Chapter 1

Introduction

> *"What do you want to achieve or avoid? The answers to this question are objectives. How will you go about achieving your desire results? The answer to this you can call strategy." – William E Rothschild*

Imagine standing at the airport and looking at the display boards of arriving and departing flights. Suddenly a number of flights are indicated as being delayed, and as you check carefully your flight is one of them. You start to wonder whether you are going to miss your connecting flight and all of the effects that this delay can have on your schedule. However, at the air traffic control room, people start to think about other issues, such as: How will these delays influence the best way of handling all of the incoming and departing aeroplanes? How can they ensure that each plane's waiting time for either landing or take-off is minimised, but in such a way that the possibility of collisions is kept to zero?

The above is just one scenario of an every day life optimisation problem. The issues that the control room have to consider are called objectives. However, these objectives are in conflict with one another: by reducing the possibility of collissions, the waiting time of either landing or departing flights are increased, and vice versa. Furthermore, the delay of flights is an event that causes a change in the environment. Therefore, this is an example of a dynamic multi-objective optimisation problem (DMOOP).

The main objective of this thesis is to propose a new algorithm that solves DMOOPs efficiently.

1.1 Motivation

Most current research in the field of multi-objective optimisation (MOO) focusses on optimisation problems where all of the sub-objectives are static [31, 35, 36, 38]. Research on solving dynamic optimisation problems, on the other hand, strongly focusses on dynamic single-objective optimisation problems (DSOOPs) [11, 13, 37, 89].

However, optimisation problems that occur in situations of everyday life are normally not static in nature and have many objectives that have to be optimised, i.e. DMOOPs. One example of a real-life DMOOP is a steel production plant, where customers place an order for specific products that have to be delivered by a specified date. In order to produce a customer's order, the material has to go through specific production lines. Each production line consists of a number of machines that can only manage a certain load. Since many orders' material is managed in the production lines at the same time, and some orders may require the same machines, the order in which the material of the various orders move through the production line has to be optimised. Since machines can break down, requiring the production lines to be re-optimised, the optimisation of a production plant is an example of a DMOOP.

Multi-objective optimisation problems (MOOPs) with conflicting objectives do not have a single solution. Therefore, MOO algorithms aim to obtain a diverse set of non-dominated solutions, i.e. solutions that balance the trade-off between the various objectives, referred to as the Pareto-optimal front (POF). Another goal of multi-objective algorithms (MOAs) is to find a POF that is as close as possible to the true POF of the problem. Many MOAs store the found non-dominated solutions in an archive. Therefore, if an algorithm finds new non-dominated solutions, the new solutions are compared with the solutions in the archive. If a new solution is dominated by any of the solutions in the archive, it is not placed in the archive. Otherwise, the new solution is placed in the archive and any solutions in the archive that are dominated by the new solution are removed from the archive. When a change in the MOOP occurs, i.e. for example an objective function changes, the solutions in the archive are not necessarily valid for the new objective functions. Furthermore, solutions in the archive that were non-dominated before the change, may have become dominated after the change. Therefore, algorithms solving DMOOPs must have the ability to track the changing POF in order to find non-dominated solutions that are close to the new true POF, and to remove solutions from the archive that have become dominated after a change occurred in the environment.

Initially not much research has been done on dynamic multi-objective optimisation (DMOO) [1, 58, 117], but in the last few years more researchers focussed on DMOO [2, 17, 46, 67, 96, 100, 129, 135, 165, 156]. However, not much research has been done on solving DMOO using particle swarm optimisation (PSO) [102, 107]. This thesis proposes a new PSO-based DMOO algorithm, namely the dynamic Vector Evaluated Particle Swarm Optimisation (DVEPSO) algorithm.

In order to determine whether an algorithm can solve DMOOPs, functions with specific characteristics that are representative of typical real-world problems are required. These functions are normally referred to as *benchmark functions*. In the field of DMOO, there is a lack of standard benchmark functions and selecting the benchmark functions to test a new DMOO algorithm is not a trivial task. This thesis provides an overview of the benchmark functions that have been proposed in the DMOO literature and proposes new benchmark functions to address the identified limitations of the current DMOOPs. In addition, the characteristics of an ideal benchmark function suite is provided, as well as a list of DMOOPs for each of the identified characteristics.

Functions that quantify the performance of a DMOO algorithm, are referred to as *performance measures* or *performance metrics*. Similar to benchmark functions, there are no standard performance measures for DMOO. Therefore, this thesis provides an overview of the performance measures that are currently used to measure the performance of DMOO algorithms. Furthermore, issues with current DMOO performance measures are discussed.

1.2 Objectives

The primary objective of this thesis is to develop a PSO MOA for solving DMOOPs, namely DVEPSO. In achieving this main objective, the following sub-objectives have been identified:

- Identifying a set of benchmark functions representative of typical real-world problems.

- Identifying a set of performance measures that adequately quantifies the performance of a DMOO algorithm.
- The development and analysis of DVEPSO.

1.3 Contributions

The contributions of this thesis with regards to DMOOPs and performance measures for DMOO are:

- A comprehensive overview of the benchmark functions that are currently used in the DMOO literature.
- The identification of limitations of current DMOO benchmark functions.
- New DMOOPs that address the identified limitations of current DMOOP benchmark functions.
- An ideal DMOO benchmark function suite that contains:
  - characteristics that an ideal DMOOP suite should exhibit.
  - suggested DMOOPs for each identified characteristic.
- A comprehensive overview of performance measures that are currently used to measure the performance of DMOO algorithms.
- The identification of issues with current DMOO performance measures.

Through empirical analysis the following observations were made that contribute to knowledge in the fields of DMOO and PSO:

- Pareto-dominance based guide update approaches lead to improved performance over approaches that do not use Pareto-dominance information.
- Managing boundary constraint violations with the clamping (placing any particle that violates a specific boundary of the search space on or close to the violated boundary) approach produced the best performance.
- Re-initialising particles after a change in the environment occurs lead to improved performance over re-evaluation of the particles.
- For DMOOPs where the POF changes over time (Type II and Type III), removing all solutions from the archive after a change in the environment produced better results than re-evaluating the solutions and removing the solutions that became

dominated after the change. However, for Type I DMOOPs where the POF remains static, removing all solutions from the archive after a change lead to poor performance.

- PSO successfully solves DMOOPs of various types.

1.4 Research Methodology

Firstly, the DMOO literature was reviewed to determine the limitations with regards to:

- the development of DMOO algorithms, especially with reference to PSO algorithms.
- benchmark functions for DMOO. The review revealed that there are no standard benchmark functions for DMOO. Therefore, this thesis proposes an ideal set of DMOOPs that consists of current DMOOPs, as well as newly proposed DMOOPs.
- performance measures to determine whether these performance measures are adequate. Issues with regards to current DMOO performance measures were identified through empirical studies on DVEPSO. These issues are discussed and illustrated in this thesis.

Secondly, problems were identified with vector evaluated particle swarm optimisation (VEPSO) when solving DMOOPs. Therefore, various methods were proposed to adapt VEPSO for DMOO. An empirical analysis of DVEPSO was done to investigate the effect of these proposed changes on the performance of DVEPSO. Using formal hypothesis testing and statistical analysis, a final best performing configuration of DVEPSO was identified.

Thirdly, the best configuration of DVEPSO was compared against current state-of-the-art DMOO algorithms, namely:

- DNSGA-II-A and DNSGA-II-B, two NSGA-II algorithms adapted for DMOO and proposed by Deb *et al.* [46]. The source code of the static NSGA-II was obtained from [109] and adapted for DMOO according to [46].
- dCOEA, a dynamic competitive-cooperative coevolutionary algorithm proposed by Goh and Tan [67]. The source code of dCOEA was obtained from the first author of [67].

- MOPSO algorithm, a PSO algorithm adapted for DMOO by Lechuga [102].

For each of these state-of-the-art DMOO algorithms an empirical analysis was performed to determine the best configuration of the algorithm for the comparison study. Formal hypothesis testing and statistical analysis were performed to compare the performance of these DMOO algorithms and DVEPSO with one another.

1.5 Thesis Outline

The remainder of this thesis is organised in three main parts, namely optimisation background, computational intelligence algorithms and DVEPSO. The outline of each of these sections are provided next.

The outline of the part on optimisation background is as follows:

- **Chapter 2** presents the formal definitions of basic concepts required as background for various types of optimisation problems, namely single-objective optimisation problems (SOOPs), MOOPs and DMOOPs.
- **Chapter 3** provides an overview of DMOO benchmark functions that are currently used. Limitations of the DMOOPs are identified and new DMOOPs are proposed to address the limitations. An ideal set of benchmark functions are presented, highlighting the characteristics of an ideal benchmark function suite. Furthermore, example DMOOPs are suggested for each of the identified characteristics.
- **Chapter 4** provides an overview of DMOO performance measures. In addition, issues with currently used performance measures are illustrated and discussed.

The part on computational intelligence algorithms is organised as follows:

- **Chapter 5** provides basic background on single-objective optimisation (SOO) computational intelligence algorithms that are referred to later in this thesis. Basic concepts of PSO and genetic algorithms (GAs) are discussed.
- **Chapter 6** provides information about population-based algorithms that were used to solve MOOPs and that are referred to in later chapters of the thesis. A description of non-dominated sorting genetic algorithm II (NSGA-II), cooperative-coevolution evolutionary algorithm (CCEA) and multi-objective Particle Swarm Optimisation (MOPSO) are provided.

- **Chapter 7** covers vector-evaluated MOO algorithms. The vector evaluated genetic algorithm (VEGA), as well as the VEPSO algorithm that is inspired by VEGA, are discussed. A differential evolution (DE) version of VEGA, namely vector evaluated differential evolution (VEDE), is also discussed. Furthermore, information is provided about a hybrid algorithm that uses both VEPSO and VEDE to solve MOOPs.
- **Chapter 8** discusses population-based DMOO algorithms. Methods used by DMOO algorithms to detect and respond to changes are covered.

The part on DVEPSO discusses the DMOO algorithm that is proposed in the thesis. The outline of the DVEPSO part is:

- **Chapter 9** introduces the DVEPSO algorithm. The adaptation of VEPSO for DMOO, as well as the various parameters of DVEPSO, are discussed. New guide update approaches that use Pareto-dominance infmormation are proposed. An empirical study is performed to determine the influence of various guide update approaches on the performance of DVEPSO.
- **Chapter 10** presents an empirical study investigating the effect that various knowledge sharing approaches, approaches to manage boundary constraint violations and various responses to a change in the environment have on the performance of DVEPSO.
- **Chapter 11** investigates the performance of DVEPSO in comparison with other DMOO algorithms. An empirical study is discussed that compares the performance of DVEPSO with four other state-of-the-art DMOO algorithms.

Finally, **Chapter 12** concludes the work that has been presented in this thesis.

Additional information is provided in the Appendices as follows:

- **Appendix A** lists and defines the mathematical symbols used in this thesis, categorised according to the relevant chapter in which they appear.
- **Appendix B** provides a list of the important acronyms used or newly defined in the thesis, as well as their associated definitions.
- **Appendix C** discusses the calculation of a DMOOPs true POF.
- **Appendix D** presents the performance measure values and the p-values obtained

from the experiments discussed in Chapters 9, 10 and 11 respectively.

- **Appendix E** lists the publications derived from this research.