An Analysis of Particle Swarm Optimizers

by

Frans van den Bergh

Abstract

Many scientific, engineering and economic problems involve the optimisation of a set of parameters. These problems include examples like minimising the losses in a power grid by finding the optimal configuration of the components, or training a neural network to recognise images of people's faces. Numerous optimisation algorithms have been proposed to solve these problems, with varying degrees of success. The Particle Swarm Optimiser (PSO) is a relatively new technique that has been empirically shown to perform well on many of these optimisation problems. This thesis presents a theoretical model that can be used to describe the long-term behaviour of the algorithm. An enhanced version of the Particle Swarm Optimiser is constructed and shown to have guaranteed convergence on local minima. This algorithm is extended further, resulting in an algorithm with guaranteed convergence on global minima. A model for constructing cooperative PSO algorithms is developed, resulting in the introduction of two new PSO-based algorithms. Empirical results are presented to support the theoretical properties predicted by the various models, using synthetic benchmark functions to investigate specific properties. The various PSO-based algorithms are then applied to the task of training neural networks, corroborating the results obtained on the synthetic benchmark functions.

Thesis supervisor: Prof. A. P. Engelbrecht
Department of Computer Science
Degree: Philosophiae Doctor

Chapter 1

Introduction

> *You awaken to the sound of your alarm clock. A clock that was manufactured by a company that tried to maximise its profit by looking for the optimal allocation of the resources under its control. You turn on the kettle to make some coffee, without thinking about the great lengths that the power company went to in order to optimise the delivery of your electricity. Thousands of variables in the power network were configured to minimise the losses in the network in an attempt to maximise the profit of your electricity provider. You climb into your car and start the engine without appreciating the complexity of this small miracle of engineering. Thousands of parameters were fine-tuned by the manufacturer to deliver a vehicle that would live up to your expectations, ranging from the aesthetic appeal of the bodywork to the specially shaped side-mirror cowls, designed to minimise drag. As you hit the gridlock traffic, you think "Couldn't the city planners have optimised the road layout so that I could get to work in under an hour?"*

Optimisation forms an important part of our day-to-day life. Many scientific, social, economic and engineering problems have parameters that can be adjusted to produce a more desirable outcome.

Over the years numerous techniques have been developed to solve such optimisation problems. This thesis investigates the behaviour of a relatively new technique known as Particle Swarm Optimisation, a technique that solves problems by simulating swarm behaviour.

1.1 Motivation

It is clear that there will always be a need for better optimisation algorithms, since the complexity of the problems that we attempt to solve is ever increasing. The Particle Swarm Optimiser was introduced in 1995 [38, 70], yet very few formal analyses of the behaviour of the algorithm have been published. Most of the published work was concerned with empirical results obtained by changing some aspect of the original algorithm.

Without a formal model of why the algorithm works, it was impossible to determine what the behaviour of the algorithm would be in the general case. If the algorithm has been shown to be able to solve 10 difficult optimisation problems, what could be said about the infinite number of problems that have not yet been studied empirically?

While the results obtained from empirical comparisons provided useful insights into the nature of the PSO algorithm, it was clear that a general, theoretical description of the behaviour of the algorithm was needed. This thesis constructs such a model, which is subsequently used to analyse the convergence behaviour of the PSO algorithm.

Several new PSO-based algorithms were subsequently developed, with the aid of the theoretical model of the PSO algorithm. These algorithms were constructed to address specific weaknesses of the PSO algorithm that only became apparent once the theoretical convergence behaviour of the PSO was understood.

1.2 Objectives

The primary objectives of this thesis can be summarised as follows:

- To develop a theoretical model for the convergence behaviour of the Particle Swarm Optimisation algorithm, and the various derived algorithms introduced in this thesis.

- To extend the PSO algorithm so that it becomes a global optimisation technique with guaranteed convergence on global optima.

- To develop and test cooperative Particle Swarm Optimisation algorithms, based on models that have proven to be successful when applied to other evolutionary algorithms.
- To obtain empirical results to support the predictions offered by the theoretical models.
- To investigate the application of various PSO-based algorithms to the task of training summation and product unit neural networks.

1.3 Methodology

The theoretical models developed in this thesis are used to characterise the behaviour of all the newly introduced algorithms. Each new algorithm is theoretically analysed to show whether it is guaranteed to converge on either a local or global minimum, depending on whether the algorithm is a local or global search algorithm, respectively.

Empirical results were obtained using various synthetic benchmark functions with well-known characteristics. These results are used as supporting evidence for the theoretical convergence characteristics of the various algorithms. Owing to the stochastic nature of all these algorithms, it is not always possible to directly observe the characteristics predicted by the theoretical model, *i.e.* a stochastic global optimisation algorithm may require an infinite number of iterations to guarantee that it will find the global minimiser. Therefore the probability of observing this algorithm locate a global minimiser in a finite number of iterations is very small. Despite this problem, it is still possible to see whether the algorithm is still making progress toward its goal, or whether it has become trapped in a local minimum.

The results of two Genetic Algorithm-based optimisation techniques are also reported for the same synthetic benchmark functions. These results provide some idea of the relative performance of the PSO-based techniques when compared to other stochastic, population-based algorithms.

A second set of experiments were performed on a real-world problem to act as a control for the results obtained on the synthetic functions. The task of training both

summation and product unit neural networks was selected as an example of a real-world optimisation problem. On these problems the results of the PSO-based algorithms were compared to that of the GA-based algorithms, as well as that of two efficient gradient-based algorithms.

1.4 Contributions

The main contributions of this thesis are:

- A theoretical analysis of the behaviour of the PSO under different parameter settings. This analysis led to the development of a model that can be used to predict the long-term behaviour of a specific set of parameters, so that these parameters can be classified as leading to convergent or divergent particle trajectories.
- The discovery that the original PSO algorithm is not guaranteed to converge on a local (or global) minimiser. An extension to the existing PSO algorithm is presented that enables the development of a formal proof of guaranteed local convergence.
- The development of a technique for extending the PSO algorithm so that it is guaranteed to be able to locate the global minimiser of the objective function, together with a formal proof of this property.
- The application of existing cooperative models to the PSO algorithm, leading to two new PSO-based algorithms. These new algorithms offer a significant improvement in performance on multi-modal functions. The existing cooperation model is then extended to produce a new type of cooperative algorithm that does not suffer from the same weaknesses as the original model.

1.5 Thesis Outline

Chapter 2 starts with an introduction to the theory of optimisation, followed by a brief review of existing evolutionary techniques for solving optimisation problems. This is

followed by a description of the Particle Swarm Optimiser, including a discussion of the numerous published modifications to the PSO algorithm. The focus then shifts slightly to the topic of coevolutionary algorithms, since these methods form the basis of the work presented in Chapter 4.

Chapter 3 presents a theoretical analysis of the behaviour of the PSO algorithm, including formal proofs of convergence for the various new PSO-based algorithms introduced there.

Several cooperative PSO algorithms, based on the models discussed in Chapter 2, are introduced in Chapter 4. The convergence properties of these cooperative algorithms are investigated, with formal proofs where applicable.

Chapter 5 presents an empirical analysis of the behaviour of the various PSO-based algorithms introduced in Chapters 3 and 4, applied to minimisation tasks involving synthetic benchmark functions. These synthetic functions allow specific aspects of PSO behaviour to be tested.

In Chapter 6, the same PSO-based algorithms are used to train summation and product unit networks. These results are presented to show that the new algorithms introduced in this thesis have similar performance on both real-world and synthetic minimisation tasks.

Chapter 7 presents a summary of the findings of this thesis. Some topics for future research are also discussed.

The appendices present, in order, a glossary of terms, a definition of frequently used symbols, a derivation of the closed-form PSO equations, a set of 3D-plots of the synthetic benchmark functions used in Chapter 5, a description of the gradient-based algorithms used in Chapter 6 and a list of publications derived from the work presented in this thesis.

Chapter 2

Background & Literature Study

This chapter reviews some of the basic definitions related to optimisation. A brief discussion of Evolutionary Algorithms and Genetic Algorithms is presented. The origins of the Particle Swarm optimiser are then discussed, followed by an overview of the various published modifications to the basic PSO algorithm. Next an introduction to coevolutionary and cooperative algorithms is presented, followed by a brief overview of the important issues that arise when cooperative algorithms are implemented.

2.1 Optimisation

The task of optimisation is that of determining the values of a set of parameters so that some measure of optimality is satisfied, subject to certain constraints. This task is of great importance to many professions, for example, physicists, chemists and engineers are interested in design optimisation when designing a chemical plant to maximise production, subject to certain constraints, *e.g.* cost and pollution. Scientists require optimisation techniques when performing non-linear curve or model fitting. Economists and operation researchers have to consider the optimal allocation of resources in industrial and social settings. Some of these problems involve only linear models, resulting in *linear optimisation* problems, for which an efficient technique known as linear programming [58] exists. The other problems are known as *non-linear optimisation* problems, which are generally very difficult to solve. These problems are the focus of the work

presented in this thesis.

The term optimisation refers to both minimisation and maximisation tasks. A task involving the maximisation of the function f is equivalent to the task of minimising $-f$, therefore the terms minimisation, maximisation and optimisation are used interchangeably.

This thesis deals mostly with *unconstrained minimisation* tasks, formally defined as

$$\begin{aligned} &\text{Given } f : \mathbb{R}^n \to \mathbb{R} \\ &\text{find } \mathbf{x}^* \in \mathbb{R}^n \text{ for which } f(\mathbf{x}^*) \leq f(\mathbf{x}),\ \forall \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{2.1}$$

Some problems require that some of the parameters satisfy certain constraints, *e.g.* all the parameters must be non-negative. These types of problems are known as *constrained minimisation* tasks. They are typically harder to solve than their equivalent unconstrained versions, and are not dealt with explicitly here.

Another class of optimisation problems are known as *least-squares* problems, which are of the form

$$\begin{aligned} &\text{Given } \mathbf{r} : \mathbb{R}^n \to \mathbb{R}^m, \quad n < m \\ &\text{find } \mathbf{x}^* \in \mathbb{R}^n \text{ for which } \sum_{i=1}^{m} (r_i(\mathbf{x}))^2 \text{ is minimised.} \end{aligned} \tag{2.2}$$

These optimisation problems present themselves when there are more non-linear requirements than there are degrees of freedom. Note that the least-squared problem can be solved using the same approach as used in solving (2.1), by defining

$$f(\mathbf{x}) = \sum_{i=1}^{m} (r_i(\mathbf{x}))^2$$

and minimising f. Neural Network training is sometimes solved as such a non-linear least-squares problem (see Chapter 6 for more details).

Techniques used to solve the minimisation problems defined above can be placed into two categories: Local and Global optimisation algorithms.

2.1.1 Local Optimisation

A *local minimiser*, $\mathbf{x}_B^*$, of the region B, is defined so that

$$f(\mathbf{x}_B^*) \leq f(\mathbf{x}),\ \forall \mathbf{x} \in B \tag{2.3}$$

where $B \subset S \subseteq \mathbb{R}^n$, and S denotes the search space. Note that $S = \mathbb{R}^n$ when dealing with unconstrained problems. More importantly, note that B is a proper subset of S. A given search space S can contain multiple regions B_i such that $B_i \cap B_j = \emptyset$ when $i \neq j$. It then follows that $\mathbf{x}^*_{B_i} \neq \mathbf{x}^*_{B_j}$, so that the minimiser of each region B_i is unique. Any of the $\mathbf{x}^*_{B_i}$ can be considered a minimiser of B, although they are merely local minimisers. There is no restriction on the value that the function can assume in the minimiser, so that $f(\mathbf{x}^*_{B_i}) = f(\mathbf{x}^*_{B_j})$ is allowed. The value $f(\mathbf{x}^*_{B_i})$ will be called the *local minimum*.

Most optimisation algorithms require a starting point $\mathbf{z}_0 \in S$. A local optimisation algorithm should guarantee that it will be able to find the minimiser $\mathbf{x}^*_B$ of the set B if $\mathbf{z}_0 \in B$. Some algorithms satisfy a slightly weaker constraint, namely that they guarantee to find a minimiser $\mathbf{x}^*_{B_i}$ of some set B_i, not necessarily the one closest to $\mathbf{z}_0$.

Many local optimisation algorithms have been proposed. A distinction will be made between deterministic, analytical algorithms and the stochastic algorithms discussed in Sections 2.2–2.4. The deterministic local optimisation[1] algorithms include simple Newton-Raphson algorithms, through Steepest Descent [11] and its many variants, including the Scaled Conjugate Gradient algorithm (SCG) [87] and the quasi-Newton [11, 30] family of algorithms. Some of the better known algorithms include Fletcher-Reeves (FR), Polar-Ribiere (PR), Davidon-Fletcher-Powell (DFP), Broyden-Fletcher-Goldfarb-Shanno (BFGS) [104, 11]. There's even an algorithm that was designed specifically for solving least-squares problems, known as the Levenberg-Marquardt (LM) algorithm [11].

2.1.2 Global Optimisation

The *global minimiser*, $\mathbf{x}^*$, is defined so that

$$f(\mathbf{x}^*) \leq f(\mathbf{x}), \ \forall \mathbf{x} \in S \tag{2.4}$$

where S is the search space. For unconstrained problems it is common to choose $S = \mathbb{R}^n$, where n is the dimension of $\mathbf{x}$. Throughout this thesis the term *global optimisation* will refer strictly to the process of finding $\mathbf{x}^*$ as defined in (2.4). The term *global minimum* will refer to the value $f(\mathbf{x}^*)$, and $\mathbf{x}^*$ will be called the *global minimiser*. A global optimisation

[1] see Section 2.1.2 for an explanation as to why these algorithms are classified as local methods here.

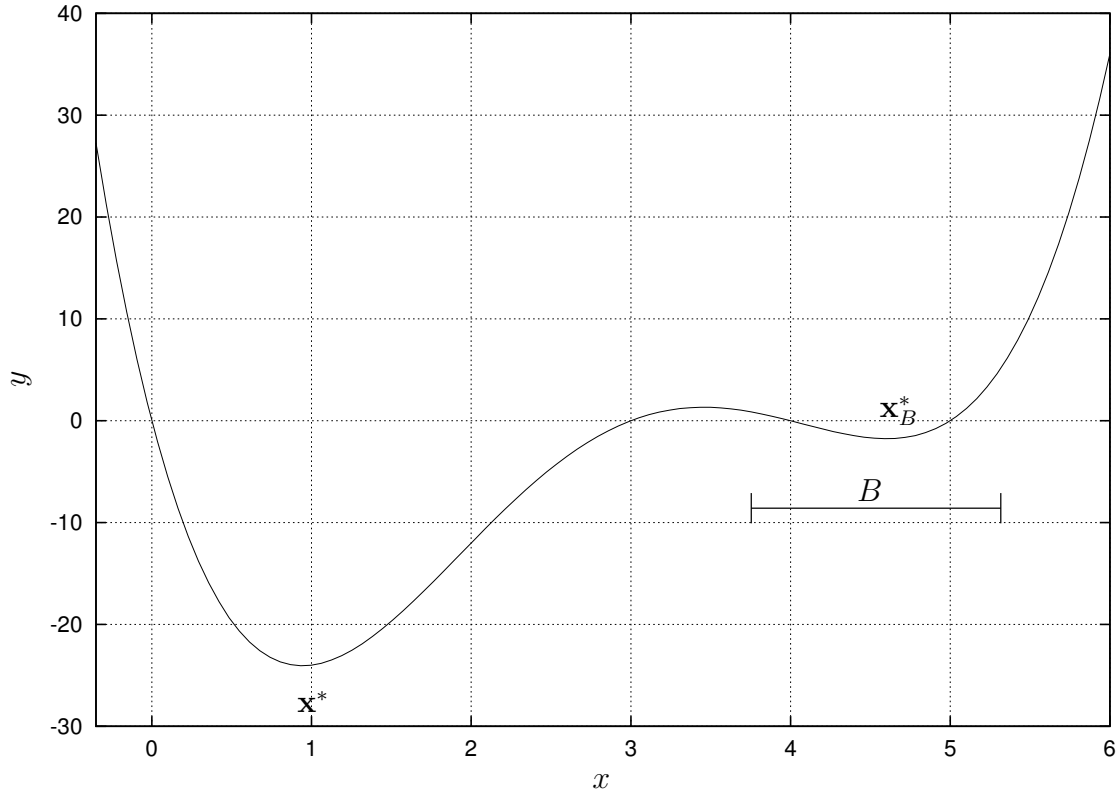

Figure 2.1: The function $f(x) = x^4 - 12x^3 + 47x^2 - 60x$, indicating the global minimiser $\mathbf{x}^*$, as well as a local minimiser $\mathbf{x}^*_B$.

algorithm, like the local optimisation algorithms described above, also starts by choosing an initial starting position $\mathbf{z}_0 \in S$.

Contrary to the definition above in (2.4), some texts (*e.g.* [30]) define a global optimisation algorithm differently, namely an algorithm that is able to find a (local) minimiser of $B \subset S$, regardless of the actual position of $\mathbf{z}_0$. These algorithms consist of two processes: "global" steps and "local" steps. Their local steps are usually the application of a local minimisation algorithm, and their "global" steps are designed to ensure that the algorithm will move into a region B_i, from where the "local" step will be able to find the minimiser of B_i. These methods will be referred to as *globally convergent* algorithms, meaning that they are able to converge to a local minimiser regardless of their starting position $\mathbf{z}_0$. These methods are also capable of finding the global minimiser, given that the starting position $\mathbf{z}_0$ is chosen correctly. There is no known reliable, general way of doing this, though.

Figure 2.1 illustrates the difference between the local minimiser $\mathbf{x}_B^*$ and the global minimiser $\mathbf{x}^*$. A true global optimisation algorithm will find $\mathbf{x}^*$ regardless of the choice of starting position $\mathbf{z}_0$. Dixon and Szegø have edited two collections of papers on the topic of true global optimisation algorithms [31, 32]. The topic of global optimisation algorithms will be revisited in Chapter 3.

2.1.3 No Free Lunch Theorem

One of the more interesting developments in optimisation theory was the publication of the "No Free Lunch" (NFL) theorem by Wolpert and Macready [144, 145]. This theorem states that the performance of all optimisation (search) algorithms, amortised over the set of all possible functions, is equivalent.

The implications of this theorem are far reaching, since it implies that no algorithm can be designed so that it will be superior to a linear enumeration of the search space, or even a purely random search. The theorem is only defined over finite search spaces, however, and it is as yet not clear whether the result applies to infinite search spaces. All computer implementations of search algorithms will effectively operate on finite search spaces, though, so the theorem is directly applicable to all existing algorithms.

Although the NFL theorem states that all algorithms perform equally well over the

set of *all* functions, it does not necessarily hold for all subsets of this set. The set of all functions over a finite domain includes the set of all the permutations of this domain. Many of these functions do not have compact descriptions, so that they appear to be largely "random". Most real-world functions, however, have some structure, and usually have compact descriptions. These types of functions form a rather small subset of the set of all functions. This concern lead to the development of sharpened versions of the NFL [117], showing that it holds for much smaller subsets than initially believed.

A more constructive approach is to try and characterise the set of functions over which the NFL does *not* hold. Christensen *et al.*proposed a definition of a "searchable" function [18], as well as a general algorithm that provably performs better than random search on this set of searchable functions.

This thesis will side with the latter approach, assuming that it is possible to design algorithms that perform, on average, better than others (*e.g.* random search) *over a limited subset* of the set of all functions. No further attempt will be made to characterise this subset. Instead, empirical results will be used to show that real-world applications can benefit from improved algorithms.

Reference

1. F. van den Bergh. Particle Swarm Weight Initialization in Multi-layer Perceptron Artificial Neural Networks. In *Development and Practice of Artificial Intelligence Techniques*, pages 41–45, Durban, South Africa, September 1999.

2. F. van den Bergh and A. P. Engelbrecht. Cooperative Learning in Neural Networks using Particle Swarm Optimizers. *South African Computer Journal*, (26):84–90, November 2000.

3. F. van den Bergh and A. P. Engelbrecht. Effects of Swarm Size on Cooperative Particle Swarm Optimisers. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 892–899, San Francisco, USA, July 2001.

4. F. van den Bergh and A. P. Engelbrecht. Training Product Unit Networks using Cooperative Particle Swarm Optimisers. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 126–132, Washington DC, USA, July 2001.

5. F. van den Bergh and A. P. Engelbrecht. A cooperative approach to particle swarm optimisation. *IEEE Transactions on Evolutionary Computation*. Submitted December 2000.