consider a 2-dimensional objective function $f = x^t \cdot A \cdot x$ for some positive-definite matrix $A \in \mathbb{R}^{2\times 2}$. Let $b$ be the diameter of the search space and assume that the particles are initialized such that $E[\sum_{n=1}^{N} Y_t^{n,d}] \leq C_Y \cdot \sqrt{b}$ and $E[1/\Phi_{t+1}^1] \leq C_\Phi$ for two constants $C_Y, C_\Phi > 0$. This is the case, e. g., when the particles' positions are initialized independently and uniformly over $[-b, b]^2$ and if the velocities have finite expectation. Define $\tau := \min\{t \geq 0 \mid \sum_{n=1}^{N} |G_t^{1,n}| + |G_t^{2,n}| \leq 2^{-k} \cdot b\}$. Then there is a constant $c$, depending on the swarm parameters $\chi$, $c_1$, $c_2$ and $N$ and on the objective function $f$, respectively on the matrix $A$, such that the following holds:

$$E[\tau] \leq c \cdot (k+1).$$

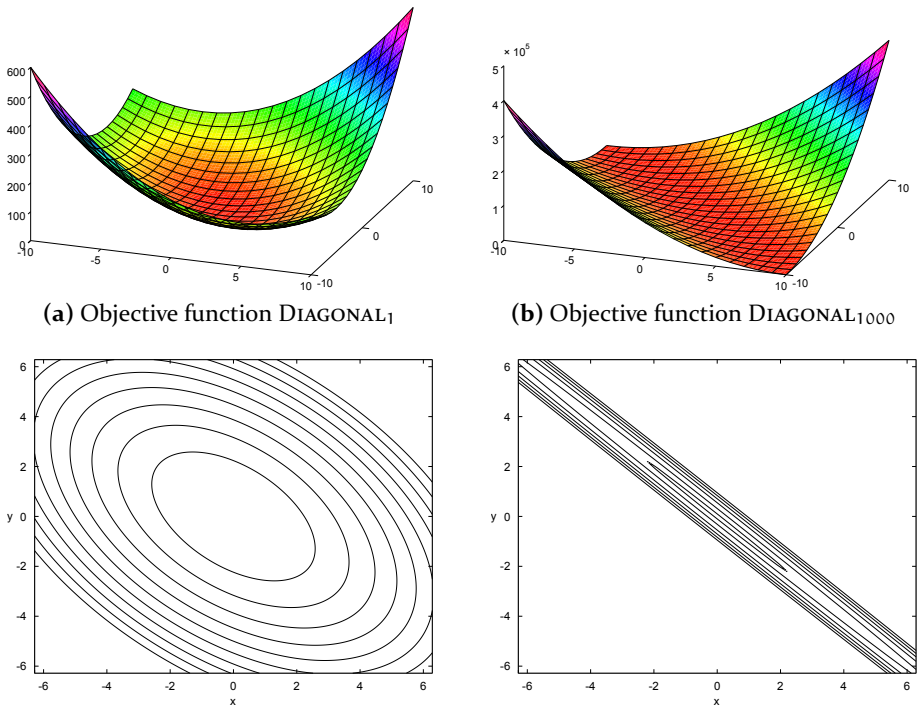
### Imbalanced Potentials in more than 2 Dimensions

While the particle swarm was actually able to heal itself even from the bad event of imbalanced potentials in the 2-dimensional situation, this is generally not possible when the problem dimension gets higher. Instead, the imbalance gets worse and the particle swarm converges towards a non-optimal search point. To see this imbalanced convergence phenomenon, it is not necessary to initialize the swarm with imbalanced potentials.

**Experiment 5.4.** We initialize the particles uniformly at random over the search space $[-100, 100]^D$, where the dimension $D$ varies between 3 and 10. We test PSO with different swarm sizes between 2 and 10 and apply it to various objective functions, namely Sphere, Schwefel, $\text{Diagonal}_1$ and $\text{Diagonal}_{1000}$, where $\text{Diagonal}_r$, obtained from [Raß14], is defined as

$$\text{Diagonal}_r(\vec{x}) = \sum_{d=1}^{D} x_d^2 + r \cdot \left( \sum_{d=1}^{D} x_d \right)^2 .$$

Figure 5.12 shows the 2-dimensional function $\textsc{Diagonal}_r$ for two different choices of $r$. We choose this function, because experiments indicate that when processing it, it is hard for the particles to overcome the bad event of imbalanced potentials. Therefore, this function is a good candidate to show the phenomenon of imbalanced convergence, especially if $r$ is large.




**(a)** Objective function $\textsc{Diagonal}_1$

**(b)** Objective function $\textsc{Diagonal}_{1000}$

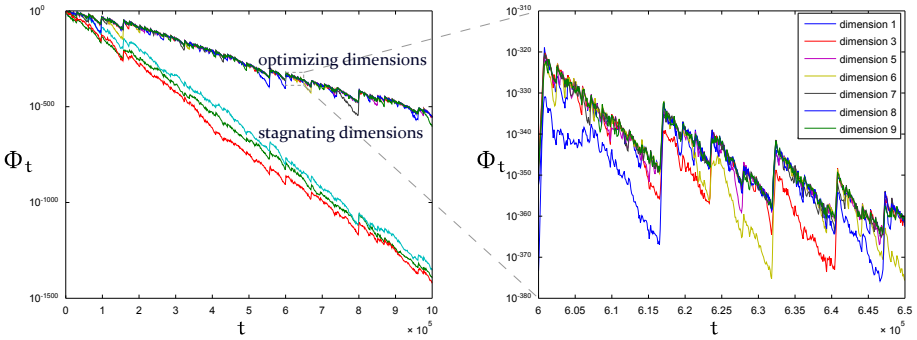
**(c)** Level lines of objective function $\textsc{Diagonal}_1$

**(d)** Level lines of objective function $\textsc{Diagonal}_{1000}$

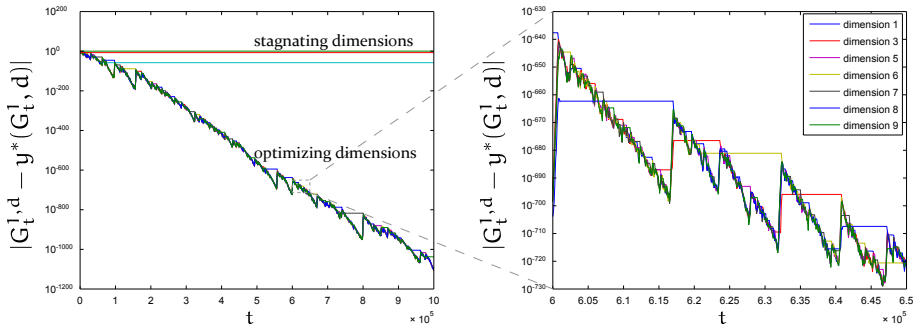
**Figure 5.12:** Objective functions $\textsc{Diagonal}_r$ for $r = 1$ and $r = 1000$.

Figure 5.13 shows the course of the potential $\Phi_t$ and the quality of the global attractor, measured in terms of $y^*(G_t^1, d)$, while processing the 10-dimensional objective function Sphere.

Note that here we just present the result of a single run of PSO instead of the geometric mean of several runs. On the left, we can see a clear separation of the dimensions. While for 7 "optimizing" dimensions, the swarm shows the desired behavior of moderately decreasing potentials and improving the respective entries of the global attractor, there are three other, "stagnating"




**(a)** Potential curve of N = 3 particles processing the 10-dimensional function SPHERE.




**(b)** Curve of the global attractor getting improved over time.

**Figure 5.13:** Particle swarm suffering from imbalanced potentials while processing 10-dimensional objective function SPHERE with N = 3 particles. Dimensions 2, 4 and 10 stagnate and are not optimized.

dimensions (dimension 2, dimension 4 and dimension 10 in Figure 5.13), for which the swarm shows a completely different behavior. For this dimensions, the potential decreases much faster than for the optimizing ones, while the respective coordinates of the global attractor are not significantly improved. The result is a swarm that optimizes only a 7-dimensional problem by ignoring the 3 stagnating dimensions.

At the right part of Figure 5.13, an enlarged part of the process is shown, restricted to the optimizing dimensions. We can see the phases, during which the swarm becomes running into one of the 7 optimizing dimensions, typically the one with the worst entry of the global attractor since this is the

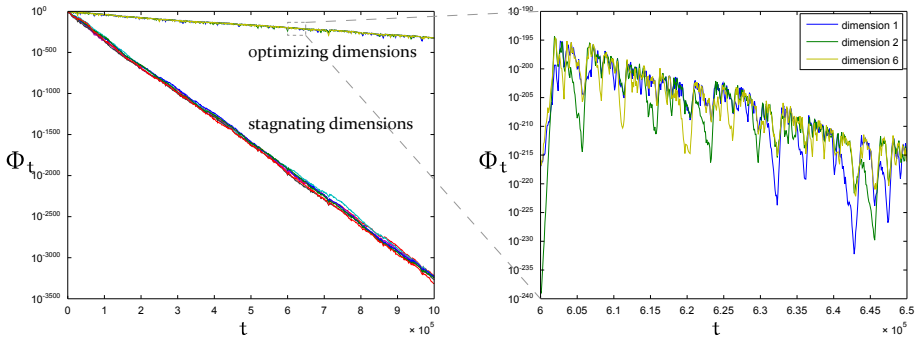
dimension with the largest gradient. The lengths of the running phases differ widely. Sometimes the swarm is running for more than 100 iterations, sometimes it is only running for very few iterations. Between such running phases, there are imbalanced phases during which one or more than one dimension improve their corresponding entries of the global attractor while others have a too low potential to contribute much to the decisions on attractor updates.

An explanation for this separation of the dimensions is as follows. During a running phase, the dimension $d_0$, in which the swarm is running, gains more potential than the others. Assuming that at some point $d_0$ becomes the dimension with the highest potential, the imbalance between $d_0$ and the dimension with the lowest potential increases. After some iterations, the running phase stops. As described before, from that moment on, the more influence the $d$'th components have on decisions about attractor updates, the faster does the potential in this dimension $d$ decrease. Therefore, this phase tends to rebalance the potentials in the different dimensions until one dimension $d_1$, which until then had only little influence on the attractor updates, regains sufficient influence, such that the swarm becomes running in dimension $d_1$. However, $d_1$ is not necessarily the dimension with the lowest potential. The influence of a dimension on attractor updates depends on two factors, namely the potential in this dimension and the absolute value of the objective function's derivative in this dimension. While stagnating dimensions typically have the larger derivative because the corresponding entries are farther away from the optimal point where the derivative is $0$, they have a much smaller potential. Therefore, if the potential of a dimension is too small, then the probability that the swarm becomes running in this dimension is also very small.

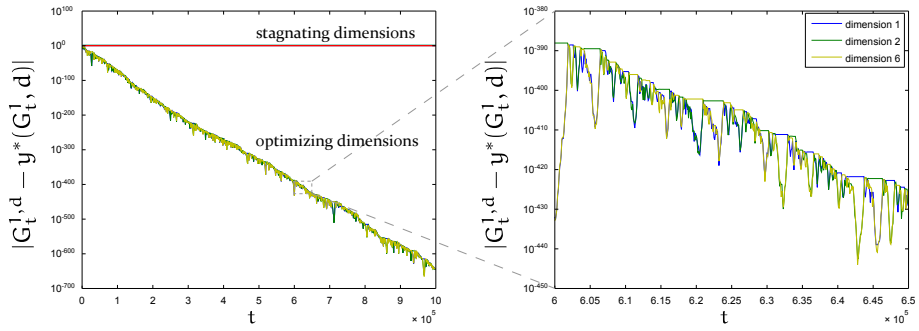
Over the time, the swarm becomes running in the different optimizing dimensions but never in the stagnating dimensions, therefore the running phases increase the imbalance of the potentials between the optimizing and the stagnating dimensions sufficiently fast to compensate both, the decreasing derivative in the optimizing dimensions and the short imbalanced phases. As a result, there is a critical number $D_{opt}$ of dimensions which are optimized, i. e., if $D_{opt}$ dimensions alternate in becoming running, then the imbalance between those $D_{opt}$ dimensions and the remaining dimensions grows fast enough to maintain the separation.

This number $D_{opt}$ depends on the objective function and on the swarm size, because the number of particles determines the shape of the running phases. Experiments indicate that a larger swarm size results in a larger $D_{opt}$.

That is because the effect that allows the swarm to partially balance the potentials of the different dimensions is strengthened by a larger swarm size. See also Figure 5.9, containing the results of Experiment 5.3, where we can see that in the 2-dimensional case, a larger swarm heals itself faster from imbalanced potentials.



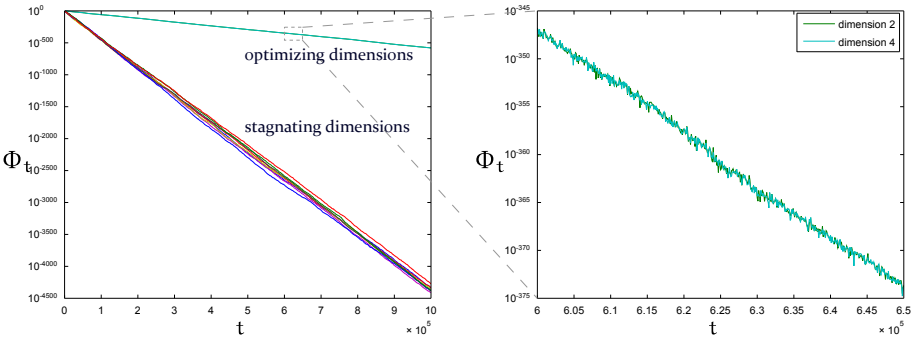

**(a)** Potential curve of $N = 2$ particles processing the 10-dimensional function DIAGONAL$_1$.



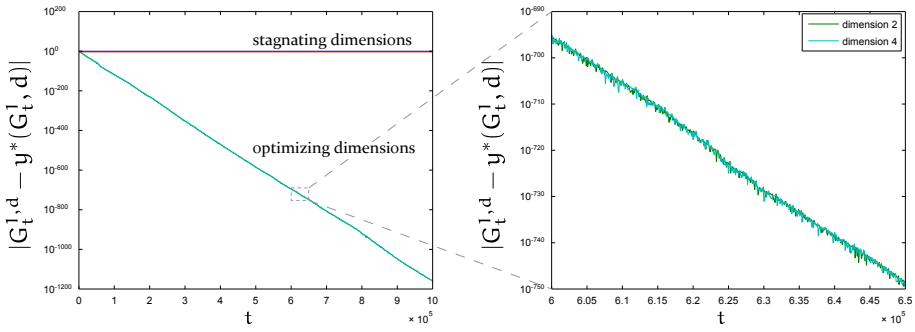

**(b)** Curve of the global attractor getting improved over time.

**Figure 5.14:** Particle swarm suffering from imbalanced potentials while processing 10-dimensional objective function DIAGONAL$_1$ with $N = 2$ particles. Only dimensions 1, 2 and 6 are optimized.

Analyzing the influence of the objective function is not that simple. A function that turns out to be difficult to optimize and yields only a small $D_{opt}$ is the function DIAGONAL$_r$ (see Figure 5.12), particularly for large values of $r$. In Figure 5.14, we can see that in case of objective function DIAGONAL$_1$ and with $N = 2$ particles, the number of optimizing dimensions is $D_{opt} = 3$. If $r$ is increased and the swarm processes the function DIAGONAL$_{1000}$, then there




**(a)** Potential curve of $N = 3$ particles processing the 10-dimensional function $\textsc{Diagonal}_{1000}$.



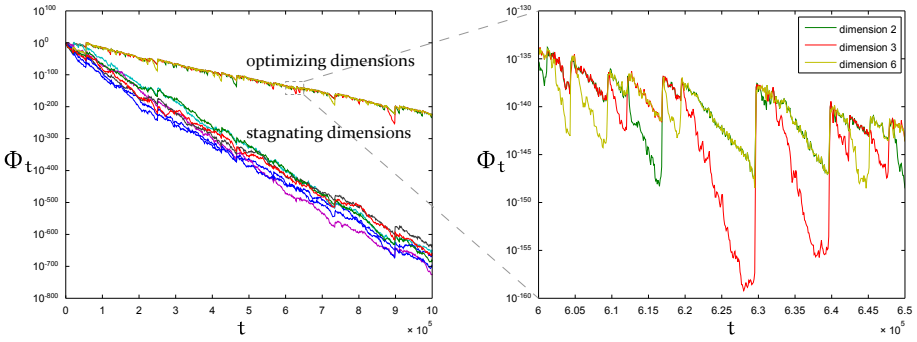

**(b)** Curve of the global attractor getting improved over time.

**Figure 5.15:** Particle swarm suffering from imbalanced potentials while processing 10-dimensional objective function $\textsc{Diagonal}_{1000}$ with $N = 3$ particles. Only dimensions 2 and 4 are optimized.

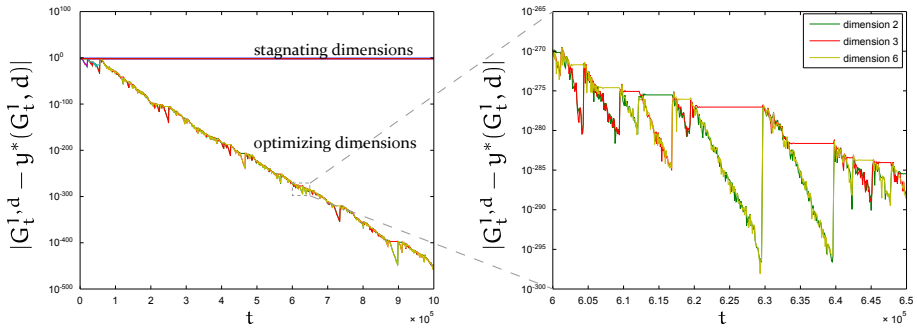
are only $D_{opt} = 2$ optimizing dimensions, which is due to the considerations from the 2-dimensional case the minimum value of $D_{opt}$ for any combination of a swarm size $N \geq 2$ and an objective function $f(\vec{x}) = \vec{x}^t \cdot A \cdot \vec{x}$ with a positive-definite matrix $A$.

Provided with one additional particle, the swarm is able to optimize the function $\textsc{Diagonal}_{1000}$ in 3 dimensions. The results of an example run are presented in Figure 5.16.

If we restrict the $D$-dimensional function $\textsc{Sphere}$ to a $(D-k)$-dimensional function by fixing $k$ components of the input vector, the result is a (possibly shifted) version of the $(D-k)$-dimensional objective function $\textsc{Sphere}$. The same holds for the function $\textsc{Diagonal}_r$. Therefore, the number $D_{opt}$ is for




**(a)** Potential curve of $N = 4$ particles processing the 10-dimensional function DIAGONAL$_{1000}$.

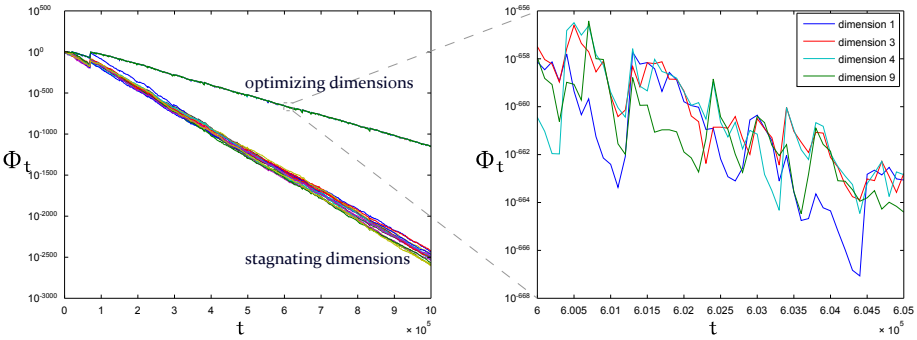



**(b)** Curve of the global attractor getting improved over time.

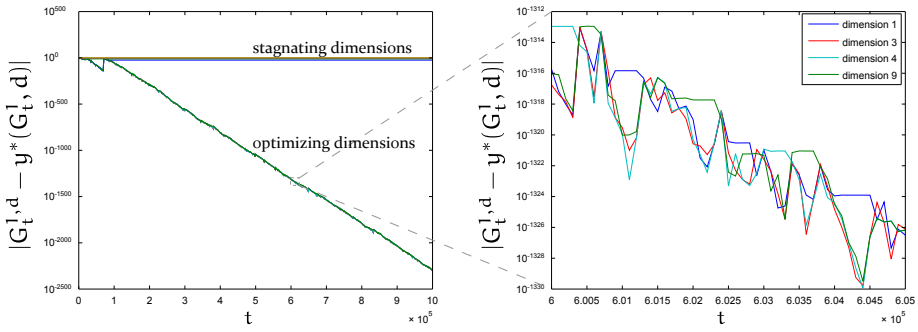
**Figure 5.16:** Particle swarm suffering from imbalanced potentials while processing 10-dimensional objective function DIAGONAL$_{1000}$ with $N = 4$ particles. Only dimensions 2, 3 and 6 are optimized.

both functions independent of the search space dimension D, unless D is too small for any stagnating dimensions to occur. E. g., with $N = 3$ particles, we have $D_{opt} = \min\{D, 7\}$ for the objective function SPHERE with an arbitrary dimension D. The same holds for the function DIAGONAL$_r$.

For objective functions that are less symmetric, the situation is different and no fixed value $D_{opt}$ can be given. As an example, Figure 5.17 and Figure 5.18 each show the results of a run, in which a swarm of $N = 3$ particles processes the 20-dimensional objective function SCHWEFEL. However, as we can see, in the first case, there are $D_{opt} = 4$ optimizing dimensions, while in the second case, it turns out that $D_{opt} = 5$ dimensions are optimizing. The reason for this is that objective function SCHWEFEL is not symmetric




**(a)** Potential curve of $N = 3$ particles processing the 20-dimensional function SCHWEFEL.

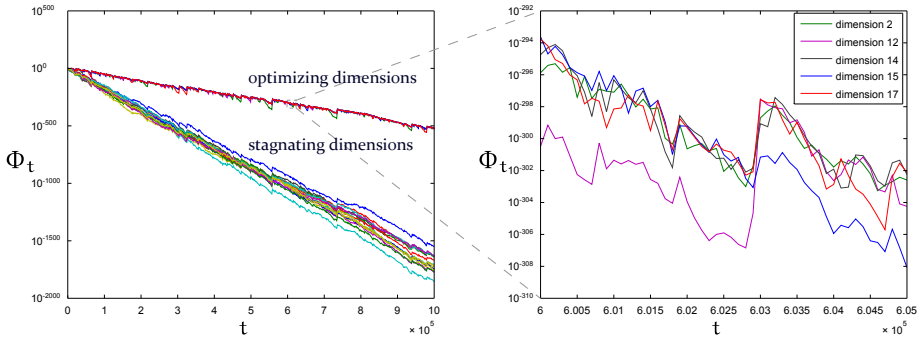



**(b)** Curve of the global attractor getting improved over time.

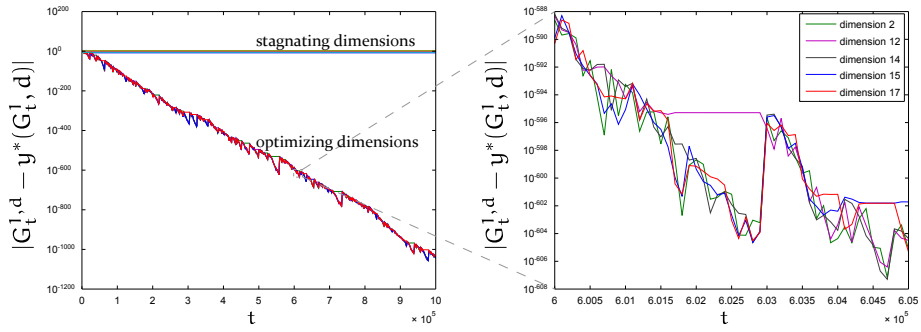
**Figure 5.17:** Particle swarm suffering from imbalanced potentials while processing 20-dimensional objective function SCHWEFEL with $N = 3$ particles. Only dimensions 1, 3, 4 and 9 are optimized.

and in contrast to objective functions SPHERE and $\text{DIAGONAL}_r$ not invariant under permutations of the dimensions. Therefore, some dimensions of the function SCHWEFEL are harder to optimize than others.

It is unclear if there is a swarm size $N_{opt}$, such that a swarm of size at least $N_{opt}$ can optimize any objective function $f(\vec{x}) = \vec{x}^t \cdot A \cdot \vec{x}$ with a positive-definite matrix $A$ and any search space dimension $D$. It is even unclear if for a fixed search space dimension $D$, there is a value $N_{opt}(D)$ such that a swarm of at least $N_{opt}(D)$ particles can optimize any function of the given form. E. g., for $D = 10$ and $N_{opt}(D) = 5$, no stagnating dimensions occurred while processing any of the considered objective functions.




**(a)** Potential curve of $N = 3$ particles processing the 20-dimensional function SCHWEFEL.



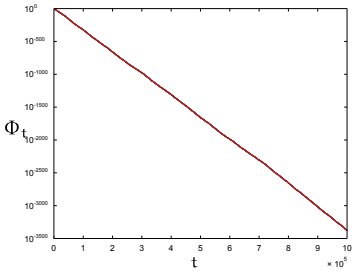

**(b)** Curve of the global attractor getting improved over time.

**Figure 5.18:** Particle swarm suffering from imbalanced potentials while processing 20-dimensional objective function SCHWEFEL with $N = 3$ particles. Only dimensions 2, 12, 14, 15 and 17 are optimized.

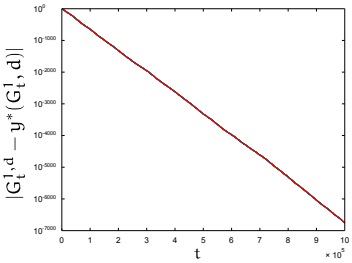
The optimization speed heavily depends on the swarm size and using a swarm size of more than $N_{opt}(D)$ particles might pay off. As an example for the dependency of the optimization speed on the objective function and the swarm size, see Figure 5.19. In Figure 5.19a and 5.19a, we can see the comparatively high optimization speed when optimizing the objective function SPHERE with a swarm size of $N = 3$ particles. In Figure 5.19c and 5.19d, we use the same swarm size to optimize $\text{DIAGONAL}_{1000}$. One can clearly see that the optimization speed is orders of magnitudes slower than when SPHERE is processed, i. e., after the same time in which the swarm processing SPHERE approaches the optimum up to a distance of $< 10^{-7500}$ in every dimension, the swarm optimizing $\text{DIAGONAL}_{1000}$ still has a distance $> 10^{-35}$ to the opti-

mum. Here, increasing the swarm size pays off. Figures 5.19e and 5.19f show a swarm with 10 particles processing again $\textsc{Diagonal}_{1000}$. After 500.000 iterations, the same number of function evaluations are used as in case of a swarm with size $N = 5$ after 1.000.000 iterations, but the obtained value has already a distance of $\approx 10^{-400}$ to the optimum.

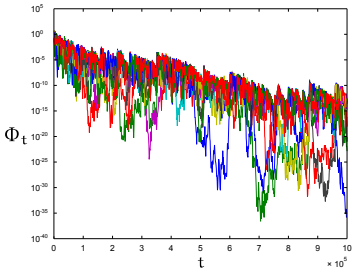
Finding the exact values $N_{opt}(D)$, the minimal swarm size for allowing the swarm to optimize at all, and the optimal swarm size that results in the fastest optimization, remain interesting and promising topics for future research.



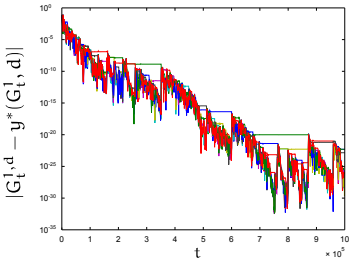

**(a)** Potential curve of $N = 5$ particles processing 10-dimensional function Sphere.



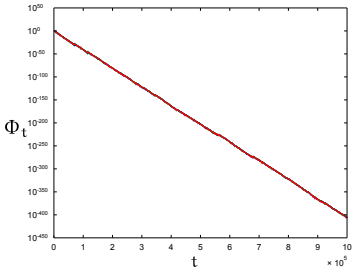

**(b)** Curve of the global attractor getting improved over time.



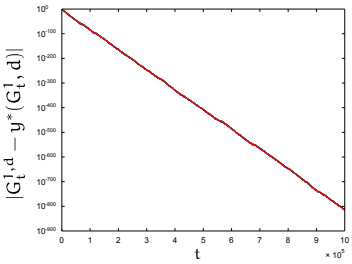

**(c)** Potential curve of $N = 5$ particles processing 10-dimensional function $\text{Diagonal}_{1000}$.




**(d)** Curve of the global attractor getting improved over time.




**(e)** Potential curve of $N = 10$ particles processing 10-dimensional function $\text{Diagonal}_{1000}$.




**(f)** Curve of the global attractor getting improved over time.

**Figure 5.19:** Influence of objective function and swarm size on the speed with which the particle swarm recovers from imbalanced potentials.

## 5.2 Modified Particle Swarm Optimization Almost Surely Finds Local Optima

Since the bad event of imbalanced potentials is a fatal event from which classical PSO cannot recover on its own, we want to modify the algorithm, such that it can overcome this particular situation, but stays as close as possible to the classical PSO. Several modifications are possible, but in order to keep the algorithm as simple and close to the original PSO, we propose the following modification.

**Definition 5.1** (Modified PSO)**.** For some arbitrary small but fixed $\delta > 0$, we define the modified PSO via the same equations as the classic PSO in Definition 3.8, only modifying the third part of the movement equations to

$$V_{t+1}^{n,d} = \begin{cases} (2 \cdot r_t^{n,d} - 1) \cdot \delta, \\ \text{if } \forall n' \in \{1, ..., N\} : |V_t^{n',d}| + |G_{t+1}^{n',d} - X_t^{n',d}| < \delta, \\ \\ \chi \cdot V_t^{n,d} + c_1 \cdot r_t^{n,d} \cdot (L_t^{n,d} - X_t^{n,d}) \\ \qquad + c_2 \cdot s_t^{n,d} \cdot (G_{t+1}^{n,d} - X_t^{n,d}), \\ \text{otherwise.} \end{cases}$$

Whenever the first case applies, we call the step and the whole iteration *forced*.

An algorithmic overview over the modified PSO is given in Algorithm 4. In words: As soon as in one dimension the sum of the velocity and the distance between the position and the global attractor are below the bound of $\delta$ for every single particle, the updated velocity of this dimension is drawn u. a. r. from the interval $[-\delta, \delta]$. Note the similarity between this condition and the definition of the potential. Indeed, we could have used the condition $\Phi_{t+1}^{n,d} < \delta$ (with some fixed $a$) or $\sum_{n=1}^{N} Y_t^{n,d} < \delta$ instead, but the modification as defined in Definition 5.1 is chosen to be as simple, natural and independent from the terms occurring in the analysis as possible. Now the phenomenon of imbalanced convergence can no longer occur because if the potential decreases below a certain bound, a random value, which on expectation has an absolute value of $\delta/2$, is assigned to the velocity. Therefore, the potential of every dimension has a stochastic lower bound.

This modified PSO is similar to the Noisy PSO proposed by Lehre and Witt in [LW11] where the authors generally add a random perturbation drawn

**Algorithm 4:** modified PSO

**input** : Objective function $f : S \to \mathbb{R}$ to be minimized
**output**: $G \in \mathbb{R}^D$

```
// Initialization
```
**for** $n = 1 \to N$ **do**
  Initialize position $X^n \in \mathbb{R}^D$ randomly;
  Initialize velocity $V^n \in \mathbb{R}^D$;
  Initialize local attractor $L^n := X^n$;
Initialize $G := \operatorname{argmin}_{\{L^1,\dots,L^n\}} f$;
```
// Movement
```
**repeat**
  **for** $n = 1 \to N$ **do**
    **for** $d = 1 \to D$ **do**
      **if** $\forall n' \in \{1, \dots, N\} : |V^{n',d}| + |G^d - X^{n',d}| < \delta$ **then**
        $V^{n,d} := (2 \cdot \text{rand}() - 1) \cdot \delta$;
      **else**
        $V^{n,d} := \chi \cdot V^{n,d} + c_1 \cdot \text{rand}() \cdot (L^{n,d} - X^{n,d}) + c_2 \cdot \text{rand}() \cdot (G^d - X^{n,d})$;
      $X^{n,d} := X^{n,d} + V^{n,d}$;
    **if** $f(X^n) \leq f(L^n)$ **then** $L^n := X^n$;
    **if** $f(X^n) \leq f(G)$ **then** $G := X^n$;
**until** *termination criterion met*;

u. a. r. from $[-\delta/2, \delta/2]$ for some small $\delta$ and prove that their swarm is able to find a local optimum. However, their analysis is restricted to one specific 1-dimensional objective function.

Another famous modified version of PSO with a comparable modification is the Guaranteed Convergence PSO (GCPSO) found in [vdBE02] (see Chapter 2, Section 2.3.3). Here, the authors made more complex changes of the movement equations, enabling the particles to count the number of times they improved the global attractor and use that information. Empirical evidence for the capability of the GCPSO to find local optima on common benchmarks is given.

In case of the proposed modified PSO, the change from the classic PSO are considered comparatively simple. The main difference to previous approaches (e. g., [vdBE02]) is that our PSO uses the modification not as its

engine. Rather, it will turn out that the number of forced steps is small and if the swarm is not already within an $\delta$-neighborhood of a local optimum, after some forced steps the potential increases and the swarm switches back to classical steps, a behavior which can also be observed experimentally (see Section 5.3.4)

Note that, however, the convergence of the swarm is sacrificed in order to increase the quality of the solution, since the potential cannot approach 0 anymore. Instead, we can only expect the global attractor to converge. Presumably, we can hope for linear convergence speed until the distance to the optimum is of order $\delta$. From that moment on, the global attractor will still continue converging towards the optimum, but the speed decreases significantly because the swarm is only driven by its modification, which leads to a behavior similar to "blind search" over some interval of size $\delta$. However, since $\delta$ is a user-defined parameter which can be made arbitrarily small, any practical application can take this behavior into account and choose the value $\delta$ according to the desired precision of the result.

Now the question arises how much of the results from the 1-dimensional PSO can be transferred to the general, $D$-dimensional case. Although a rigorous runtime analysis is generally hardly possible because the influence of the objective function and its derivatives is not easy to handle, we can indeed prove that the modified PSO algorithm finds local optima similar to the unmodified PSO in the 1-dimensional case for a comparatively large class of objective functions. For technical reasons, additionally to the requirements of Definition 4.1, in this section we assume that the objective function $f$ has a continuous first derivative. This leads to the following definition of admissible objective functions.

**Definition 5.2.** Let $f : \mathbb{R}^D \to \mathbb{R}$ be a function. $f \in \mathbb{F}$ if and only if

(i) there is a compact set $K \subset \mathbb{R}^D$ with positive Lebesgue measure, such that $P(X_0^n \in K) = 1$ for every $n$ and $\left\{x \in \mathbb{R}^D \;\middle|\; f(x) \leq \sup_K f\right\}$ (the island) is bounded;

(ii) $f \in \mathcal{C}^1(\mathbb{R}^D)$, i. e., $f$ is continuous and has a continuous derivative.

For objective functions satisfying the requirements of Definition 5.2, we prove the following theorem, which is the $D$-dimensional counterpart to Theorem 4.1.

**Theorem 5.1.** Using the modified PSO algorithm, every accumulation point of $G = (G_t^n)_{n=1,\dots,N;t\in\mathbb{N}}$ is a local minimum of $f$ almost surely.

*Proof.* Assume, for contradiction, that there is some accumulation point $z$ of $G$ that is no local minimum. Then, in any neighborhood of $z$ and therefore in particular in $B_\delta(z)$, there is a point $x_0 \in B_\delta(z)$ with $f(x_0) < f(z)$. Since $f$ is continuous, $x_0$ has some neighborhood $B_\tau(x_0)$, such that $f(x) < f(z)$ for every $x \in B_\tau(x_0)$. Figure 5.20 gives an overview over the situation.

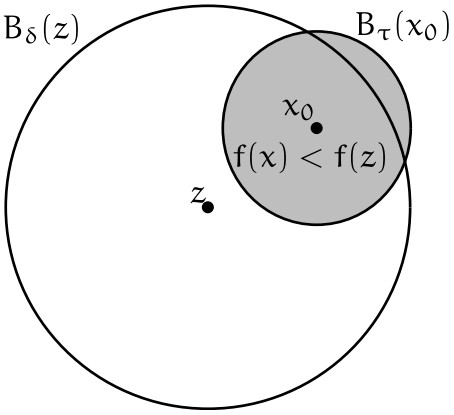



**Figure 5.20:** Every point $x_0$ with $f(x_0) < f(z)$ has a neighborhood $B_\tau(x_0)$, such that $f(x) < f(z)$ for $x \in B_\tau(x_0)$

The set $B_\tau(x_0)$ plays the role of the interval $(z, z + \tau)$ from the proof of Theorem 4.1. Now we investigate what happens when $G$ enters $B_\varepsilon(z)$. This will for each $\varepsilon > 0$ happen infinitely often because $z$ is an accumulation point. The modification of the PSO algorithm enables the construction of a sequence of steps leading a particle into $B_\tau(x_0)$. In principle, the sequence can be obtained by using the sequence from the proof of Theorem 4.1 for every single dimension. However, that may result in a particle being at the desired positions in two distinct dimensions at two different points in time, so the sequence is constructed basically by a simple composition of modified sequences from Theorem 4.1 for every single dimension in which the steps are not forced. The modification of the sequences ensuring that they all have the same length is straight-forward. For forced dimensions, the probability for hitting $B_\tau(x_0)$ within the next step is obviously positive and so is the probability for obtaining a velocity suitable for ensuring that the next step will be forced in case the other dimensions are not at the end of their sequences. Note that due to the modification, splitting cases on whether there is a second accumulation point or not is unnecessary. □

This result is not surprising because in the modified PSO random perturbations occur when the swarm tends to converge and it is easy to see that small random perturbations can optimize any continuous function (but with a very poor runtime). Note that the proof of Theorem 5.1 does neither make use of $f$ having a continuous derivative nor of Lemma 4.1. To supplement this result, we will prove a statement about how often the modification actually applies. It is obvious that for $\delta$ chosen too large, the behavior of the swarm is dominated by its forced steps. The case of $\delta$ being small with respect to the structure of the function is the interesting one. On the other hand, if the distance of a particle and a local optimum is smaller than $\delta$, presumably many of the upcoming steps will be forced because there is no room for further improvements. But we can show that, given the swarm is sufficiently far away from the closest local optimum, the forced steps only balance the potentials between the different dimensions and enable the swarm to become running. In particular, consider the following situation: Let for some dimension $d_0$ and some $c \gg 1$ be $\frac{\partial f}{\partial d_0} < 0$ on a $(c \cdot \delta)$-neighborhood of the current global attractor and let the swarm have low potential, i. e., every particle has in every dimension potential of order $\delta$. Instead of only being driven by the random perturbation, we would like the swarm to become running in direction $d_0$ (or some other direction), increasing the potential in that direction, so the velocity updates can be done according to the classical movement equations again.

**Theorem 5.2.** In the situation described above, the probability for the swarm to become running within a constant number of iterations is positive and independent of $\delta$.

*Proof.* We can explicitly describe a possible sequence of iterations enabling the swarm to become running. First, the particles decrease their distance to the global attractor in every single dimension to at most $\delta \cdot \varepsilon/2$ with $\varepsilon \ll 1$ and a velocity of absolute value less than $\delta \cdot (1 - \varepsilon/2)$, such that the local attractor is updated for all particles except the one whose local attractor is equal to the global attractor. If the current global attractor $G_t^n$ is no local maximum, this can be done because every local attractor has a function value worse than the global attractor and since $f$ is continuous, so the function values of $f$ approach $f(G_t^n)$ when $x$ approaches $G_t^n$. The case of $G_t^n$ being a local maximum has probability 0. Then the next step of each particle is forced. In the next iteration, the velocity of every particle gets smaller than $\delta \cdot \varepsilon/2$ in each dimension except $d_0$. In dimension $d_0$, one particle obtains velocity greater than $\delta \cdot (1+\varepsilon)/2$, such that it gets to a search point that is in dimension

$d_0$ more than $\delta/2$ and in any other dimension at most $\varepsilon \cdot \delta$ away from the previous global attractor. For $\varepsilon$ sufficiently small, this particle will update the global attractor since $f$ has a positive partial derivative in dimension $d_0$. Every other particle obtains in $d_0$ a velocity less than $-\delta \cdot (1+\varepsilon)/2$, making sure that its new position and the new global attractor after that step differ by more than $\delta$. So the next step will not be forced and the potentials have order $\sqrt{\delta}$ in dimension $d_0$ and only $\sqrt{\delta \cdot \varepsilon}$ in every other dimension. So for $\varepsilon$ sufficiently small with respect to the function $f$, the swarm will become running and therefore the steps will actually become unforced.

□

The behavior of the modified PSO is the same as of the classic PSO, except that due to the modification the particles can overcome "corners," i. e., in presence of imbalanced potentials, the modification helps to balance the potentials of the different dimensions. The "blind" algorithm that just randomly checks a point around the previous best solution with range $\delta$ would of course find a local minimum, too, but with a very poor running time because it can not accelerate and therefore its step size will only be of order $\delta$.

## 5.3 Experimental Results with a Standard Implementation

To supplement the results about the behavior of PSO in that "artificial" setting, we run it on two well-known benchmark functions, using standard double precision number and standard methods like calculation of the *arithmetic mean* instead of the geometric mean, to show that the bad event of the imbalanced potentials actually occurs on common benchmark instances and affects the optimization even under common experimental conditions. The following experiments are performed using MATLAB version 8.2.0.701 (R2013b).

### 5.3.1 The Problem of Imbalanced Potentials on Standard Benchmarks

Since the described scenario may happen with positive but, depending on the situation, small probability, we choose the number of particles $N$ small

compared to the number of dimensions D in order to be able to view the phenomenon in a preferably pure condition.

We run PSO on the objective functions SPHERE with optimal solution $z^* = (0, ..., 0)$ and ROSENBROCK with optimal solution $z^* = (1, ..., 1)$ (found in [Ros60]). The function ROSENBROCK is defined as follows:

$$\text{ROSENBROCK}(\vec{x}) = \sum_{d=1}^{D-1} \left( (1 - x_d)^2 + 100 \cdot (x_{d+1} - x_d^2)^2 \right).$$

Figure 5.21 shows the function ROSENBROCK for $D = 2$ dimensions.

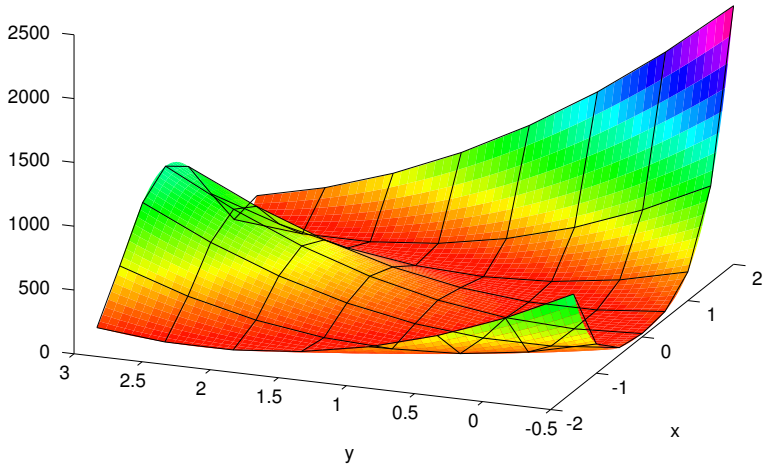



**Figure 5.21:** Objective function ROSENBROCK.

**Experiment 5.5.** We initialize the particles' positions uniformly at random over $[-100, 100]^D$ and the velocities over $[-50, 50]^D$ for processing function SPHERE. For the function ROSENBROCK, we distribute the initial population randomly over $[-5, 10]^D$ and the initial velocity over $[-2.5, 5]^D$. For the search space dimension $D = 5$, we set the swarm size to $N = 2$ and the total number of iterations $t_{max}$ to 10.000 and for $D = 50$, we use $N = 8$ particles and $t_{max} = 100.000$ iterations. We repeat each experiment 1000 times and calculate the arithmetic means.

Table 5.1 lists the results. For each repetition, we determine the dimension with the minimal and the one with the maximal value for the potential $\Phi$ after the last iteration (see columns $\Phi$), together with the difference between the global attractor and the optimal solution in the dimension with the lowest and highest remaining potential, respectively.

**Table 5.1:** Imbalanced Potentials

| Function | | Sphere | | Rosenbrock | |
|---|---|---|---|---|---|
| D | | 5 | 50 | 5 | 50 |
| N | | 2 | 8 | 2 | 8 |
| $t_{max}$ | | 10000 | 100000 | 10000 | 100000 |
| Value | | 247.83 | 26.2706 | $4.19 \cdot 10^{6}$ | $4.1256 \cdot 10^{5}$ |
| min. $\Phi$ | $\Phi$ | $0^{*}$ | $0^{*}$ | $0^{*}$ | $1.0320 \cdot 10^{-5}$ |
| | dist. opt. | - | 0.9188 | - | 0.3235 |
| max. $\Phi$ | $\Phi$ | $0^{*}$ | $2.2778 \cdot 10^{-4}$ | $0^{*}$ | 0.1789 |
| | dist. opt. | - | $2.5449 \cdot 10^{-8}$ | - | 104.2775 |

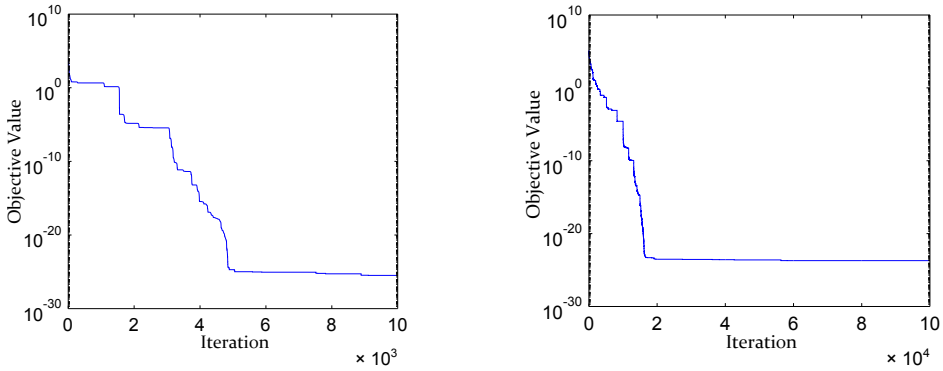
$^{*}$ Due to double precision.

In the 5-dimensional case, the potential reaches 0 due to double precision in every dimension, so there is no single dimension with highest or lowest potential. However, the function value obtained at the point where the particles converge to is still far away from the optimum. In case of the 50-dimensional SPHERE, we can see that the dimension with the highest value for $\Phi$ usually is much closer to its optimal value than the dimension with the lower value. This confirms the concerns about the classical PSO in connection with imbalanced potentials. Since function ROSENBROCK is non-separable, the same relationship between the remaining potentials and the distance to the optimum cannot be obtained. However, what we can see is that the remaining potentials are much smaller than the distance to the optimum.

### 5.3.2 Avoiding Imbalanced Convergence

We repeat Experiment 5.5 in the same setting as before, but using the modified PSO as defined in Definition 5.1 with $\delta := 10^{-12}$. The results can be seen in Table 5.2. It turns out that the modified PSO algorithm actually leads

to a significantly better solution than the unmodified one. In particular, the values obtained by the modified algorithm processing objective function SPHERE are of order $\delta^2$, which means that the swarm was already closer than $\delta$ to the optimum. Figure 5.22 shows the function value of the global attractor at each time during two particular runs. We can see that as long as this value is larger than $\delta^2$, the swarm sometimes stagnates and does not improve until it finds another promising direction and accelerates again. The results from processing function ROSENBROCK look different. Here, even the values obtained by modified algorithm are still far away from the optimum. So, some of the runs are not converged up to an error of $\delta$ when they are stopped.




**(a)** D = 5, N = 2 **(b)** D = 50, N = 8

**Figure 5.22:** Curve of the objective function value of the global attractor when processing function SPHERE with the modified PSO.

### 5.3.3 Differentiability

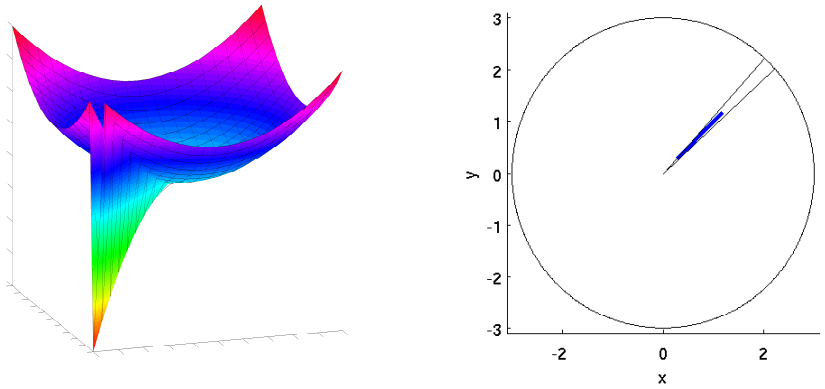
As stated earlier, the only substantial restriction for the objective function f is that f must have a continuous first derivative. In the following, we provide

**Table 5.2:** Comparison between the classic and the modified PSO algorithm

| Function | D | N | $t_{max}$ | $\delta$ | Value |
|---|---|---|---|---|---|
| Sphere | 5 | 2 | 10000 | $10^{-12}$ | $1.91 \cdot 10^{-26}$ |
| Sphere | 5 | 2 | 10000 | - | 247.83 |
| Sphere | 50 | 8 | 100000 | $10^{-12}$ | $2.1402 \cdot 10^{-24}$ |
| Sphere | 50 | 8 | 100000 | - | 26.27 |
| Rosenbrock | 5 | 2 | 10000 | $10^{-12}$ | $2.67 \cdot 10^{5}$ |
| Rosenbrock | 5 | 2 | 10000 | - | $4.19 \cdot 10^{6}$ |
| Rosenbrock | 50 | 8 | 100000 | $10^{-12}$ | 220.66 |
| Rosenbrock | 50 | 8 | 100000 | - | $4.13 \cdot 10^{5}$ |

an example, showing what can happen when $f$ is only continuous. For some fixed $b > 1$, we define the $D$-dimensional function $f$ as follows:

$$f(\vec{x}) = \begin{cases} \sum_{i=1}^{n} x_i^2, & \exists i,j : x_i \geq b \cdot x_j \vee x_j \geq b \cdot x_i \\ \frac{\sum_{i=1}^{n} x_i^2}{b-1} \cdot \left( 2 \max_{i \neq j} \left\{ \frac{x_i}{x_j} \right\} - b - 1 \right), & \text{otherwise} \end{cases}$$




**(a)** Continuous, not differentiable function $f$

**(b)** Particles making almost only forced steps through the valley

**Figure 5.23:** (a) Function $f$, (b) behavior of the particles on $f$

Figure 5.23a shows a plot of the 2-dimensional function $f$. For $y$ not between $x/b$ and $x \cdot b$, this function behaves like the well-known function

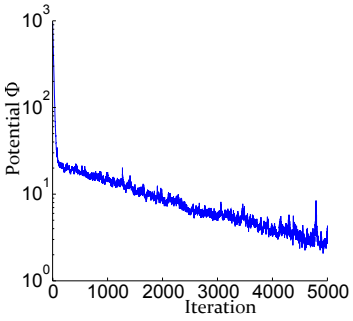
SPHERE. For $x = y$, $f(x, y) = -2 \cdot x^2$ and from $y = x/b$ ($y = x \cdot b$) to $y = x$, the function falls into a valley. It is easy to see that this function is continuous but has no derivative. The construction of a continuous function which behaves like f on a bounded set and tends to infinity for $|x| + |y| \to \infty$ is straightforward. Therefore, the particles must be able to pass through the valley.

**Experiment 5.6.** We initialize the particles' positions uniformly at random over $[-100; 100]^D$ (except for the first particle, which is initialized at $(1, ..., 1)$ such that the swarm could see the direction where the improvements are possible) and the velocities over $[-50; 50]^D$, with the value $D = 3$. We perform a total of 1000 runs, each with 5000 iterations. We determine the potential of the dimension with the highest potential after the last iteration and calculate the mean and standard deviation of the respective dimensions over the 1000 repetitions. This is done for two different swarm sizes, namely $N = 10$ and $N = 50$.

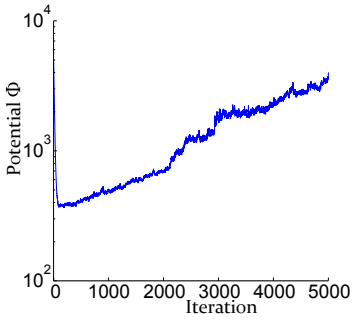
We repeat the experiment with 10 particles and only 100 iterations, using the function $f_{rot}$, which is obtained by first rotating the input vector and then applying f such that the valley now leads the particles along the $x_1$-axis. Formally speaking, the rotation maps the vector $(\sqrt{N}, 0, \ldots, 0)$ to $(1, 1, \ldots, 1)$ and keeps every vector that is orthogonal to this two invariant.

The results of Experiments 5.6 can be seen in Figure 5.24. In all three cases, for about the first 20 iterations, the swarm behaves like on the function SPHERE and reduces its potential. Then, it discovers the valley and tries to move through it. However, in the unrotated case with 10 particles (Figure 5.24a), the swarm fails to accelerate and instead, it converges towards a non-optimal point. With much more effort, the swarm consisting of 50 particles (Figure 5.24b) is able to accelerate, but the acceleration rate and therefore the speed are comparatively poor. Finally, Figure 5.24c shows how the swarm handles the rotated version much better than the original function f before. Here, after only 100 iterations, the potential increased to a value of about $10^{45}$. The reason for this large difference between the behavior on f and on $f_{rot}$ is the capability of the swarm to favor one direction only if this direction is parallel to one of the axes.

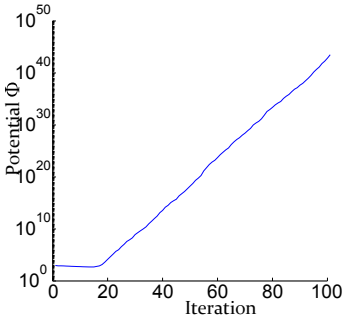
In particular, this experiment also confirms the results in [HRM+11], namely that PSO is not invariant under rotations of the search space.




**(a)** Function f, $b = 1.1$, 10 particles




**(b)** Function f, $b = 1.1$, 50 particles




**(c)** Function $f_{rot}$, $b = 1.1$. 10 particles

**Figure 5.24:** Behavior of the particles on functions f and $f_{rot}$

### 5.3.4 Impact of the Modification

To make sure that the modification does not fully take over, we track the forced points, i. e., the positions of particles before performing a forced step, in order to see how many of them occur and where the modification is used.

**Experiment 5.7.** For processing the 2-dimensional function Sphere, we initialize the particles uniformly at random over $[-100, 100]$ and the velocities over $[-50, 50]$. We set the parameter $\delta$ to $10^{-7}$. The swarm size is set to $N = 2$ and the number of iterations to $t_{max} = 100.000$.

The points at which particle performed a forced step can be seen in Figure 5.25.

As can be seen in the figure, the particles get forced near $(-2 \cdot 10^{-5}, 0)$ but their movement does not stay forced. Instead, the swarm becomes run-

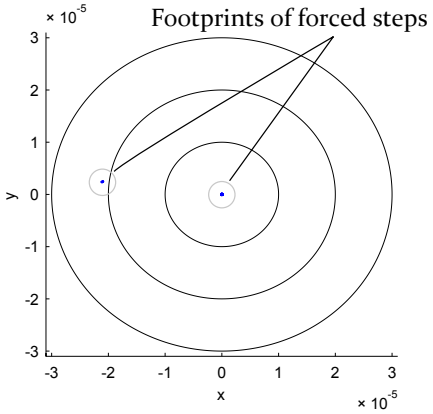



**Figure 5.25:** Behavior of the modified PSO on function SPHERE

ning again until the particles approached the optimum at $(0, 0)$. This implies that for sufficiently smooth functions, the modification does not take over, replacing PSO by some random search routine. Instead, the modification just helps to overcome "corners." As soon as there is a direction parallel to an axis with decreasing function value, the swarm becomes running again and the unmodified movement equations apply.

# 6. Conclusion

In this thesis, we studied the process of convergence in detail. As a main tool for the analysis, we defined the potential of a particle swarm and analyzed its course in order to measure, how far the swarm at a certain time is already converged. With the help of this potential, we could prove the first main result, namely that in a 1-dimensional situation, the swarm with probability 1 converges towards a local optimum for a comparatively wide range of admissible objective functions.

In order to measure the runtime, so-called bad events, i. e., situations in which the particle swarm optimization (PSO) does not make significant progress, where studied. We could proof that in the 1-dimensional situation, the swarm is able to recover from encountering such a bad event within reasonable time. Applying drift theory led to the second main result, namely the formal proof that the swarm obtains a precision of $k$ digits in time $\mathcal{O}(k)$.

In the general $D$-dimensional case, it turned out that there exists a bad event from which the swarm is unable to recover, namely the situation when some dimensions have a potential orders of magnitude smaller than others. Such dimensions with a too small potential loose their influence on the behavior of the algorithm, and therefore the respective entries are not optimized. In order to solve this issue, a slightly modified PSO was proposed that again guarantees convergence towards a local optimum. Experiments where presented, indicating that indeed the modified swarm recovers from this bad event, and also indicating that the modification does not govern the whole algorithm.

### Directions of future research

A future research goal is to formally prove the mentioned aspects of the swarm's behavior in situations with more than one dimension. A first attempt on this is made in [Raß14], where the author provides a theoretical

framework for formally proving that the swarm sometimes stagnates at locations that are no local optimum.

The technique of studying the bad events can be expanded to more general cases, e. g., a possible next step is to provide convergence proof and runtime bounds for the 2-dimensional case using the unmodified PSO. Another opportunity lies in further studying the modified PSO from Section 5.2. As long as $k < \log(1/\delta)$, one could expect the modified PSO to obtain a precision of $k$ digits in time $\mathcal{O}(k)$. Since $\delta$ is a user-defined parameter, this can be sufficient for practical applications.

Finally, the proposed continuous drift theorem looks promising and it is likely that it can be applied in order to analyze other continuous optimization heuristics. Therefore, this direction of drift theory on its own is worth further studies in order to, e. g., generalize it to other situations on the one hand and on the other hand specialize it in order to find tighter bounds when the investigated stochastic process has stronger properties, e. g., the Markov property.

# Bibliography


[ABEF05] Julio E. Alvarez-Benitez, Richard M. Everson, and Jonathan E. Fieldsend. A MOPSO algorithm based exclusively on pareto dominance concepts. In *Proceedings of the Conference on Evolutionary Multi-Criterion Optimization (EMO)*, volume 3410 of *Lecture Notes in Computer Science*, pages 459–473. Springer, 2005. doi:10.1007/978-3-540-31880-4_32.

[AP09] Davide Anghinolfi and Massimo Paolucci. A new discrete particle swarm optimization approach for the single-machine total weighted tardiness scheduling problem with sequence-dependent setup times. *European Journal of Operational Research*, 193(1):73–85, 2009.

[AT10] Anne Auger and Olivier Teytaud. Continuous lunches are free plus the design of optimal optimization algorithms. *Algorithmica*, 57(1):121–146, 2010. doi:10.1007/s00453-008-9244-5.

[Bau96] Heinz Bauer. *Probability Theory*, volume 23 of *Studies in Mathematics*. De Gruyter, 1996.

[BBD+09] Surender Baswana, Somenath Biswas, Benjamin Doerr, Tobias Friedrich, Piyush P. Kurur, and Frank Neumann. Computing single source shortest paths using single-objective fitness. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, Proceedings of the ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA), pages 59–66. ACM, 2009. doi:10.1145/1527125.1527134.

[BF05] Bogdan Bochenek and Paweł Foryś. Structural optimization against instability using particle swarms. In *Proceedings of the 6th World Congress on Structural and Multidisciplinary Optimization*, 2005. CD-ROM Proceedings.

[BK07] Daniel Bratton and James Kennedy. Defining a standard for particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 120–127, 2007. doi:10.1109/SIS.2007.368035.

[BM06] Jürgen Branke and Sanaz Mostaghim. About selecting the personal best in multi-objective particle swarm optimization. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund K. Burke, Juan J. Merelo Guervós, L. Darrell Whitley, and Xin Yao, editors, *Proceedings of the Conference on Parallel Problem Solving from Nature (PPSN)*, volume 4193 of *Lecture Notes in Computer Science*, pages 523–532. Springer, 2006. doi:10.1007/11844297_53.

[BM14] Mohammad Reza Bonyadi and Zbigniew Michalewicz. Spso 2011: Analysis of stability; local convergence; and rotation sensitivity. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 9–16. ACM, 2014. doi:10.1145/2576768.2598263.

[BML14] Mohammad Reza Bonyadi, Zbigniew Michalewicz, and Xiaodong Li. An analysis of the velocity updating rule of the particle swarm optimization algorithm. *Journal of Heuristics*, pages 1–36, 2014. doi:10.1007/s10732-014-9245-2.

[BS02] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies - a comprehensive introduction. *Natural Computing*, 1(1):3–52, 2002. doi:10.1023/A:1015059928466.

[BSMD08] Sanghamitra Bandyopadhyay, Sriparna Saha, Ujjwal Maulik, and Kalyanmoy Deb. A simulated annealing-based multiobjective optimization algorithm: Amosa. *IEEE Transactions on Evolutionary Computation*, 12(3):269–283, 2008. doi:10.1109/TEVC.2007.900837.

[CCZ09] Zhihua Cui, Xingjuan Cai, and Jianchao Zeng. Stochastic velocity threshold inspired by evolutionary programming. In *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 626–631. IEEE, 2009. doi:10.1109/NABIC.2009.5393434.

[CD01] Anthony Carlisle and Gerry Dozier. An off-the-shelf PSO. In *Proceedings of the Particle Swarm Optimization Workshop*, pages 1–6, 2001.

[CK02] Maurice Clerc and James Kennedy. The particle swarm – explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6:58–73, 2002. doi:10.1109/4235.985692.

[Cle03] Maurice Clerc. Tribes, a parameter free particle swarm optimizer. http://clerc.maurice.free.fr/pso/, 2003.

[Cle04] Maurice Clerc. *Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem*, pages 219–239. Springer Berlin Heidelberg, 2004.

[Cle06a] Maurice Clerc. Confinements and biases in particle swarm optimisation. http://clerc.maurice.free.fr/pso/, 2006.

[Cle06b] Maurice Clerc. Stagnation analysis in particle swarm optimization or what happens when nothing happens. http://hal.archives-ouvertes.fr/hal-00122031, 2006. Technical report.

[Cle07] Maurice Clerc. Back to random topology. http://clerc.maurice.free.fr/pso/, 2007.

[Coe02] Carlos A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, 191:1245–1287, 2002. doi:10.1016/s0045-7825(01)00323-1.

[CZS06] Zhihua Cui, Jianchao Zeng, and Guoji Sun. Adaptive velocity threshold particle swarm optimization. In Guoyin Wang, James F. Peters, Andrzej Skowron, and Yiyu Yao, editors, *Proceedings of the International Conference on Rough Sets and Knowledge Technology (RSKT)*, volume 4062 of *Lecture Notes in Computer Science*, pages 327–332. Springer, 2006. doi:10.1007/11795131_47.

[DG97] Marco Dorigo and Luca Maria Gambardella. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997. doi:10.1109/4235.585892.

[DG10] Benjamin Doerr and Leslie Ann Goldberg. Drift analysis with tail bounds. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature, PPSN XI*, volume 6238 of *Lecture Notes in Computer Science*, pages 174–183. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-15844-5_18.

[DJ10] Benjamin Doerr and Daniel Johannsen. Edge-based representation beats vertex-based representation in shortest path problems. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, pages 759–766. ACM, 2010. doi:10.1145/1830483.1830618.

[DJW02] Stefan Droste, Thomas Jansen, and Ingo Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1–2):51–81, 2002. doi:10.1016/S0304-3975(01)00182-7.

[DJW12] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64(4):673–697, 2012. doi:10.1007/s00453-012-9622-x.

[dSC08] Leandro dos Santos Coelho. A quantum particle swarm optimizer with chaotic mutation operator. *Chaos, Solitons & Fractals*, 37(5):1409–1418, 2008.

[EK95] Russell C. Eberhart and James Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the 6th International Symposium on Micro Machine and Human Science*, pages 39–43, 1995. doi:10.1109/MHS.1995.494215 .

[Fan02] Huiyuan Fan. A modification to particle swarm optimization algorithm. *Engineering Computations*, 19(7-8):970–989, 2002. doi:10.1108/02644400210450378.

[FG02] P. C. Fourie and Albert A. Groenwold. The particle swarm optimization algorithm in size and shape optimization. *Struc-*

*tural and Multidisciplinary Optimization*, 23:259–267, 2002. doi:10.1007/s00158-002-0188-0.

[GADP89] S. Goss, Serge Aron, Jean L. Deneubourg, and Jacques Pasteels. Self-organized shortcuts in the argentine ant. *Naturwissenschaften*, 76(12):579–581, 1989. doi:10.1007/bf00462870.

[GAHG05] Crina Grosan, Ajith Abraham, Sangyong Han, and Alexander F. Gelbukh. Hybrid particle swarm - evolutionary algorithm for search and optimization. In *Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI)*, volume 3789 of *Lecture Notes in Computer Science*, pages 623–632. Springer, 2005. doi:10.1007/11579427_63.

[GW03] Oliver Giel and Ingo Wegener. Evolutionary algorithms and the maximum matching problem. In Helmut Alt and Michel Habib, editors, *STACS 2003*, volume 2607 of *Lecture Notes in Computer Science*, pages 415–426. Springer Berlin Heidelberg, 2003. doi:10.1007/3-540-36494-3_37.

[GWHK09] Alexandr Gnezdilov, Stefan Wittmann, Sabine Helwig, and Gabriella Kókai. Acceleration of a relative positioning framework. *International Journal of Computational Intelligence Research*, 5:130–140, 2009.

[Haj82] Bruce Hajek. Hitting-time and occupation-time bounds implied by drift analysis with applications. *Advances in Applied Probability*, 14(3):502–525, 1982.

[HBM13] Sabine Helwig, Jürgen Branke, and Sanaz Mostaghim. Experimental analysis of bound handling techniques in particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 17(2):259–271, April 2013. doi:10.1109/TEVC.2012.2189404 .

[HE02a] Xiaohui Hu and Russell Eberhart. Multiobjective optimization using dynamic neighborhood particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1677–1681, Washington, DC, USA, 2002. IEEE Computer Society. doi:10.1109/CEC.2002.1004494 .

[HE02b] Xiaohui Hu and Russell Eberhart. Solving constrained nonlinear optimization problems with particle swarm optimization.

In *Proceedings of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI)*, pages 203–206, 2002.

[Hel10] Sabine Helwig. *Particle Swarms for Constrained Optimization*. PhD thesis, Department of Computer Science, University of Erlangen-Nuremberg, 2010. urn:nbn:de:bvb:29-opus-19334.

[HES03a] Xiaohui Hu, Russell C. Eberhart, and Yuhui Shi. Engineering optimization with particle swarm. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 53–57, 2003. doi:10.1109/SIS.2003.1202247.

[HES03b] Xiaohui Hu, Russell C. Eberhart, and Yuhui Shi. Swarm intelligence for permutation optimization: a case study of n-queens problem. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 243–246, 2003. doi:10.1109/SIS.2003.1202275 .

[HM06] Werner Halter and Sanaz Mostaghim. Bilevel optimization of multi-component chemical systems using particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1240–1247, 2006. doi:10.1109/CEC.2006.1688451.

[HNW09] Sabine Helwig, Frank Neumann, and Rolf Wanka. Particle swarm optimization with velocity adaptation. In *Proceedings of the International Conference on Adaptive and Intelligent Systems (ICAIS)*, pages 146–151, 2009. doi:10.1109/ICAIS.2009.32.

[HRM+11] Nikolaus Hansen, Raymond Ros, Nikolas Mauny, Marc Schoenauer, and Anne Auger. Impacts of invariance in search: When CMA-ES and PSO face ill-conditioned and non-separable problems. *Applied Soft Computing*, 11:5755–5769, 2011. doi:10.1016/j.asoc.2011.03.001 .

[HW08] Sabine Helwig and Rolf Wanka. Theoretical analysis of initial particle swarm behavior. In *Proceedings of the 10th International Conference on Parallel Problem Solving from Nature (PPSN)*, pages 889–898, 2008. doi:10.1007/978-3-540-87700-4_88.

[HY01] Jun He and Xin Yao. Drift analysis and average time complexity of evolutionary algorithms. *Artificial Intelligence*, 127(1):57–85, 2001. doi:10.1016/S0004-3702(01)00058-3.

[HY04] Jun He and Xin Yao. A study of drift analysis for estimating computation time of evolutionary algorithms. *Natural Computing*, 3(1):21–35, 2004. doi:10.1023/B:NACO.0000023417.31393.c7.

[Jäg03] Jens Jägersküpper. Analysis of a simple evolutionary algorithm for minimization in euclidean spaces. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 2719 of *Lecture Notes in Computer Science*, pages 1068–1079. Springer, 2003. doi:10.1007/3-540-45061-0_82.

[Jäg07] Jens Jägersküpper. Algorithmic analysis of a basic evolutionary algorithm for continuous optimization. *Theoretical Computer Science*, 379(3):329–347, 2007. doi:0.1016/j.tcs.2007.02.042.

[Jäg08] Jens Jägersküpper. A blend of markov-chain and drift analysis. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 41–51. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-87700-4_5.

[JHW08] Johannes Jordan, Sabine Helwig, and Rolf Wanka. Social interaction in particle swarm optimization, the ranked FIPS, and adaptive multi-swarms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 49–56, 2008. doi:10.1145/1389095.1389103.

[JLY07a] Ming Jiang, Yupin P. Luo, and Shiyuan Y. Yang. Particle swarm optimization – stochastic trajectory analysis and parameter selection. In Felix T. S. Chan and Manoj Kumar Tiwari, editors, *Swarm Intelligence – Focus on Ant and Particle Swarm Optimization*, pages 179–198. I-TECH Education and Publishing, 2007. Corrected version of [JLY07b].

[JLY07b] Ming Jiang, Yupin P. Luo, and Shiyuan Y. Yang. Stochastic convergence analysis and parameter selection of the stan-

dard particle swarm optimization algorithm. *Information Processing Letters (IPL)*, 102:8–16, 2007. Corrected by [JLY07a], doi:10.1016/j.ipl.2006.10.005.

[JM05] Stefan Janson and Martin Middendorf. A hierarchical particle swarm optimizer and its adaptive variant. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(6):1272–1282, 2005. doi:10.1109/TSMCB.2005.850530.

[KE95] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995. doi:10.1109/ICNN.1995.488968 .

[KE97] James Kennedy and Russell C. Eberhart. A discrete binary version of the particle swarm algorithm. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 4104–4108. IEEE Computer Society, 1997. doi:10.1109/ICSMC.1997.637339 .

[Ken97] James Kennedy. The particle swarm: Social adaptation of knowledge. In *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC)*, pages 303–308, 1997. doi:10.1109/ICEC.1997.592326.

[Ken99] James Kennedy. Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, volume 3, pages 1931–1938, 1999. doi:10.1109/CEC.1999.785509.

[Ken03] James Kennedy. Bare bones particle swarms. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 80–87, 2003. doi:10.1109/SIS.2003.1202251.

[KGV83] Scott Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi:10.1126/science.220.4598.671.

[Kle06] Achim Klenke. *Wahrscheinlichkeitstheorie*. Springer Berlin Heidelberg New York, 2006.

[KM02] James Kennedy and Rui Mendes. Population structure and particle swarm performance. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, volume 2, pages 1671–1676, 2002. doi:10.1109/CEC.2002.1004493.

[KM06] James Kennedy and Rui Mendes. Neighborhood topologies in fully informed and best-of-neighborhood particle swarms. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 36(4):515–519, 2006. doi:10.1109/TSMCC.2006.875410.

[KSF06] Visakan Kadirkamanathan, Kirusnapillai Selvarajah, and Peter J. Fleming. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation*, 10(3):245–255, 2006. doi:10.1109/TEVC.2005.857077.

[LS05a] J. J. Liang and Ponnuthurai N. Suganthan. Dynamic multiswarm particle swarm optimizer. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 124–129, 2005. doi:10.1109/SIS.2005.1501611.

[LS05b] J. J. Liang and Ponnuthurai N. Suganthan. Dynamic multiswarm particle swarm optimizer with local search. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, volume 1, pages 522–528, 2005. doi:10.1109/CEC.2005.1554727.

[LT01] Panta Lucic and Dusan Teodorovic. Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence. *Preprints of the TRISTAN IV triennial symposium on transportation analysis*, pages 441–445, 2001.

[LW11] Per Kristian Lehre and Carsten Witt. Finite first hitting time versus stochastic convergence in particle swarm optimisation, 2011. arXiv:1105.5540.

[MC99] Jacqueline Moore and Richard Chapman. Application of particle swarm to multiobjective optimization. Department of Computer Science and Software Engineering, Auburn University. (Unpublished manuscript), 1999.

[MF02] Vladimiro Miranda and Nuno Fonseca. Epso-evolutionary particle swarm optimization, a new algorithm with applications in

power systems. In *Proceedings of the IEEE/PES Transmission and Distribution Conference and Exhibition*, volume 2, pages 745–750, 2002. doi:10.1109/TDC.2002.1177567.

[MKN03] Rui Mendes, James Kennedy, and José Neves. Watch thy neighbor or how the swarm can learn from its environment. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 88–94, 2003. doi:10.1109/SIS.2003.1202252.

[MKN04] Rui Mendes, James Kennedy, and José Neves. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, 2004. doi:10.1109/TEVC.2004.826074.

[MN04] Rui Mendes and José Neves. What makes a successful society? experiments with population topologies in particle swarms. In Ana L. C. Bazzan and Sofiane Labidi, editors, *Proceedings of the Brazilian Symposium on Artificial Intelligence (SBIA)*, volume 3171 of *Lecture Notes in Computer Science*, pages 346–355. Springer, 2004. doi:10.1007/978-3-540-28645-5_35.

[mpfl4] The GNU multiple precision arithmetic library. https://gmplib.org/, 2014.

[MS96] Zbigniew Michalewicz and Marc Schoenauer. Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, 4(1):1–32, 1996. doi:10.1162/evco.1996.4.1.1.

[MS05] Christopher K. Monson and Kevin D. Seppi. Linear equality constraints and homomorphous mappings in PSO. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 73–80. IEEE, 2005. doi:10.1109/CEC.2005.1554669.

[MT03] Sanaz Mostaghim and Jürgen Teich. Strategies for finding good local guides in multi-objective particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, 2003. doi:10.1109/SIS.2003.1202243 .

[MWP04] Arvind S. Mohais, Christopher Ward, and Christian Posthoff. Randomized directed neighborhoods with edge migration in particle swarm optimization. In *Proceedings of the Congress*

*on Evolutionary Computation (CEC)*, volume 1, pages 548–555, 2004. doi:10.1109/CEC.2004.1330905.

[NSW09] Frank Neumann, Dirk Sudholt, and Carsten Witt. Analysis of different mmas aco algorithms on unimodal functions and plateaus. *Swarm Intelligence*, 3(1):35–68, 2009. doi:10.1007/s11721-008-0023-3.

[NW07] Frank Neumann and Ingo Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theoretical Computer Science*, 378(1):32–40, 2007. doi:10.1016/j.tcs.2006.11.002.

[NW08] Frank Neumann and Carsten Witt. Ant colony optimization and the minimum spanning tree problem. In Vittorio Maniezzo, Roberto Battiti, and Jean-Paul Watson, editors, *Learning and Intelligent Optimization*, volume 5313 of *Lecture Notes in Computer Science*, pages 153–166. Springer Berlin Heidelberg, 2008. doi:10.1007/978-3-540-92695-5_12.

[OD10] Jérôme E. Onwunalu and Louis J. Durlofsky. Application of a particle swarm optimization algorithm for determining optimum well location and type. *Computational Geosciences*, 14:183–198, 2010. doi:10.1007/s10596-009-9142-1.

[OH07] Alan Owen and Inman Harvey. Adapting particle swarm optimisation for fitness landscapes with neutrality. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 258–265, 2007. doi:10.1109/SIS.2007.367946.

[OHMW11] Ludmila Omeltschuk, Sabine Helwig, Moritz Mühlenthaler, and Rolf Wanka. Heterogeneous constraint handling for particle swarm optimization. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, 2011. doi:10.1109/SIS.2011.5952578.

[OM99] Ender Ozcan and Chilukuri K. Mohan. Particle swarm optimization: Surfing the waves. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the IEEE Congress of Evolutionary Computation (CEC)*, volume 3, pages 1939–1944. IEEE Press, 1999. doi:10.1109/CEC.1999.785510.

[PB07] Riccardo Poli and David S. Broomhead. Exact analysis of the sampling distribution for the canonical particle swarm optimiser and its convergence during stagnation. In Hod Lipson, editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 134–141. ACM, 2007. doi:10.1145/1276958.1276977.

[PBBK07] Riccardo Poli, Dan Bratton, Tim Blackwell, and James Kennedy. Theoretical derivation, analysis and empirical evaluation of a simpler particle swarm optimiser. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pages 1955–1962, 2007. doi:10.1109/CEC.2007.4424713.

[PC04] Gregorio Toscano Pulido and Carlos A. Coello Coello. A constraint-handling mechanism for particle swarm optimization. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, volume 2, pages 1396–1403, 2004. doi:10.1109/CEC.2004.1331060.

[PC10] Magnus E. H. Pedersen and Andrew J. Chipperfield. Simplifying particle swarm optimization. *Applied Soft Computing*, 10(2):618–628, 2010. doi:10.1016/j.asoc.2009.08.029.

[PE03] Ulrich Paquet and Andries P. Engelbrecht. A new particle swarm optimiser for linearly constrained optimisation. In *Proceedings of the Congress on Evolutionary Computation (CEC)*, volume 1, pages 227–233, 2003. doi:10.1109/CEC.2003.1299579.

[PKB07] Riccardo Poli, James Kennedy, and Tim Blackwell. Particle swarm optimization – an overview. *Swarm Intelligence*, 1(1):33–57, 2007. doi:10.1007/s11721-007-0002-0.

[PL07] Riccardo Poli and William B. Langdon. Markov chain models of bare-bones particle swarm optimizers. In *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, pages 142–149. ACM, 2007. doi:10.1145/1276958.1276978.

[PLCS07] Riccardo Poli, William B. Langdon, Maurice Clerc, and Christopher R. Stephens. Continuous optimisation theory made easy? finite-element models of evolutionary strategies, genetic algorithms and particle swarm optimizers. In *Proceedings of the*

*ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA)*, pages 165–193, 2007. doi:10.1007/978-3-540-73482-6_10.

[Pol08] Riccardo Poli. Dynamics and stability of the sampling distribution of particle swarm optimisers via moment analysis. *Journal of Artificial Evolution and Applications*, 2008:15:1–15:10, 2008. doi:10.1155/2008/761459.

[PSL11] Bijaya Ketan Panigrahi, Yuhui Shi, and Meng-Hiot Lim, editors. *Handbook of Swarm Intelligence — Concepts, Principles and Applications*. Springer, 2011. doi:0.1007/978-3-642-17390-5.

[PV02a] Konstantinos E. Parsopoulos and Michael N. Vrahatis. Particle swarm optimization method for constrained optimization problems. In *Proceedings of the Euro-International Symposium on Computational Intelligence (EISCI)*, pages 214–220. IOS Press, 2002.

[PV02b] Konstantinos E. Parsopoulos and Michael N. Vrahatis. Particle swarm optimization method in multiobjective problems. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 603–607. ACM Press, 2002. doi:10.1145/508791.508907.

[Raß14] Alexander Raß. Explanation of stagnation at points that are not local optima in particle swarm optimization by potential analysis. Master's thesis, Department of Computer Science, University of Erlangen-Nuremberg, 2014.

[RHW04] A. Ratnaweera, S. Halgamuge, and H.C. Watson. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*, 8(3):240–255, 2004. doi:10.1109/TEVC.2004.826071.

[RHW10] Thomas Ritscher, Sabine Helwig, and Rolf Wanka. Design and experimental evaluation of multiple adaptation layers in self-optimizing particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, 2010. doi:10.1109/C EC.2010.5586255.

[RIR14] Retrospective image registration evaluation project (rire). http://www.insight-journal.org/rire/, 2014.

[Ros60] Howard H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3:175–184, 1960. doi:0.1093/comjnl/3.3.175.

[RPPN09] Kiruthika Ramanathan, V. M. Periasamy, Malathy Pushpavanam, and U. Natarajan. Particle swarm optimisation of hardness in nickel diamond electro composites. *Archives of Computational Materials Science and Surface Engineering*, 1:232–236, 2009. http://archicmsse.org/vol09_4/0945.pdf.

[RRS04] Jacob Robinson and Yahya Rahmat-Samii. Particle swarm optimization in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 52(2):397–407, 2004. doi:10.1109/tap.2004.823969.

[RV03] Mark Richards and Dan Ventura. Dynamic sociometry in particle swarm optimization. *Proceedings of the Joint Conference on Information Sciences*, pages 1557–1560, 2003.

[RV04] Mark Richards and Dan Ventura. Choosing a starting configuration for particle swarm optimization. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, volume 3, pages 2309–2312, 2004. doi:10.1109/IJCNN.2004.1380986.

[SC06] Margarita Reyes Sierra and Carlos A. Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

[Sch14] Lydia Schwab. Einsatz der partikelschwarmoptimierung zur bildregistrierung in der medizinischen bildverarbeitung. Master's thesis, Department of Computer Science, University of Erlangen-Nuremberg, 2014.

[SE98] Yuhui Shi and Russell Eberhart. A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC)*, pages 69–73, 1998. doi:10.1109/ICEC.1998.699146.

[SE99] Yuhui Shi and Russell C. Eberhart. Empirical study of particle swarm optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, volume 3, pages 1945–1949, 1999. doi:10.1109/CEC.1999.785511.

[SFX04] Jun Sun, Bin Feng, and Wenbo Xu. Particle swarm optimization with particles having quantum behavior. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 325–331, June 2004. doi:10.1109/CEC.2004.1330875.

[SHL+05] Ponnuthurai N. Suganthan, Nikolaus Hansen, J. J. Liang, Kalyanmoy Deb, Ying ping Chen, Anne Auger, and Santosh Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical report, Nanyang Technological University, Singapore, 2005.

[SK06] Belram Suman and Prabhat Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, 57(18):1143–1160, 2006. doi:10.1057/palgrave.jors.2602068.

[SLL+07] X. H. Shi, Yun C. Liang, Heow P. Lee, C. Lu, and Q.X. Wang. Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, 103:169–176, 2007. doi:10.1016/j.ipl.2007.03.010.

[SP97] Rainer Storn and Kenneth Price. Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. doi:10.1023/a%253a1008202821328.

[STW04] Jens Scharnow, Karsten Tinnefeld, and Ingo Wegener. The analysis of evolutionary algorithms on sorting and shortest paths problems. *Journal of Mathematical Modelling and Algorithms*, 3(4):349–366, 2004. doi:10.1007/s10852-005-2584-0.

[Sug99] Ponnuthurai N. Suganthan. Particle swarm optimiser with neighbourhood operator. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, volume 3, 1999. doi:10.1109/CEC.1999.785514.

[SW08] Dirk Sudholt and Carsten Witt. Runtime analysis of binary pso. In Conor Ryan and Maarten Keijzer, editors, *Proceedings of the Conference on Genetic and Evolutionary Computation (GECCO)*, pages 135–142. ACM, 2008. doi:10.1145/1389095.1389114.

[Tre03] Ioan Cristian Trelea. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters*, 85:317–325, 2003. doi:10.1016/S0020-0190(02)00447-7.

[vdBE02] Frans van den Bergh and Andries P. Engelbrecht. A new locally convergent particle swarm optimiser. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, volume 3, pages 94–99, 2002. doi:10.1109/ICSMC.2002.1176018.

[vdBE06] Frans van den Bergh and Andries P. Engelbrecht. A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8):937–971, 2006. doi:10.1016/j.ins.2005.02.003.

[vdBE10] Frans van den Bergh and Andries P. Engelbrecht. A convergence proof for the particle swarm optimiser. *Fundamenta Informaticae*, 105(4):341–374, 2010. doi:10.3233/FI-2010-370.

[VOK07] Kalyan Veeramachaneni, Lisa Osadciw, and Ganapathi Kamath. Probabilistically driven particle swarms for optimization of multi valued discrete problems: Design and analysis. In *Proceedings of the IEEE Swarm Intelligence Symposium (SIS)*, pages 141–149, 2007. doi:10.1109/SIS.2007.368038.

[Wit09] Carsten Witt. Why standard particle swarm optimisers elude a theoretical runtime analysis. In *Proceedings of the 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA)*, pages 13–20, 2009. doi:10.1145/1527125.1527128.

[WM97] David Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. doi:10.1109/4235.585893.

[WSZ+04] Mark P. Wachowiak, Renata Smolíková, Yufeng Zheng, Jacek M. Zurada, and Adel S. Elmaghraby. An approach to multimodal

biomedical image registration utilizing particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8:289–301, 2004. doi:10.1109/TEVC.2004.826068.

[Yan09] Xin-She Yang. Firefly algorithms for multimodal optimization. In Osamu Watanabe and Thomas Zeugmann, editors, *Stochastic Algorithms: Foundations and Applications*, volume 5792 of *Lecture Notes in Computer Science*, pages 169–178. Springer Berlin Heidelberg, 2009. doi:10.1007/978-3-642-04944-6_14.

[Yan10] Xin-She Yang. A new metaheuristic bat-inspired algorithm. In Juan Ramón González, David A. Pelta, Carlos Cruz, Germán Terrazas, and Natalio Krasnogor, editors, *Proceedings of the International Workshop on Nature Inspired Cooperative Strategies for Optimization (NICSO)*, volume 284 of *Studies in Computational Intelligence*, pages 65–74. Springer, 2010. doi:10.1007/978-3-642-12538-6_6.

[YD09] Xin-She Yang and Suash Deb. Cuckoo search via Lévy flights. In *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pages 210–214. IEEE, 2009. doi:10.1109/NABIC.2009.5393690.

[YII03] Keiichiro Yasuda, Azuma Ide, and Nobuhiro Iwasaki. Adaptive particle swarm optimization. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, volume 2, pages 1554–1559, 2003. doi:10.1109/ICSMC.2003.1244633.

[YKF+00] Hirotaka Yoshida, Kenichi Kawata, Yoshikazu Fukuyama, Shinichi Takayama, and Yosuke Nakanishi. A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Transactions on Power Systems*, 15(4):1232–1239, 2000. doi:10.1109/59.898095.

[ZWLK09] Karin Zielinski, Petra Weitkemper, Rainer Laur, and Karl-Dirk Kammeyer. Optimization of power allocation for interference cancellation with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 13(1):128–150, 2009. doi:10.1109/TEVC.2008.920672.

[ZXB04] Wenjun Zhang, Xiao-Feng Xie, and De-Chun Bi. Handling boundary constraints for numerical optimization by particle swarm flying in periodic search space. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, volume 2, pages 2307–2311, 2004. doi:10.1109/CEC.2004.1331185.

[ZZLL07] Jing-Ru Zhang, Jun Zhang, Tat-Ming Lok, and Michael R. Lyu. A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training. *Applied Mathematics and Computation*, 185(2):1026–1037, 2007. doi:10.1016/j.amc.2006.07.025.

# Author's Own Publications


[BSW14*] Bernd Bassimir, Manuel Schmitt, and Rolf Wanka. How much forcing is necessary to let the results of particle swarms converge? In *Proc. 1st Int. Conf. in Swarm Intelligence Based Optimization (ICSIBO)*, pages 98–105, 2014. doi:10.1007/978-3-319-12970-9_11.

[LSW14*] Vanessa Lange, Manuel Schmitt, and Rolf Wanka. Towards a better understanding of the local attractor in Particle Swarm Optimization: Speed and solution quality. In *Proc. 3rd International Conference on Adaptive and Intelligent Systems (ICAIS)*, pages 90–99, 2014. doi:10.1007/978-3-319-11298-5_10.

[SW13a*] Manuel Schmitt and Rolf Wanka. Exploiting independent subformulas: A faster approximation scheme for #k-SAT. *Information Processing Letters (IPL)*, 113:337–344, 2013. doi:10.1016/j.ipl.2013.02.013.

[SW13b*] Manuel Schmitt and Rolf Wanka. Particle swarm optimization almost surely finds local optima. In *Proc. 15th Genetic and Evolutionary Computation Conference (GECCO)*, pages 1629–1636, 2013. doi:10.1145/2463372.2463563.

[SW13c*] Manuel Schmitt and Rolf Wanka. Particles prefer walking along the axes: Experimental insights into the behavior of a particle swarm. In *Companion of Proc. 15th Genetic and Evolutionary Computation Conference (GECCO)*, pages 17–18, 2013. Full version arXiv:1303.6145.

[SW15*] Manuel Schmitt and Rolf Wanka. Particle swarm optimization almost surely finds local optima. *Theoretical Computer Science*, 561A:57–72, 2015. doi:10.1016/j.tcs.2014.05.017.

# Acronyms

| | |
|---|---|
| **PSO** | particle swarm optimization |
| **VVC** | Volt/Var Control |
| **UMTS** | Universal Mobile Telecommunications System |
| **CDMA** | Code Division Multiple Access |
| **EA** | evolutionary algorithm |
| **H-PSO** | Hierarchical PSO |
| **FIPS** | fully informed particle swarm |
| **GCPSO** | Guaranteed Convergence PSO |
| **QPSO** | Quantum PSO |
| **MOPSO** | multi-objective PSO |
| **TSP** | Traveling Salesperson Problem |
| **SA** | Simulated Annealing |
| **AA** | ant algorithm |
| **SSSP** | single source shortest path |
| **MM** | maximum matching |
| **MST** | minimum spanning tree |
| **NFL** | No Free Lunch |
| **CT** | Computer Tomography |
| **MRI** | Magnetic Resonance Imaging |

# Index

Particle swarm optimization (PSO) is a very popular, randomized, nature-inspired meta-heuristic for solving continuous black box optimization problems. The main idea is to mimic the behavior of natural swarms like, e. g., bird flocks and fish swarms that find pleasant regions by sharing information. The movement of a particle is influenced not only by its own experience, but also by the experiences of its swarm members.

In this thesis, we study the convergence process in detail. In order to measure how far the swarm at a certain time is already converged, we define and analyze the potential of a particle swarm. This potential analysis leads to the proof that in a 1-dimensional situation, the swarm with probability 1 converges towards a local optimum for a comparatively wide range of objective functions. Additionally, we apply drift theory in order to prove that for unimodal objective functions the result of the PSO algorithm agrees with the actual optimum in $k$ digits after time $\mathcal{O}(k)$.

In the general D-dimensional case, it turns out that the swarm might not converge towards a local optimum. Instead, it gets stuck in a situation where some dimensions have a potential that is orders of magnitude smaller than others. Such dimensions with a too small potential lose their influence on the behavior of the algorithm, and therefore the respective entries are not optimized. In the end, the swarm stagnates, i. e., it converges towards a point in the search space that is not even a local optimum. In order to solve this issue, we propose a slightly modified PSO that again guarantees convergence towards a local optimum.

ISBN 978-3-944057-30-9

FAU UNIVERSITY PRESS 2015