A Study of Simulated Annealing Techniques for Multi-Objective Optimisation

Submitted by Kevin Ian Smith, to the University of Exeter as a thesis for the degree of Doctor of Philosophy in Computer Science, October 2006.

I certify that all material in this thesis which is not my own work has been identified and that no material has previously been submitted and approved for the award of a degree by this or any other university.

Abstract

Many areas in which computational optimisation may be applied are multi-objective optimisation problems; those where multiple objectives must be minimised (for minimisation problems) or maximised (for maximisation problems). Where (as is usually the case) these are competing objectives, the optimisation involves the discovery of a set of solutions the quality of which cannot be distinguished without further preference information regarding the objectives. A large body of literature exists documenting the study and application of evolutionary algorithms to multi-objective optimisation, with particular focus being given to evolutionary strategy techniques which demonstrate the ability to converge to desired solutions rapidly on many problems.

Simulated annealing is a single-objective optimisation technique which is provably convergent, making it a tempting technique for extension to multi-objective optimisation. Previous proposals for extending simulated annealing to the multi-objective case have mostly taken the form of a traditional single-objective simulated annealer optimising a composite (often summed) function of the objectives. The first part of this thesis deals with introducing an alternate method for multi-objective simulated annealing, dealing with the dominance relation which operates without assigning preference information to the objectives. Non-generic improvements to this algorithm are presented, providing methods for generating more desirable suggestions for new solutions. This new method is shown to exhibit rapid convergence to the desired set, dependent upon the properties of the problem, with empirical results on a range of popular test problems with comparison to the popular NSGA-II genetic algorithm and a leading multi-objective simulated annealer from the literature. The new algorithm is applied to the commercial optimisation of CDMA mobile telecommunication networks and is shown to perform well upon this problem.

The second section of this thesis contains an investigation into the effects upon convergence of a range of optimiser properties. New algorithms are proposed with the properties desired to investigate. The relationship between evolutionary strategies and the simulated annealing techniques is illustrated, and explanation of the differing performance of the previously proposed algorithms across a standard test suite is given. The properties of problems on which simulated annealer approaches are desirable are investigated and new problems proposed to best provide comparisons between different simulated annealing techniques.

Acknowledgements

This thesis has benefitted from the contributions of many people in assorted styles of influence, in some parts contributing directly to my PhD research and in others providing me with motivation, outlets for frustrations, nutritional sustenance, or interventions in the name of sanity during stressful periods.

Within the academic community of the school I would like to thank my supervisors, Richard Everson and Dragan Savic. I would also like to thank Jonathan Fieldsend for his extensive collaborations and support over the course of recent years.

For their sponsorship, I thank Motorola and particularly Tim Charity, Rashmi Misra and Chris Murphy for their support.

Those outstanding PhD students I had the pleasure to share lunches, complaints, at times late nights working, and often receive ideas from also have my thanks: Mai Fadel, James Hood, Alex Schmolck, Jufen Zhang and particularly Michelle Fisher who suffered possession of the desk beside mine for much of my PhD and was consequently required to offer guidance throughout.

I would also like to thank David Barker, Chris Bazley, Jennifer Evans, Hannah Khirwadkar, Melanie Mitchell, Nick Murison, Matt Searle, Gareth Smith, Martin Smith and Remko Tronçon for their support and friendship in recent years. Especially deserving of gratitude are my parents, Colin and Sandra Smith, for their continuing support.

I give huge thanks to Catherine Morgan-James for her support throughout, for listening to my complaints, for proof-reading the thesis several times more than safe for any human, for encouragement during lows and enthusiasm during highs and, of course, for the best chocolate brownies on Earth.

Finally, I would like to thank Ed Keedwell and Xin Yao for examining my thesis and for the feedback they have given to strengthen it.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

- $\mathbf{x}$ Solution to the minimisation of $f(x)$.
- f_i Function mapping $\mathbf{x}$ to y_i.
- y_i i^{th} objective.
- x_i $i^(th)$ parameter.
- D Number of objectives.
- P Number of parameters.
- $\mathbf{x}'$ Perturbation of $\mathbf{x}$.
- $\mathcal{P}$ Pareto Front.
- F Estimated Pareto Front.
- $\prec$ The dominates relation.
- ω State.
- ω' Perturbation of ω.
- T Computational Temperature.
- T_0 Initial Computational Temperature.
- K Number of annealing epochs.
- $E(\omega)$ Energy of state ω
- $\delta E(\omega, \omega')$ Difference in energy changing from ω to ω'.
- $\bar{d}(F, \mathcal{P})$ Median distance of points in the set F from $\mathcal{P}$.
- $\mathcal{V}(\mathcal{P}, F)$ Volume of space dominated by $\mathcal{P}$ but not by F.

Chapter 1

Introduction

In this introductory chapter, the motivation for investigating simulated annealing methods for multi-objective optimisation is discussed, together with introductions into multi-objective optimisation and simulated annealing. The principal contributions of the work presented in this thesis are then presented and an overview of the content of the following chapters follows.

1.1 Motivation

Many problems are optimisation problems, those whereby the configuration of a system must be determined which will maximise the performance of that system. In an optimisation system with a single performance metric (objective), the aim is to locate the configuration which is superior to all other configurations. For real-world problems, it is usually the case that there is more than a single objective of the system which it is desirable to optimise. Such systems are said to represent multi-objective optimisation problems. As these objectives are generally competing it is no longer possible to find the single solution which is superior to all others in every objective and the aim then becomes to find all the solutions for which there exists no superior solution.

For problems with only a single objective, simulated annealing is an optimisation technique which is particularly desirable as there exists a proof that it is able to converge upon the optimal solution, even for problems with features which present difficulties for other popular optimisation techniques such as those where the location of the global optima is deceptive. As such, the motivation for this

work is to investigate the extension of simulated annealing to multi-objective optimisation.

1.1.1 Multi-Objective Optimisation

Optimisation problems often involve the simultaneous optimisation of competing objectives. One simple and intuitive example of a class of multi-objective problem is that of mechanical or electronic system design. In this case it is usual to want to minimise the cost of manufacture of the system, while maximising its performance; generally these are in competition (although many other objectives will often exist in such problems, such as maximising the efficiency and the reliability of such systems, for this introductory section the discussion is limited to two objectives). In such an optimisation there will be (potentially many) variables which describe the construction of the system, called the parameters of the optimisation problem, such as the combination and layout of components in a system design problem, and a configuration of these parameters is referred to as a solution. The performance of a solution to the problem can be evaluated for each objective to be optimised; doing this provides a series of values describing its performance (two values in the two-objective cost/performance example).

While it is clear that some solutions are wholly better than others, there exist many possible solutions which, when compared to each other, are not entirely better or worse; in these cases some objectives may be better while others are worse, leading to solutions which are not clearly ordered. Where one solution is wholly better than another, the better solution is said to dominate the poorer; a dominating solution must be no worse for any objective and must show an improvement in at least one objective. As optimisation in the multi-objective case is generally a compromise between competing solutions, it is usual for there to be a set of solutions, none of which dominate each other (it is said that they are mutually non-dominating), and for which no feasible configurations dominate them. This set is called the Pareto set, and represents the optimal trade-offs between objectives available. In the previous example, it represents all the configurations from those which are expensive but well-performing, through the possible compromises to the cheapest solution (which has poor performance). The Pareto set, also known as the Pareto frontier, was named after use of the concept in studies on the distribution of resources in an economy such as Pareto [1964].

Figure 1 illustrates these concepts for a two-objective minimisation problem, where it is desirable to have small values for each objective. The small circles each represent a parameterisation which has

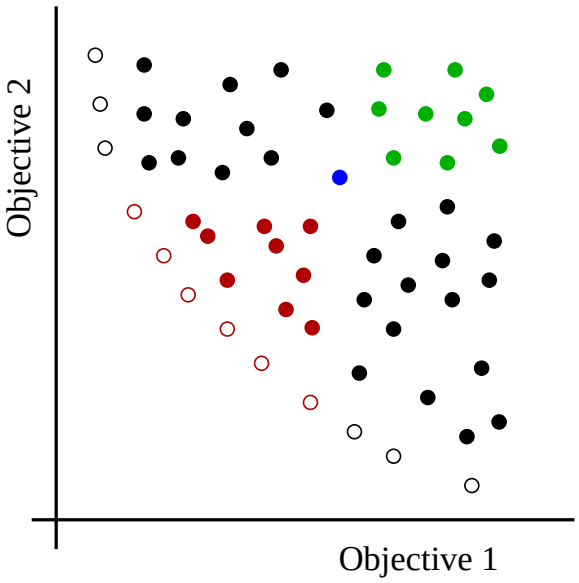

Figure 1: An illustration of dominance.

been evaluated and the corresponding point in the two-objective space plotted. Of these solutions, the unfilled circles represent the Pareto front; these are the points which no other point dominates. The points are coloured to illustrate their relative dominance to the blue point; those points in red dominate the blue point, while those points in green are dominated by the blue point, solutions in black neither dominate, nor are dominated by, the blue solution. The Pareto *set* is the set of parameterisations, the images of which lie upon the Pareto *front* in objective space.

The aim of multi-objective optimisation is therefore, given a definition of possible parameterisations and functions to evaluate the performance in each objective, to find the set of all parameter configurations which correspond to the members of the Pareto front: those configurations for which no configuration is possible for which the objective functions signify wholly better quality. As it is often the case that the Pareto front is continuous, containing infinitely many members, an approximation of the Pareto set is sought, aiming to contain many members, uniformly spaced across the full range, of the Pareto front.

A more extensive look at the concepts and terminology used in multi-objective optimisation is presented in Chapter 2. It is important to note that *many*-objective optimisation is not addressed in this thesis, and deals with problems having many objectives (far greater than 3).

1.1.2 Simulated Annealing

Simulated annealing [Kirkpatrick et al., 1983] is a single-objective optimisation technique inspired by the natural process of annealing solids. The physical process of annealing is the cooling of a metal sufficiently slowly so that it adopts a low-energy, crystalline state. When the temperature of the metal is high, the particles within the metal are able to move around, changing the structure of the metal, freely. As the temperature is lowered, the particles are limited in the movements they can make as many movements have a high energy cost and are increasingly limited to only those configurations with lower energy than the previous state. Simulated annealing draws inspiration from the physical process, in a computational model of the physical system.

The basic simulated annealing algorithm maintains both a state and a computational temperature, which is initially high and is reduced to (near-)zero during the algorithm's execution. The configuration is usually a solution to the optimisation, and at each iteration of the algorithm this solution is perturbed in some manner to produce a new solution. The quality of a solution is said to be the energy of the state, analogous with real-world annealing. The quality of both solutions is evaluated, using the objective function, and a new state is selected from the two solutions. When the new solution is no worse than the previous solution, the new solution is selected as the state. Where the new solution is of lower quality than the existing solution, it may be accepted with a probability dependent upon both the current computational temperature and the magnitude of the difference in quality. Solutions are most likely to be accepted if they are only slightly worse, or if the temperature is high. Largely inferior solutions are unlikely to be accepted, and as the temperature approaches zero, it becomes vanishingly unlikely that any inferior solution will be accepted.

Geman and Geman [1984] present a proof which states that if the algorithm is run for sufficiently many iterations, the final state will represent the optimal solution configuration. Simulated annealing algorithms are generally run on schedules shorter than those for which convergence on the global minimum is guaranteed and, as such, it is usual to maintain a record of the best-performing solution found during an optimisation, not only the final solution. While the cooling schedule required to guarantee convergence is infeasibly slow, simulated annealing generally works well even on much shorter schedules and as such, it is worth extending simulated annealing for multi-objective optimisation.

Simulated annealing concepts and terminology are presented more thoroughly in Chapter 2.

1.1.3 Multi-Objective Simulated Annealing

The motivation of this thesis, then, is to apply simulated annealing techniques to multi-objective optimisation problems. Bringing the advantages of simulated annealing to multi-objective optimisation holds the promise of relatively rapid convergence on problems which are very difficult to converge to with many of the existing algorithms.

1.2 Principal Contributions

A general overview of the aims of this work is as follows:

- The development of a measure of the quality of a multi-objective solution, suitable for use in a simulated annealer, such that no prior knowledge of the relative importance of the objectives is required for the search and optimisation process. This measure, utilising an archive estimating the Pareto front, does not use the performance metrics and is, as such, not susceptible to convergence difficulties for non-uniformly scaled problems. It additionally assigns equal value to all solutions on the Pareto front, which removes the bias found in many measures towards certain regions of the front.

- The development of a multi-objective simulated annealing algorithm which utilises this performance measure which can be shown, through empirical tests, to outperform an existing multi-objective simulated annealer and to perform at least comparably with a popular and well-regarded multi-objective evolutionary algorithm. Several additional contributions are made as part of this work, including a method for sampling from the attainment surface of a set of non-dominating solutions and a method for generating solution perturbations with desirable properties.

- The development of additional multi-objective simulated annealing algorithms utilising differing approaches to state construction and solution quality measurement. These additional algorithms use sets of solutions as their states and optimise these sets using different energy functions, utilising both dominating solutions, and dominated volumes.

- The empirical study of these simulated annealing techniques to ascertain the convergence properties of the various approaches and their relationship with existing evolutionary strategies.

- The investigation of test problem properties and the identification of problem traits which lend themselves to the application of simulated annealing techniques as opposed to existing evolutionary techniques.

1.3 Thesis Overview

Following this introductory section, the necessary background for the body of the thesis is presented in Chapter 2. The concepts and techniques currently used in existing multi-objective simulated annealing are introduced, together with a brief overview of evolutionary optimisation techniques which both inspire the techniques, and are used to provide performance comparisons, in the later parts of the thesis.

The first novel techniques are presented in Chapter 3, where an energy function for simulated annealing, based upon the relative dominance of a solution and a historical archive, is introduced. This energy function quantifies changes in quality of multi-objective solutions without encapsulating metric qualities, or assigning any preference information. A simulated annealer utilising this energy measure is compared to an existing multi-objective simulated annealer, and is shown to dramatically outperform it, and also to the frequently used NSGA-II genetic algorithm [Deb et al., 2002a], which it generally out-performs. The applications of this annealer to the optimisation of a mobile telecommunications network is described and results are presented.

Chapter 4 compares several novel approaches to multi-objective simulated annealing. Techniques similar to those used in the algorithm presented in Chapter 3 are extended to create a simulated annealer which maintains a set of solutions to be perturbed as the current state, instead of the traditional single solution. The relationship between these algorithms and existing evolutionary strategies is discussed. A third simulated annealer, which calculates the volume dominated by a set of solutions for use as the quality of a state, is also proposed. Comparisons are performed which demonstrate the relative worth of set and single solution states and the performance of annealers using both dominance and volumes is compared. The chapter ends with an investigation comparing greedy search and simulated annealers, along with the properties of problems on which it is preferable to use either technique. A new test problem is presented which can be trivially converted between each case.

Finally, the conclusion of this thesis discusses the novel techniques presented in the thesis, the

knowledge gained from the study of experimental results within and directions of future research as a consequence of this work.

1.4 Author Declaration

Work from a number of papers partially contributes to the content of this thesis. These papers, either published or submitted for publication during his period of research at the University of Exeter towards a PhD, are listed, together with the chapters they appear in, as follows:

- Chapter 3 includes work from the study "*Dominance Measures for Multi-Objective Simulated Annealing*" in proceedings of the 2004 Congress on Evolutionary Computation (CEC2004) by K.I. Smith. R.M. Everson and J.E. Fieldsend.

- Chapter 3 also includes work from the study "*Multi-Objective Simulated Annealing for Optimisation of CDMA Network Air-Interface*" for the 2004 Motorola S^3 Symposium by K.I. Smith, R.M. Everson, J.E. Fieldsend, R. Misra and C. Murphy.

- Chapter 3 also includes work from the study "*Dominance-Based Multi-Objective Simulated Annealing*" (submitted to IEEE Transactions on Evolutionary Computation, February 2005) by K.I. Smith, R.M. Everson, J.E. Fieldsend, C. Murphy and R. Misra.

- Chapter 4 also includes work from the study "*Simulated Annealing and Greedy Searches for Multi-objective Optimisation Problems*" (to be submitted to IEEE Transactions on Evolutionary Computation, November 2006) by K.I. Smith, R.M. Everson and J.E.Fieldsend.

Chapter 2

Background

This chapter introduces and discusses the issues relevant to the remaining chapters of the thesis, presenting the terminology, definitions and concepts involved in multi-objective optimisation and simulated annealing, and providing an overview of the literature. The basis of algorithmic performance comparison is also presented, discussing performance metrics and test problems used later.

2.1 Multi-Objective Optimisation

Multi-objective optimisation problems are those problems for which there exist multiple dependent variables (objectives) of a system which it is desirable to optimise. As these objectives are generally competing it is no longer possible, as it is for single objective optimisation problems, to locate a single solution to the problem which is wholly better than all other solutions; there generally exists a trade-off curve (for two-objective problems, a surface for three-objective problems *etc.*), which may be plotted in the objective-space, upon which those solutions lie for which there exists no wholly better solution. A more formal discussion of the concepts involved in the optimisation of multi-objective problems follows.

2.1.1 Dominance and Pareto Optimality

In multi-objective optimisation one attempts to simultaneously maximise or minimise D objectives, y_i, while satisfying J inequality constraints, $g_j \geq 0$, and K equality constraints $h_k = 0$, which are

functions of P variable parameters or decision variables, $\mathbf{x} = (x_1, x_2, \ldots, x_P)$:

$$y_i = f_i(\mathbf{x}), \qquad i = 1, \ldots, D. \tag{1}$$

While problems exist for which the decision variables are discrete, this thesis is concerned with problems for which each decision variable, x_i, is continuous, between a lower bound x_i^L and an upper bound x_i^H: $\mathbf{x} \in \mathbb{R}^P$.

Without loss of generality, it can be assumed that the objectives are to be minimised, as maximisation of y_i is equivalent to the minimisation of $1/y_i$ or $-y_i$ and that constraints are of the 'greater than or equal to' form (similarly, less than or equal to constraints may be multiplied by -1 for expression in this manner); as such the the multi-objective optimisation problem may be expressed as:

$$\begin{aligned} \text{Minimise} \quad & \mathbf{y}_i = \mathbf{f}(\mathbf{x}), \quad i = 1, \ldots, D; \\ \text{Subject to} \quad & g_j(\mathbf{x}) \geq 0, \quad j = 1, \ldots, J; \\ & h_k(\mathbf{x}) = 0, \quad k = 1, \ldots, K; \end{aligned} \tag{2}$$

The idea of dominance is generally used to compare two solutions $\mathbf{f}$ and $\mathbf{g}$. If $\mathbf{f}$ is no worse for all objectives than $\mathbf{g}$ and wholly better for at least one objective it is said that $\mathbf{f}$ *dominates* $\mathbf{g}$, written $\mathbf{f} \prec \mathbf{g}$. Thus $\mathbf{f} \prec \mathbf{g}$ iff:

$$\begin{aligned} f_i \leq g_i \quad & \forall i = 1, \ldots, D \quad \text{and} \\ f_i < g_i \quad & \text{for at least one } i. \end{aligned} \tag{3}$$

Although a slight abuse of notation, it is useful to extend dominance in *objective* space to *parameter* space; as such it is said that $\mathbf{a} \prec \mathbf{b}$ iff $\mathbf{f}(\mathbf{a}) \prec \mathbf{f}(\mathbf{b})$.

The dominates relationship is not a total order and two solutions are *mutually non-dominating* if neither dominates the other. A set F of points in objective space is said to be a non-dominating set if no element of the set dominates any other:

$$\mathbf{a} \not\prec \mathbf{b} \; \forall \; \mathbf{a}, \mathbf{b} \in F \tag{4}$$

A solution to 2 is said to be globally non-dominated, or Pareto-optimal, if no other feasible solution

dominates it, where a feasible solution is one which satisfies the constraints. The set of all Pareto-optimal solutions is known as the Pareto-optimal front, or the Pareto front, $\mathcal{P}$; solutions in the Pareto front represent the possible optimal trade-offs between competing objectives. A human operator can select a solution with a knowledge of the trade-offs involved once this set has been revealed. Heuristic procedures, such as the multiple objective evolutionary algorithms and the multi-objective simulated annealing algorithms discussed later, yield sets of mutually non-dominating solutions which will be only an approximation to the true Pareto front. Some care with terminology is therefore required, and in this thesis the set produced by such an algorithm is referred to as the estimated Pareto front, which is denoted F. These ideas are discussed further by Deb [2001].

Within the context of multi-objective optimisation, an algorithm is said to have *converged* when the set of solutions it produces is equal to the Pareto front. As it is often the case that the Pareto front is continuous and algorithms produce a finite set, and the precision of calculation within a computer system is generally insufficient for such results, it is, slightly loosely, said that an algorithm, of the set of solutions it has produced, have converged to the true Pareto front when they lie a very small distance apart (distances of 10^-3 on a possible range upwards of 10^2 are common later in this thesis).

2.2 Evolutionary Multi-Objective Optimisation

Prior to the introduction of evolutionary algorithms for multi-objective optimisation, many deterministic optimisation techniques were employed. Some of these techniques functioned by performing a deterministic search of parameter space using information about the gradients, obtained through the derivatives, of the objective functions; there exist, however, many problems for which this gradient information does not allow optimisation, either because the functions are discontinuous, or because they can not be differentiated. Other methods relying only upon the objective functions (and corresponding constraints) were employed, but these were typically slow, requiring an extensive search of parameter space. Both approaches suffered from a tendency to locate solutions which were locally, but not globally, optimal.

Evolutionary multi-objective optimisation covers the use of many types of heuristic optimisers inspired by the natural process of evolution. As in nature, a population of individuals (solutions to the problem) exist and, through a process of change and competition between these individuals, the quality of the population is advanced. The motivation for using these evolutionary algorithms is that

of the shortcomings of classical methods; that the search spaces for optimisation problems are usually too large for an exhaustive search and may not be suitable for optimisation through differentiation of the objective functions. Deb [2001] provides an extensive review of early non-evolutionary multi-objective optimisation techniques and the introduction of evolutionary algorithms for multi-objective as the state of the art.

2.2.1 Evolution Strategies and Genetic Algorithms

Two common approaches to evolutionary optimisation are genetic algorithms and evolution strategies, which are briefly discussed in general here; some popular multi-objective implementations are discussed later. Extensive reviews of multi-objective evolutionary optimisation are given by Deb [2001] and Coello Coello et al. [2002].

Evolution Strategies

An evolution strategy (ES) takes an initial population of solutions (potentially of size 1) and each generation perturbs some number of these solutions by some small amount. Some subset of these solutions is then passed as the population for the following generation; the manner in which these solutions are selected is dependent upon the type of ES, but generally the "fitter" solutions are carried forward into the next generation so that the overall fitness of the population increases over time.

In (μ, λ) ES and $(\mu + \lambda)$ ES, μ parent solutions are perturbed to create λ offspring, the offspring then compete for participation in the following generation based upon their fitness. In the more common $(\mu + \lambda)$ ES, the offspring also compete with the parents, while in (μ, λ) ES the parents do not participate in the following generations

A common form of ES is the $(1 + 1)$ ES, which is a $(\mu + \lambda)$ ES where $\mu = 1$ and $\lambda = 1$; a single solution is perturbed and this is then compared with the parent solution, with the fittest of these passing into the following generation at the expense of the less fit. $(1 + 1)$ evolution strategies are a simple example of *greedy* algorithms; the fitter solution is chosen over the less fit, meaning the algorithm always makes the locally optimal move.

Genetic Algorithms

Genetic algorithms (GAs) (see Goldberg [1989], Fogel [1998] for overviews) are an effective tool for uni-objective optimisation. GAs perform in a manner similar to evolutionary strategies, but are characterised by their different methods for generating new solutions, and for selecting the population for subsequent generations. It is also usual to binary-encode the decision vector in a GA, although they may be left in their real form, as in an ES.

Solutions may be generated through either mutation or crossover. Crossover occurs between two parents and involves the (random) recombination of their parameters, or sequences of their binary encoding to create offspring. Mutation usually occurs through the flipping of a random selection of the bits in a binary encoding, but can also occur through random perturbation of a real encoding. When a solution survives into a subsequent generation of a GA, it is considered to have undergone reproduction, as each solution in a generation is then a new individual, in common with natural evolution. The mutation operator is used to permit exploration of search space, while the crossover operator is intended to combine promising elements of solutions into a "child".

A genetic algorithm is considered to be "Elitist" if it maintains the best solutions located so far in the search, allowing these to participate in the generation of the new solutions, ensuring that the algorithm does not lose this information of good solutions.

The selection method used in the GAs discussed here is binary tournament selection; when solutions are to be selected for the population of the next generation, they are paired off randomly and the fitter of the two solutions survives into the next generation. A more extensive introduction to genetic algorithms is provided by Mitchell [1996].

2.2.2 Composite Functions

Several early attempts at multi-objective genetic algorithms involved the combination of objectives into a single fitness function, using a scheme such as:

$$E(\mathbf{x}) = \sum_{i=1}^{D} w_i f_i(\mathbf{x}). \tag{5}$$

Here the single fitness function, $E(\mathbf{x})$, is the combination of the D objectives, given weightings $w_1 \dots w_D$. Using this scheme, the $E(\mathbf{x})$ function will be minimised for some $\mathbf{x} \in \mathcal{P}$, making this

an attractive approach for locating solutions on $\mathcal{P}$. Several algorithms were developed around this, such as those proposed by Koski [1988], Jahn et al. [1991]. Some approaches, such as the Vector Evaluated Genetic Algorithm (VEGA), proposed by Schaffer [1985] were later noted to be equivalent to the optimisation of such a composite function [Richardson et al., 1989].

Using a composite fitness function constructed through combination of the objectives is appealing because the minimisation of $E(\mathbf{x})$ produces a solution, the image of which in objective space lies on the Pareto front $y \in \mathcal{P}$. Through varying the weights $w_1 \dots w_D$ it is possible to located different regions of the Pareto front. Das and Dennis [1997] demonstrate, however, that composite functions are undesirable, even when a large number of different weights are used to sample across the Pareto front. A proof is provided that for problems where the Pareto front is non-convex, an algorithm minimising weighted sums of the objectives will be unable to converge upon some regions of the Pareto front. As such, it is desirable to implement multi-objective optimisers which do not utilise composite functions in this manner. Despite the problems with composite functions and the desire to develop alternative methods, optimisation techniques utilising composite functions have been shown to perform well, especially on many-objective problems (those with a large number of objectives) [Hughes, 2005].

2.2.3 MOGA: An Early Non-Composite Algorithm

Fonseca and Fleming [1993] introduced a multi-objective genetic algorithm which does not sum the objectives to create a scalar fitness function, as in Equation 5, but which utilises pre-defined goals for the comparison of solutions. The algorithm requires an operator to define a goal for each objective, where this goal is a value below which the solutions should lie. When comparing solutions located by the GA, non-dominating solutions can be compared in this manner and those that meet the goals are considered superior to those which do not.

While not being subject to the problems with convergence characteristic to weighted-sum composite functions, this method did still require advanced knowledge of the search space, and the relative priorities of the objectives. The algorithms discussed subsequently in this section are those, more modern, algorithms which have avoided this requirement and, as such, provide both optimal solutions and knowledge of unknown search spaces.

Algorithm 1 Framework of Laumanns *et al.*

1: $t := 0$
2: $(A^0, B^0, p_e^0) := initialise()$
3: `while`$terminate(A^t, B^t, t) = false$`do`
4: $t := t + 1$
5: $A^t := truncate(update(A^{t-1}, B^{t-1}))$
6: $p_e^t := adapt(A^t, B^{t-1}, p_e^{t-1})$
7: $B^t := vary(sample(evaluate(A^t, B^{t-1}, p_e^t)))$
8: `end`

2.2.4 A generic MOEA framework

Many evolutionary algorithms developed within the last few years have been *elitist* algorithms. As introduced earlier for single-objective genetic algorithms, an elitist algorithm for multi-objective optimisation is one which maintains an estimate of the Pareto front out of the non-dominated solutions located so far in the search, allowing these to participate in the generation of the new solutions. This prevents information encapsulated within previously located 'good' solutions from being lost in future generations of solutions and may provide a mechanism for evaluating the quality of new solutions.

Laumanns et al. [2000] present a general framework model for multi-objective evolutionary algorithms with elitism. As shown in Algorithm 1 Laumanns *et al.* propose that elitist algorithms can be defined using the *initialize*, *evaluate*, *sample*, *adapt*, *vary*, *update*, *truncate* and *terminate* operators, with a 'normal' population, B, an archive of elite individuals, A, and an 'elitism intensity', p_e (an additional operator *iterate* is presented for concurrent algorithms but is not used in the 'general (elitist) MOEA' presented). The elitism intensity is used as a parameter which determines the probability of parent individuals being selected from A instead of B during the generation of an offspring population. The *evaluate* operator is used for fitness assignment and selection, the *sample* operator then samples from the individuals and the *vary* operator generates the offspring, through mutation or recombination/crossover. Management of the elite set occurs in the *update* and *truncate* operators; the individuals to be stored within A are selected with *update* and the archive size may then be bounded with the *truncate* operator. The framework is used in Algorithm 1 to present a general elitist multi-objective evolutionary algorithm. This provides a convenient context for the discussion of several state-of-the-art evolutionary algorithms. Of the algorithms in the literature, the PAES, SPEA, NSGA-II and SPEA-II algorithms are prominent as being both popular and effective

for multi-objective optimisation. While these algorithms may be considered representative, a larger review of the field is presented by both Deb [2001] and Coello Coello et al. [2002].

2.2.5 PAES

PAES, the Pareto Archived Evolution Strategy proposed by Knowles and Corne [1999, 2000] is a 1+1 evolution strategy performing local search, presented as being the 'simplest non-trivial algorithm capable of generating diverse solutions in the Pareto optimal set'. A key feature of the PAES algorithm is that an archive of non-dominated solutions is maintained not as a pool of candidate solutions for parentage of offspring, as in elitist algorithms, but is used for the calculation of solution quality when a candidate solution is not dominated. The archive's size is bounded and a gridding method, whereby the number of solutions lying within each grid region in objective space is calculated, is utilised to ensure this limit is not exceeded; when a new non-dominated solution is found and the archive is 'full', a solution from the most crowded region of objective space is removed to allow insertion of the new solution (unless the new solution itself is in the most dense region). This gridding scheme is also used as the acceptance criteria when neither the candidate nor the current solution are dominated by members of the archive; in this case, the solution from the least crowded region is selected. This is designed so as to promote search of the entire Pareto front and to prevent solutions clustering in a single region.

Within the framework shown in Algorithm 1, A is the non-dominated archive, B (the population) is a single solution, and p_e is fixed at 0 such that offspring are always generated from B and never A. The *truncate* and *update* operators maintain the archive: *update* inserts the new solution into the archive if it is non-dominated, and removes dominated solutions and *truncate* then applies the gridding scheme to reduce the size of the archive if necessary. The *sample* operator always selects the B^t as a parent and *vary* makes a perturbation, B', to the parent, with selection of $B^{t+1} = B'$ if B' is non-dominated relative to A and it occupies a less populated region of the grid than B^t, else $B^{t+1} = B^t$.

Despite its relative simplicity, PAES has been shown to perform well on many problems, although results later in this thesis demonstrate that it has difficulty converging to $\mathcal{P}$ for problems constructed with complex search spaces.

2.2.6 SPEA

The Strength Pareto Evolutionary Algorithm (SPEA) proposed by Zitzler and Thiele [1999] is a simple and effective genetic algorithm which ranks solutions purely on dominance. An external archive of previously located non-dominated solutions is maintained and a clustering method ensures that the archive does not grow larger than a predefined limit, while maintaining the diversity within the archive. Parents for reproduction are selected from the union of the previous offspring and the archive through binary tournament selection with replacement, where the quality of solutions is ranked by the proportion of solutions which they dominate or are dominated by. SPEA is an effective algorithm for multi-objective optimisation, being conceptually simple but has since been superceded by the NSGA-II and SPEA2 algorithms discussed later.

Fitting SPEA into Algorithm 1, the elitist archive is A and the population of offspring is B. The *truncate* and *update* operators maintain the archive through a clustering method, while *sample* and *vary* select parents and create offspring using the binary tournament selection based on the relative dominance of the solutions to the population or archive. In SPEA, a solution from B may only be selected in the binary tournament if it is paired against a solution from B also, as all solutions in A are ranked as superior. As such, the probability of selecting a parent from A is dependent upon the size of A and B, as identified by Laumanns et al. [2000], and is $p_e^t = 1 - (|B|/(|A| + |B|))^2$.

2.2.7 NSGA-II

Deb et al. [2000, 2002a] proposed the Nondominated Sorting Genetic Algorithm-II (NSGA-II), an elitist multi-objective genetic algorithm. Unlike SPEA and PAES, NSGA-II does not maintain an external archive of non-dominated solutions and elitism is maintained through the selection criteria. At each generation, offspring are generated through tournament selection using the current population as parents. The population for the following generation is then selected from the union of the parent and offspring populations of the current generation, using both the notions of dominance, and a measure of density of solutions in objective space. The combined population of parents and offspring is split into several 'fronts' each comprised of mutually non-dominating solutions; this is performed by selecting a non-dominated front from the set comprised of all the solutions in the combined population which are not dominated by any others, assigning that to the first front, F_1, assigning the non-dominated subset of the remaining solutions to F_2 and continuing in this manner

such that the combined population is split into multiple non-dominating fronts. Within each front, a measure of closeness is then applied which calculates the spacing of each solution to their nearest neighbours in objective space. The population for the following generation is selected with preference given primarily to more dominating fronts and secondarily through the crowding measure; a solution in a more dominating front will be chosen, but where solutions lie within the same front, the solution within the less crowded region will be selected. Solutions from this population are chosen, using binary tournament selection on the measures described earlier, to act as parents, and offspring are then generated from these parents. The offspring and parents are then combined to form a population for the following generation as the process iterates.

The NSGA-II algorithm does not fit neatly into the elitist framework in Algorithm 1 due to the absence of an elitist archive (and therefore of an elitism intensity). However it is widely used and regarded as the leading multi-objective genetic algorithm and is successful in locating Pareto fronts on a number of test problems.

2.2.8 SPEA2

Subsequent to the initial SPEA work, an updated version, SPEA2, was introduced by Zitzler et al. [2002] with an updated fitness assignment procedure which incorporates density calculation similar to the NSGA-II, together with a modified elitist archive, which is no longer purely elitist but which is of a fixed size, filled with dominated solutions when there is a shortage of non-dominated solutions. The operation of SPEA2 is very similar to that of SPEA although binary tournament selection is now performed upon the archive only (the population does not participate).

The fitness function used in the tournament contains density information similar to that of NSGA-II but, unlike NSGA-II, a single fitness is calculated from the sum of the strength of dominance relative to the union of the archive and the population and the density. In this case, a two-tiered approach of first comparing dominance and secondarily comparing density, as in NSGA-II, is not necessary as the scales of the two terms being summed are such that the ordering of solutions is similar to that of NSGA-II. SPEA2 has been shown to perform as well as NSGA-II on some test problems but NSGA-II remains the technique prevalent in the literature.

Fitting the SPEA2 into Algorithm 1, B is again the population of offspring and A is now no longer an archive purely of elite solutions, but contains 'filler' solutions from the offspring if necessary.

Parents are selected only from A, never B: $p_e^t = 1$. The *truncate* and *update* operators maintain the archive through a clustering method (where $|A|$ is greater than a predefined maximum) and through augmentation with solutions from B (when $|A|$ is too slow). The *sample* and *vary* operators select parents and create offspring using the binary tournament selection based on the relative dominance of the solutions to the population or archive, together with the density of other solutions nearby in objective space.

2.3 Scalar Simulated Annealing

Simulated annealing, introduced by Kirkpatrick et al. [1983] may be thought of as the computational analogue of slowly cooling a metal so that it adopts a low-energy, crystalline state. In physical annealing, at high temperatures particles are free to move around, whereas as the temperature is lowered they are increasingly confined to the crystal lattice due to the high energy cost of movement. When the temperature is cooled sufficiently slowly, the system can settle into the crystalline lattice, but if cooled too rapidly the atoms may become frozen into misaligned irregular domains, which have a higher energy.

The act of annealing a metal such that it adopts the state of minimum energy can be thought of as a minimising optimisation problem, and as such it is interesting to use a similar technique for computational minimisation. It is physically appealing to call the function of computational optimisation using simulated annealing to be minimised the *energy*, $E(\omega)$, of the state ω, and to introduce a parameter T, the computational temperature, which is lowered throughout the simulation according to an annealing schedule. At each T the SA algorithm aims to draw samples from the equilibrium distribution $\pi_T(\omega) \propto \exp\{-E(\omega)/T\}$. As $T \to 0$ more and more of the probability mass of π_T, is concentrated in the region of the global minimum of E, so eventually, assuming a sufficiently slow annealing schedule is used, any sample from π_T will almost surely lie at the minimum of E. It is usual in simulated annealing algorithms for the state, ω, to represent a single solution, $\mathbf{x}$, and so correspondingly for a perturbed state, ω', to represent a perturbation, $\mathbf{x}'$, to this solution but this is not a requirement and Chapter 4 utilises other state representations.

Sampling from the equilibrium distribution $\pi_T(\omega)$ at any particular temperature is usually achieved by Metropolis-Hastings sampling [Metropolis et al., 1953], which involves making proposals

Algorithm 2 Simulated annealing

Inputs:

$\{L_k\}_{k=1}^K$	*Sequence of epoch durations*
$\{T_k\}_{k=1}^K$	*Sequence of temperatures,* $T_{k+1} < T_k$
$\mathbf{x}$	*Initial feasible solution*

```
1:  for k := 1, ..., K
2:    for i := 1, ..., L_k
3:      x' := perturb(x)
4:      δE(ω, ω') := E(ω') − E(ω)
5:      u := rand(0, 1)
6:      if u < min(1, exp(−δE(ω, ω')/T_k))
7:        ω := ω'
8:      end
9:    end
10: end
```

ω' that are accepted with probability

$$A = \min\left(1, \exp\{-\delta E(\omega, \omega')/T\}\right) \tag{6}$$

where $\delta E(\omega, \omega')$ is the difference in energy between the current state and the proposed state:

$$\delta E(\omega, \omega') \equiv E(\omega') - E(\omega). \tag{7}$$

Intuitively, when T is high perturbations from ω to ω' which increase the energy are likely to be accepted (in addition to perturbations which decrease the energy, which are always accepted) and the samples can explore the state space. Subsequently, as T is reduced, only perturbations leading to small increases in E are accepted, so that only limited exploration is possible as the system settles on (hopefully) the global minimum. The algorithm is summarised in Algorithm 2: during each of K epochs, the computational temperature is fixed at T_k and L_k samples are drawn from π_{T_k} before the temperature is lowered in the next epoch. Each sample is a perturbation ('mutation' in the nomenclature of evolutionary algorithms) of the current state from a proposal density (line 3); the perturbed state ω' is accepted with probability given by (6), as shown in lines 4-8.

Convergence is guaranteed if and only if the cooling schedule is sufficiently gradual [Geman and Geman, 1984], but experience has shown SA to be a very effective optimisation technique even with relatively rapid cooling schedules [Ingber, 1993, Salamon et al., 2002, e.g.].

2.4 Multi-Objective Simulated Annealing

As mentioned earlier, most real-world optimisation problems are multi-objective and it is therefore desirable to extend the simulated annealing techniques for multi-objective optimisation. Several approaches to multi-objective simulated annealing have been presented, mostly based upon composite functions similar to those discussed earlier. Several approaches from the literature are discussed and compared, after presentation of some techniques common to these.

An attractive approach to multi-objective simulated annealing (MOSA), adopted by several investigators [Serafini, 1994, Ulungu et al., 1999, Czyżak and Jaszkiewicz, 1998, Nam and Park, 2000, Hapke et al., 2000, Suppapitnarm et al., 2000, Tuyttens et al., 2003], is to combine the objectives as a weighted sum, in the manner previously discussed in this chapter. Equation 5 is restated here :

$$E(\omega) = \sum_{i=1}^{D} w_i f_i(\mathbf{x}). \tag{8}$$

The composite objective is then used as the energy to be minimised in a scalar SA optimiser. An equivalent alternative [Engrand, 1997] is to sum $\log f_i(\mathbf{x})$, and others (e.g., Ulungu et al. [1999], Nam and Park [2000]) have investigated a number of non-linear and stochastic composite energies. All the techniques discussed here use a single solution $\mathbf{x}$ for the state ω.

It is clear from the previous discussion of composite energies that simulated annealing with a composite energy (8) will converge to points on the Pareto optimal front where the objectives have ratios given by w_i^{-1}, if such points exist. However, it is unclear how to choose the weights in advance, indeed, one of the principal advantages of multi-objective optimisation is that the relative importance of the objectives can be decided, subsequent to optimisation, with the (estimated) Pareto front on hand. Perhaps more importantly, parts of the front are inaccessible with fixed weights, as they are for evolutionary algorithms based on composite objectives (section 2.2.2) [Das and Dennis, 1997]. Recognising this, investigators have proposed a variety of schemes for adapting the w_i during the annealing process to encourage exploration along the front. See for example Jaszkiewicz [2001].

It is natural to keep an archive, F, of all the non-dominated solutions found so far, and this archive may be utilised to further exploration by periodically restarting the annealer from a randomly chosen element of F [Suppapitnarm et al., 2000]. Most composite algorithms also keep an archive for maintainance of 'good' solutions and many also use this to some degree for proposal acceptance

(where proposals added to the archive due to non-dominance are always accepted).

A proposal ω' in scalar SA is either better or worse than the current state ω depending on the sign of $\delta E(\omega, \omega')$; except for pathological problems the probability that $\delta E = 0$ is vanishingly small. In multi-objective SA, however, the solution comprising the proposed state, $\mathbf{x}'$, may dominate the current state's solution, $\mathbf{x}$, or $\mathbf{x}'$ may be dominated by $\mathbf{x}$ or they may be mutually non-dominating: in fact, the probability that a pair of randomly chosen points in D-dimensional space are mutually non-dominating is $1 - \left(\frac{1}{2}\right)^{D-1}$, so the mutually non-dominating case becomes increasingly common with more objectives. However, energies such as (5) may lead to ω' being accepted unconditionally ($\delta E < 0$) even though $\mathbf{x}' \not\prec \mathbf{x}$, because a large negative energy change from one objective may outweigh small positive changes on the other objectives. Each multi-objective simulated annealing algorithm which utilises a composite objective function must therefore deal with this behaviour in some manner and many approaches simply accept a solution if it is not dominated by the current one unconditionally. Some of the prominent multi-objective simulated annealing algorithms are discussed below.

2.4.1 Serafini [1994]

Serafini [1994] proposed an early approach to multi-objective simulated annealing. He considers several approaches for the construction of an energy function for calculating state change probabilities. As well as methods equivalent to a weighted sum, he considers using the difference in whichever objective represents the greatest difference between solutions: $\delta E = max f_i(\mathbf{x}') - f_i(\mathbf{x})$. He also considers products, rather than sums, of objective values and using the minimum difference between objectives, in various formulations and also considers composites of these functions. Results are presented on the combinatorial travelling salesman problem on which it appears to perform well, but no comparison to other optimisation techniques is given.

2.4.2 Czyżak and Jaszkiewicz [1998]

Another example of a multi-objective simulated annealer is given by Czyżak and Jaszkiewicz [1998]. The Pareto Simulated Annealing (PSA) uses several parallel annealing chains each optimising a composite energy. These chains operate independently of each other except that an additional system to adapt the weights of the objectives in a composite function as the algorithm progresses,

the weights of each parallel chain are set so as to move it away from other chains; at each use of the composite energy function the weights are chosen such that they direct the search away from the solutions in the other chains. PSA maintains an external archive of solutions which have not yet been dominated estimating $\mathcal{P}$, in a similar manner to many elitist GAs and ESs. Results are presented on the combinatorial knapsack problem and results are presented comparing PSA with Serafini's algorithm, demonstrating superior performance for PSA.

2.4.3 Ulungu et al. [1999]

Ulungu et al. [1999] investigate another multi-objective simulated annealing method. In this method, the acceptance criteria is again calculated using a weighted sum of the objectives and an archive set of possibly optimal solutions is maintained. While this method is very similar to Serafini's prior work and results are presented on a formulation of the knapsack problem, no comparison is made to either of the previously discussed algorithms so its relative performance is difficult to determine.

2.4.4 Suppapitnarm et al. [2000]

A good example of a composite objective function approach to multi-objective simulated annealing is given by Suppapitnarm et al. [2000]. Instead of weighting and summing the objectives to produce a composite energy difference for the acceptance criteria, this algorithm uses a multiplicative function with individual temperatures for each objective, each of which is adjusted independently by the algorithm, using a statistical method based upon the standard deviation of the objective functions for accepted solutions in order to remove the need for a predetermined annealing schedule. These multiplicative energy functions are equivalent to a weighted sum of logs of the objectives. This negates the need for *a priori* weighting of the objectives, and can be considered to function as a weighted composite sum approach with algorithmically controlled weightings. This concentrates search towards a single point on the Pareto front like other composite objective techniques and Suppapitnarm *et al.* employ a return-to-base scheme whereby the current solution is re-seeded with another solution from the non-dominated archive to promote a more uniform coverage. Suppapitnarm *et al.* promote exploration along the front by unconditionally accepting proposals that are not dominated by any member of F, otherwise using (6). Results are presented on a range of test and real problems and the algorithm is shown to perform comparably to other optimisation techniques.

2.4.5 Nam and Park [2000]

Of the multi-objective simulated annealing techniques in the current literature, perhaps the most promising is that of Nam and Park [2000] due to their use of dominance in state change probabilities. In this algorithm the relative dominance of the current and proposed solutions is tested and when the proposed solution dominates the current solution the proposal is accepted; this is analogous to the automatic acceptance of proposals with a lower state energy in single-objective simulated annealing. In addition to the common practise of employing a state change probability which guarantees acceptance of strictly superior perturbations, Nam & Park modify the acceptance rule so that proposals are accepted with probability given by (6) and (8) if they are dominated by $\mathbf{x}$, but unconditionally accepted if $\mathbf{x}' \prec \mathbf{x}$ or if $\mathbf{x}'$ and $\mathbf{x}$ are mutually non-dominating. This promotes exploration of the search space and escape from local fronts but as the dimensionality increases so does the proportion of all moves which are accepted unconditionally. This limits the behaviour of the algorithm to that of a random walk through the search space when dealing with problems with high dimensionality. When the proposed solution is dominated by the current solution, Nam & Park define several schemes for calculating the energy difference controlling acceptance similar to Serafini [1994]. Based on a small empirical study of two-objective problems they suggest that the best is the average difference in objective values. Nam & Park also employ 100 separate agents during optimisation, where each agent is an independent copy of the algorithm; this serves a similar function to Suppapitnarm *et al.*'s return-to-base approach to promoting diversity of the solutions located by the algorithm.

2.4.6 Summarising Multi-Objective Simulated Annealing

The multi-objective simulated annealing techniques discussed here present an evolution of Serafini's early techniques, rather than the development of radically different approaches. All the approaches utilise composite energy functions, some of these offer variations on the weighted sum, such as the sum of logged objectives, or the difference in a single objective. Of these, Nam & Park's approach seems to be the most promising and this will be used as the basis of comparisons later in the thesis.

It is clear that the assurance of a convergence proof can be provided for a multi-objective simulated annealer using a composite objective function and fixed weights (Equation 5), for problems which are convex [Das and Dennis, 1997], since they will converge to a single point; they are simply

traditional single-objective annealers which are not concerned with the generation of a set of solutions. Such annealers are fundamentally limited in their coverage of the Pareto front. On the other hand, it is difficult to see how proofs of convergence might be obtained with the heuristic modifications designed to promote exploration transversal to the front as an acceptance criteria of mutual non-dominance almost reduces the algorithm to a random walk. The use of variable weights also invalidates the proof. Given these difficulties, defining a multi-objective simulated annealer which utilises a composite objective function is undesirable.

2.5 Comparing Results

When assessing the relative performance of multiple-objective optimisation algorithms, there exist many methods for comparing the final archive estimating the Pareto front. Comparison of results from a single-objective optimiser is simple, as one simply prefers the lower value (for minimisation problems), as one does during optimisation itself. Similar issues, however, are faced when comparing multi-objective archives of non-dominated points as are faced within the optimisers themselves. It is desirable for the set resultant from multi-objective optimisation to both minimise the distance of the set to $\mathcal{P}$ and also to maximise the coverage of this set across $\mathcal{P}$. This section briefly considers the common methods which are used for comparing the results sets in Chapters 3 and 4. Since the results presented within this thesis are for test problems, for which $\mathcal{P}$ is known in advance, the performance measures discussed here are able to utilise this knowledge.

2.5.1 Distance to the Pareto front.

Perhaps the simplest method, in concept, for comparing two archives is to measure the minimum Euclidian distance of the image of each archive member from $\mathcal{P}$ (in objective space) and to average these across an archive to provide a measure of performance. The distance to $\mathcal{P}$ is shown in the left section of Figure 2, where the red dots represent the archive estimating the front, F, the thick black line represents the true Pareto front, $\mathcal{P}$, and the blue lines each represent the minimum distance of a member of F from $\mathcal{P}$; these are then averaged to give the distance metric. Where $d^2(x, \mathcal{P}) = \underset{y \in \mathcal{P}}{min} \sum_{i=1}^{D} [f_i(x) - y]^2$, the distance of the set F from $\mathcal{P}$, $\bar{d}(F, \mathcal{P})$ is defined in Equation 9:

$$\bar{d}(F, \mathcal{P}) = \underset{\mathbf{x} \in F}{\text{median}}[d(\mathbf{x}, \mathcal{P})] \tag{9}$$

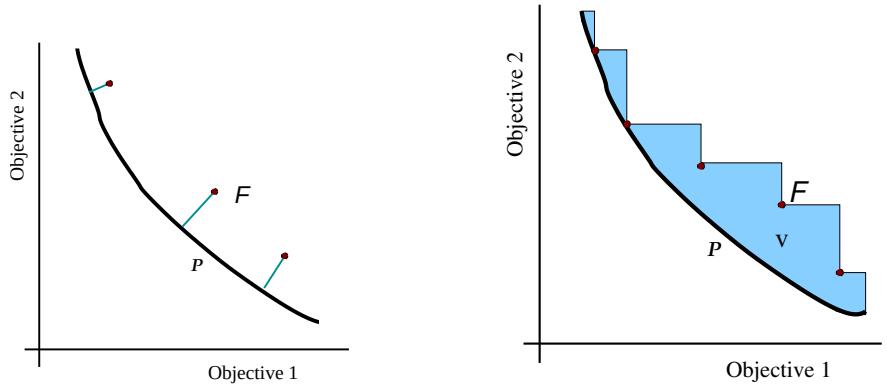

Figure 2: Archive quality metrics. **Left:** The distance to $\mathcal{P}$. **Right:** The volume metric. The black line represents the Pareto front and the red dots the archive. The blue lines represent the minimum distance of a point to $\mathcal{P}$ and the blue volume is the difference in dominated volume between $\mathcal{P}$ and the set.

The median is used here, instead of the mean, as an average as this is less susceptible to outliers.

This is a more accurate extension of the generational distance [Deb, 2001], where the distance calculated is not to the nearest point on $\mathcal{P}$, but to the nearest point in a sample of $\mathcal{P}$. Where the mathematical properties of $\mathcal{P}$ are not known, but a sample of points is available, the generational distance may be used instead. Minimising the value of this metric (achieving fronts which are very close to $\mathcal{P}$) is desirable. Calculation of this metric is entirely problem-dependent, so no general method is presented here.

This method requires advanced knowledge of $\mathcal{P}$, but this is not generally a barrier to use, as when testing algorithm performance, test problems with known properties will be used. The method does, however, provide no feedback on the coverage of a set and so cannot be used alone as a calculation of performance.

2.5.2 Dominated Volume

A useful metric for representing the coverage of a set is the Hypervolume [Deb, 2001], or covered space [Zitzler and Thiele, 1998]. This metric calculates the volume of objective space which is dominated by any member of the archive. Maximising the value of this metric (achieving fronts which dominate very large regions of search space) is desirable as this minimises the number of feasible solutions which dominate the set. In this thesis, a slightly modified version is used, which measures the

proportion of volume which is dominated by $\mathcal{P}$, but not by the archive; it is then desirable to minimise this metric (this converts the metric to give a sense of convergence to $\mathcal{P}$, rather than providing a value without a known target, this seems appropriate as this thesis is entirely concerned with minimisation problems). The volume difference from $\mathcal{P}$ is shown in the right section of Figure 2, where the red dots represent the archive estimating the front, F, the thick black line represents the true Pareto front, $\mathcal{P}$, and the blue area represents the difference in dominated volume between F and $\mathcal{P}$, $\mathcal{V}(\mathcal{P}, F)$. Since objective space may be unbounded, or its extent may not be known, it is necessary to bound objective space with a reference point, b to avoid infinite volumes. This reference point is naturally defined as the maximum co-ordinate of $\mathcal{P}$ in each dimension, where $\mathcal{P}$ is bounded; where $\mathcal{P}$ is unbounded a reference point must be placed artificially, such as the maximum co-ordinate for each objective across every set to be compared. With a slight abuse of notation this can be written as $\mathcal{V}(\mathcal{P}, F) = 1 - \frac{|\{x|\mathcal{P} \prec x \bigwedge x \prec F\}|}{|\{x|\mathcal{P} \prec x\}|}$ where $\mathcal{P} \prec x$ is given to mean that a member of $\mathcal{P}$ dominates x and where $x \prec F$ is given to mean that x dominates a member of F.

To make this precise, let H be the minimum axis-parallel hypercube in objective space which contains $\mathcal{P}$. Then $\mathcal{V}(\mathcal{P}, F)$ is the fraction of H which is dominated by $\mathcal{P}$ but not by F. Clearly this measure is zero when F covers the entire Pareto front and it approaches zero as F approaches P. Importantly however, an archive comprised of a few solutions clustered together on the true front will have a larger $\mathcal{V}(\mathcal{P}, F)$ than an archive of solutions well spread across the front and therefore dominating a larger fraction of objective space. This measure is straightforwardly calculated by Monte Carlo sampling (10^5 samples here) of H and counting the fraction of samples dominated exclusively by $\mathcal{P}$ and not F; see Fieldsend et al. [2003] for details. While the volume may also be calculated using methods such as those suggested by Fleischer [2003], these have traditionally been prohibitively computationally expensive for more than 2 objectives, for more than very small sets. Very recent work [Fonseca et al., 2006] may provide a feasible method for calculation in the future.

This metric is very susceptible to distortion through non-uniformly scaled objectives and it is therefore useful to normalise the results when there is a significant difference in the ranges of the objectives (this is not true for the test problems used in this thesis).

2.6 Test Problems

When constructing and comparing multi-objective optimisation algorithms, it is necessary to have available well-defined multi-objective optimisation problems with known properties, and without the large computational overhead usually associated with industrial problems. These problems are known as test problems and are constructed for the purpose of testing the properties of multi-objective optimisation algorithms.

The problems currently prevalent in the literature are the DTLZ problems [Deb et al., 2001, 2002c] and these are the problems used for algorithmic performance comparisons in Chapters 3 and 4 of this thesis. The DTLZ functions are successors to the popular ZD problems [Zitzler et al., 2000]. The DTLZ functions DTLZ1-DTLZ7 are used in the course of this thesis, usually in their 3-objective formulation; the DTLZ functions are appealing as $\mathcal{P}$ is known in advance and it is possible to calculate the distance of a solution from $\mathcal{P}$ mathematically in most cases (for calculation of distance to DTLZ6, the distance to one of a large sample of solutions lying in $\mathcal{P}$ is used, due to the difficulty of calculation of the minimum distance of a solution to the discontinuous front).

The true front for the DTLZ1 problem is the plane intersecting the axes at $(0, 0, 0.5)$, $(0, 0.5, 0)$ and $(0.5, 0, 0)$, and with $f_i \geq 0$. DTLZ1 has many locally optimal fronts on which an optimisation algorithm may become trapped (local fronts and their effects are studied in more detail in Chapter 4).

The DTLZ2, DTLZ3 and DTLZ4 problems are each the positive octant of a sphere centred on the origin with radius 1. DTLZ2 is a straightforward problem to solve, while DTLZ3 and DTLZ4 have added complexities to cause the problems to be harder to optimise. DTLZ3 has many locally optimal fronts upon which an optimiser may become trapped. DTLZ4 has a highly nonlinear mapping from parameter space to objective space, such that, of the preimage of $\mathcal{P}$, the majority of the parameter space corresponds to a corner of $\mathcal{P}$, while the surface of $\mathcal{P}$ is a very small proportion of the front (in parameter space).

The DTLZ5 problem is interesting, as it is constructed such that, in the 3-objective problem, $\mathcal{P}$ is a one-dimensional arc, rather than a two-dimensional surface.

In the DTLZ6 problem, $\mathcal{P}$ is split into four disconnected regions, or 'cushions', of objective space. It is anticipated that this discontinuity will cause difficulty for some optimisation algorithms.

Finally, the DTLZ7 problem uses constraints to shape the front, which is of the form of a plane

segment, and a line.

The definition of the DTLZ problems, for three objectives, are given in Table 1. Note that a couple of minor typographical errors in the description of DTLZ5 and DTLZ6 are rectified here, as the formulae published in Deb et al. [2001, 2002c] do not yield the Pareto fronts described.[1] The suggested configurations of the DTLZ problems are used in this thesis, with the numbers of parameters as given by Deb et al. [2001, 2002c]

[1] In equation (25) of Deb et al. [2002c] only θ_1 should be multiplied by $\pi/2$ when calculating $f_1, \ldots, f_M$. In equation (27) the calculation of $g(\mathbf{x}_M)$ is inconsistent with the results provided, meaning all f_3 values in the figure in Deb et al. [2001] are halved.

Table 1: Test problem definition of DTLZ1 – DTLZ7 of Deb et al for 3 objectives (using the suggested parameter sizes). (Definition of DTLZ5 corrected.)

Problem	Definition
DTLZ1	$f_1(\mathbf{x}) = \frac{1}{2}x_1x_2\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \frac{1}{2}x_1\left(1-x_2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \frac{1}{2}\left(1-x_1\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = 100(\lvert\mathbf{x}\rvert - 2 + \sum_{i=3}^{P}\left(x_i-0.5\right)^2 - \cos\left(20\pi\left(x_i-0.5\right)\right))$ $0 \le x_i \le 1$, for $i = 1, 2, \ldots, P$, $P = 7$
DTLZ2	$f_1(\mathbf{x}) = \cos\left(x_1\pi/2\right)\cos\left(x_2\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \cos\left(x_1\pi/2\right)\sin\left(x_2\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \sin\left(x_1\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = \sum_{i=3}^{P}\left(x_i-0.5\right)^2$ $0 \le x_i \le 1$, for $i = 1, 2, \ldots, P$, $P = 12$
DTLZ3	$f_1(\mathbf{x}) = \cos\left(x_1\pi/2\right)\cos\left(x_2\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \cos\left(x_1\pi/2\right)\sin\left(x_2\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \sin\left(x_1\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = 100(\lvert\mathbf{x}\rvert - 2 + \sum_{i=3}^{P}\left(x_i-0.5\right)^2 - \cos\left(20\pi\left(x_i-0.5\right)\right))$ $0 \le x_i \le 1$, for $i = 1, 2, \ldots, P$, $P = 12$
DTLZ4	$f_1(\mathbf{x}) = \cos\left(x_1^\alpha\pi/2\right)\cos\left(x_2^\alpha\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \cos\left(x_1^\alpha\pi/2\right)\sin\left(x_2^\alpha\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \sin\left(x_1^\alpha\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = \sum_{i=3}^{P}\left(x_i-0.5\right)^2$ $0 \le x_i \le 1$, for $i = 1, 2, \ldots, P$, $P = 12$
DTLZ5	$f_1(\mathbf{x}) = \cos\left(\theta_1\pi/2\right)\cos\left(\theta_2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_2(\mathbf{x}) = \cos\left(\theta_1\pi/2\right)\sin\left(\theta_2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $f_3(\mathbf{x}) = \sin\left(\theta_1\pi/2\right)\left(1+g\left(\mathbf{x}\right)\right)$ $g\left(\mathbf{x}\right) = \sum_{i=3}^{P}\left(x_i-0.5\right)^2$ $\theta_1 = x_1$ $\theta_2 = \frac{\pi}{4(1+g(\mathbf{x}))}\left(1+2g\left(\mathbf{x}\right)x_2\right)$ $0 \le x_i \le 1$, for $i = 1, 2, \ldots, P$, $P = 12$
DTLZ6	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}) = x_2$ $f_3(\mathbf{x}) = \left(1+g\left(\mathbf{x}\right)\right)h\left(f_1, f_2, f_3, g\right)$ $g\left(\mathbf{x}\right) = \frac{9}{P-2}\sum_{i=3}^{P}x_i$ $h\left(f_1, f_2, f_3, g\right) = 3 - \sum_{i=1}^{3}\left(\frac{f_i}{1+g}\left(1+\sin\left(3\pi f_i\right)\right)\right)$ $0 \le x_i \le 1$, for $i = 1, 2, \ldots, P$, $P = 22$
DTLZ7 s.t. s.t. s.t.	$f_1(\mathbf{x}) = \frac{1}{10}\sum_{i=1}^{10}x_i$ $f_2(\mathbf{x}) = \frac{1}{10}\sum_{i=11}^{20}x_i$ $f_3(\mathbf{x}) = \frac{1}{10}\sum_{i=21}^{30}x_i$ $g_1(\mathbf{x}) = f_3(\mathbf{x}) + 4f_1(\mathbf{x}) - 1 \ge 0$ $g_2(\mathbf{x}) = f_3(\mathbf{x}) + 4f_2(\mathbf{x}) - 1 \ge 0$ $g_3(\mathbf{x}) = 2f_3(\mathbf{x}) + f_1(\mathbf{x}) + f_2(\mathbf{x}) - 1 \ge 0$ $0 \le x_i \le 1$, for $i = 1, 2, \ldots, P$, $P = 30$

Chapter 3

A Dominance Based MOSA

3.1 Introduction

A popular and robust algorithm for solving single-objective optimisation problems (those in which the user cares only about a single dependant variable of the system) is simulated annealing (SA) [Kirkpatrick et al., 1983, Metropolis et al., 1953]. Geman and Geman [1984] provided a proof that simulated annealing, if annealed sufficiently slowly, converges to the global optimum, and although the required cooling rate is infeasibly slow for most practical purposes, simulated annealing often gives well converged results when run with a faster cooling schedule [Salamon et al., 2002, Ingber, 1993]. It is frequently the case in optimisation problems, however, that there are several objectives of the system which the user is interested in optimising simultaneously. Simulated annealing does not, in its usual formulation, provide a method for optimising more than a single objective. Simulated annealing has been adapted to multi-objective problems by combining the objectives into a single objective function [Engrand, 1997, Czyżak and Jaszkiewicz, 1998, Nam and Park, 2000, Hapke et al., 2000, Suppapitnarm et al., 2000]; however, as described in Chapter 2, these methods are limited (potentially severely) in their ability to fully explore the trade-off surface.

In this chapter a modified simulated annealing algorithm is proposed which maps the optimisation of multiple objectives to a single-objective optimisation using the true Pareto front, maintaining the convergence properties of the single-objective annealer while encouraging exploration of the

full Pareto front. A method of practical implementation is also described, using the available non-dominated data points from the current optimisation to overcome the limitation that the true Pareto front is unavailable for most real-world problems.

In this chapter, following on from the introductory material presented in Chapter 2, a dominance-based SA algorithm is described and, in section 3.3, methods are described for improving the quality of the optimisation energy measure when the available data points are few. Choosing an efficient scale for perturbations is an important component of scalar SA algorithms and the issue is further complicated in multi-objective algorithms because a perturbation may not only move the current state closer to or further from the Pareto front, but also transversally (i.e., across the front). In section 3.4 a method for setting the scale of perturbations and other run-time parameters is presented. Results showing that the algorithm converges on a range of standard test problems are given in section 3.5, and the algorithm is shown to compare very favourably with both the popular NSGA-II multi-objective genetic algorithm [Deb et al., 2000] and a multi-objective simulated annealer suggested by Nam and Park [2000]. Section 3.6 presents results demonstrating the simulated annealer's performance on the optimisation of the air interface of a Code Division Multiple Access (CDMA) network in the mobile telecommunications domain (the air interface of a mobile telecommunications network is the radio link used for communication between the mobile device and the network base station). Conclusions for this chapter are presented in section 3.7.

3.2 A Dominance Based Energy Function

To allow the construction of a multi-objective simulated annealer, a function is needed to transform a multi-objective solution to a single energy for a scalar annealer. In single objective optimisation problems the sign of the difference in energy between states, $\delta E(\omega, \omega')$ is used to determine whether the solution $\mathbf{x}'$, belonging to the proposed state ω' is a better, worse or (very rarely) equally good solution as the current solution $\mathbf{x}$ from the state ω. Likewise the dominance relation can be used to compare the relative merit of $\mathbf{x}'$ and $\mathbf{x}$ in multi-objective problems, but note that it gives essentially only three values of quality—better, worse, equal—in contrast to the energy difference in uni-objective problems which usually gives a continuum. These three values of difference provide an insufficient comparison between states for use as an energy function in simulated annealing, as it is not possible to evaluate an acceptance probability, but do value solutions without using metric

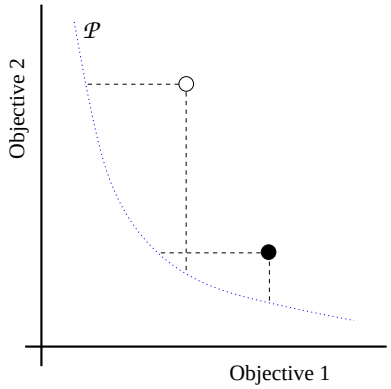

Figure 3: Energy from the area of the true Pareto front $\mathcal{P}$, represented as a dotted line, dominating a solution. Solutions are marked by circles and dashed lines indicate the regions of $\mathcal{P}$ dominating each one.

information from the objective functions.

If the true Pareto front $\mathcal{P}$ were available, the energy of ω could be simply defined as the measure of the front that dominates $\mathbf{f}(\mathbf{x})$. Let $\mathcal{P}_\mathbf{x}$ be the portion of $\mathcal{P}$ that dominates $\mathbf{f}(\mathbf{x})$:

$$\mathcal{P}_\mathbf{x} = \{\mathbf{y} \in \mathcal{P} \,|\, \mathbf{y} \prec \mathbf{f}(\mathbf{x})\}. \tag{10}$$

Then $E(\omega)$ is defined:

$$E(\omega) = \mu(\mathcal{P}_\mathbf{x}) \tag{11}$$

where μ is a measure defined on $\mathcal{P}$. It is principally interesting for heuristic optimisation to consider finite sets approximating $\mathcal{P}$ and so $\mu(\mathcal{P}_\mathbf{x})$ is taken to be simply the cardinality of $\mathcal{P}_\mathbf{x}$. If $\mathcal{P}$ is a continuous set, one can take μ to be the Lebesgue measure (informally, the length, area or volume for 2, 3 or 4 objectives); measures induced on $\mathcal{P}$ are further discussed in section 3.5.5. As illustrated in Figure 3, this energy has the properties desired: if $\mathbf{x} \in \mathcal{P}$ then $E(\mathbf{x}) = 0$, and solutions more distant from the front are, in general, dominated by a greater proportion of $\mathcal{P}$ and so have a higher energy; in Figure 3 the solution marked by an open circle has a greater energy than, and is thus considered inferior to, the one marked by a filled circle.

Clearly this formulation of an energy function does not rely on an *a priori* weighting of the

objectives and the assurances of convergence [Geman and Geman, 1984] for uni-objective SA continue to hold in this case. Since all solutions lying on the front have equal minimum energy, it is anticipated that a simulated annealer using this energy would, having reached the front, perform a random walk exploration of the front.

Fleischer [2003] has proposed an alternative measure of a non-dominated set, which may be loosely characterised as being based on the volume dominated by the set rather than the area of the dominating set. This is discussed later, in Chapter 4, as it is not possible to use this scheme with states comprised of a single solution, as is the focus of this chapter.

Unfortunately, the true Pareto front $\mathcal{P}$ is unavailable during the course of an optimisation and it is therefore proposed to use an energy defined in terms of the current estimate of the Pareto front, F, which is the set of mutually non-dominating solutions found thus far in the annealing. Denote by $\tilde{F}$ the union of the F, the current solution $\mathbf{x}$ and the proposed solution $\mathbf{x}'$, that is

$$\tilde{F} = F \cup \{\mathbf{x}\} \cup \{\mathbf{x}'\}. \tag{12}$$

Then, in a similar manner to (10), let $\tilde{F}_{\mathbf{x}}$ be the elements of $\tilde{F}$ that dominate $\mathbf{x}$:

$$\tilde{F}_{\mathbf{x}} = \{\mathbf{y} \in \tilde{F} \,|\, \mathbf{y} \prec \mathbf{x}\} \tag{13}$$

so that an energy difference is obtained between the current and proposed states of

$$\delta E(\omega, \omega') = \frac{1}{|\tilde{F}|} \left(|\tilde{F}_{\mathbf{x}'}| - |\tilde{F}_{\mathbf{x}}| \right). \tag{14}$$

Division by $|\tilde{F}|$ ensures that δE is always less than unity and provides some robustness against fluctuations in the number of solutions in F. If $\tilde{F}$ is a non-dominating set the energy difference between any two of its elements is zero. Note also that $\delta E(\omega, \omega') = -\delta E(\omega', \omega)$. The inclusion of the current solution and the proposal in $\tilde{F}$ means that $\delta E(\omega, \omega') < 0$ if $\mathbf{x}' \prec \mathbf{x}$, which ensures that proposals that correspond to strictly superior moves are always accepted and strictly inferior moves are always considered as such. Without the addition of these two points to the dominating set for the energy calculation, it is possible for a proposal which is in the same region of objective space as $\mathbf{x}$ to have an identical energy to $\mathbf{x}$, even when it is dominated by or dominates $\mathbf{x}$. Proposals

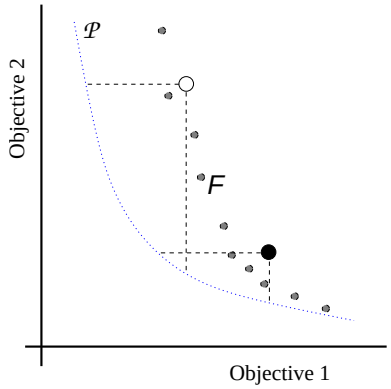

Figure 4: Energy from the proportion of the estimated Pareto front F dominating points dominating a solution. Elements of F are shown as small grey circles, solutions are shown as larger open or filled circles.

that are dominated by one or more members of the current archive are accepted with a probability depending upon the difference in the number of solutions in the archive that dominate $\mathbf{x}'$ and $\mathbf{x}$, given in Equation 15:

$$A = \min\left(1, \exp\{-\delta E(\omega, \omega')/T\}\right) \tag{15}$$

It should be emphasised that this simulated annealing acceptance probability does not depend upon metric information in objective space; there is no *a priori* weighting on the objectives and, as such, the acceptance probability and therefore the behaviour of the algorithm is unaffected by rescalings of the objectives.

A further benefit of this energy measure is that it encourages exploration of sparsely populated regions of the front. Imagine two proposals, each dominated by some solutions in F; for example, the solutions illustrated by the filled and unfilled circles in Figure 4. The solution that is dominated by fewer elements (the unfilled circle) has the lower energy and would therefore be more likely to be accepted as a proposal.

Defining the energy in this manner, unlike some proposed multi-objective enhancements to simulated annealing discussed in section 2.4, provides a single energy function encouraging both convergence to and coverage of the Pareto front without requiring other modifications to the single-objective simulated annealing algorithm (beyond the obvious storage of an archive of the estimated Pareto

front). In particular no additional rules are required for cases in which the current and proposed solutions are mutually non-dominating.

Convergence to the true Pareto front is no longer an immediate consequence of Geman & Geman's work [Geman and Geman, 1984], because the energy based on F is only an approximation to (11). However, Greening [1993] offers proof of convergence, albeit more slowly, even when the energy contains errors and an interesting area of future work would be to investigate the application of this work to MOSA and in section 3.5 empirical evidence of the convergence is offered.

An energy function based on (14) is straightforward to calculate; counting the number of elements of $\tilde{F}$ that dominate $\mathbf{x}$ and $\mathbf{x}'$ can be achieved in logarithmic time [Fieldsend et al., 2003, Jensen, 2003]. The proposed multi-objective algorithm closely follows the standard SA algorithm (Algorithm 2), the only addition that is necessary is to maintain an archive, F, of the current estimate of the Pareto front and to calculate the energy difference using (14). However, a detailed description of the algorithm is postponed until methods of increasing the empirical energy resolution have been discussed.

3.3 Increasing Energy Resolution

As mentioned earlier, the true Pareto-optimal front of solutions is, in general, unavailable to an optimisation algorithm. While use of the archive of the estimated Pareto front F provides an estimate of solution energy, when F is small the resolution in the energies can be very coarse. In fact, the difference in energy between two solutions is an integer multiple of $1/|\tilde{F}|$ between 0 and 1. Since the acceptance criterion (6) for new solutions is determined by the difference in energy $\delta E(\omega, \omega')$ between the current solution and the proposed solution, low resolution of the energies leads to a low resolution in acceptance probabilities. At low computational temperatures and with small archives it will become increasingly likely that this granularity will make it almost impossible for even slightly detrimental moves (i.e., moves that increase $E(\omega)$) to be made. This is undesirable as, at its most severe, this effect reduces the algorithm to behaviour similar to a greedy search optimiser, and prevents the exploratory behaviour provided by the acceptance of detrimental moves.

This problem is alleviated by the use of a larger set for energy calculations. There are a couple of straightforward, but ultimately inadequate, methods for artificially increasing the size of F which are now briefly discussed before description of a method using the attainment surface.

3.3.1 Conditional Removal of Dominated Points

A straightforward method for increasing the size of the archive is to not delete solutions known to be dominated if deleting them would reduce $|F|$ below some predefined minimum. However, the existence of old solutions in F may lead to desirable proposals (i.e., not non-dominated solutions) being rejected. In addition the old solutions may bias the search away from regions of the front that were previously well populated.

A further disadvantage of this method is that the retained solutions may be positioned such that they are dominated by the archive and possibly by the current point and the vast majority of proposals. In this case they serve to increase the resolution of the energy at the expense of the range. By contrast the interpolation method using the attainment surface that is proposed below insists that interpolating points are only weakly dominated by the archive.

3.3.2 Linear Interpolation

Another apparently suitable method of augmenting F is linear interpolation (in objective space) between the solutions in F. In this method, when the archive is smaller than some predefined size, new points in objective space are generated on the simplices defined by an element of F and its $D-1$ nearest neighbours in F. This overcomes the limitations of the previous method: Since new solutions are generated 'on' the current estimated Pareto front, the problems which could occur with using old, dominated elements of F in the energy calculations are avoided. The interpolated points generated can also be evenly distributed between the current estimated Pareto-optimal solutions, which is beneficial as it does not deter the algorithm from exploring any region of the estimated front which is not already densely populated. The principal disadvantage of this method is that proposals may be dominated by an interpolated point, but not by any of the real elements of F, meaning that the proposal may erroneously be disregarded.

3.3.3 Attainment Surface Sampling

Consideration of the previous two methods of augmenting the estimated Pareto front suggests that the augmenting points should have the following properties.

1. The augmenting points must be sufficiently close to the current estimation of the Pareto front

Algorithm 3 Sampling a point from the attainment surface

Inputs:
$\{L_d\}_{d=1}^{D}$ *Elements of F, sorted by increasing coordinate d*

1: `for` $i := 1, \ldots, D$ *Generate a random point,* $\mathbf{v}$
2: $v_i := \texttt{rand}(\min(L_i), \max(L_i))$
3: `end`
4: $d := \texttt{randint}(1, D)$ *Choose a dimension, d*
5: `for` $i := 1, \ldots, |F|$ *Find smallest v_d s.t. $\mathbf{v}$ is dominated by an element of F*
6: $\mathbf{u} := L_{d,i}$
7: $v_d := u_d$
8: `if` $F \prec \mathbf{v}$
9: `return` $\mathbf{v}$
10: `end`
11: `end`

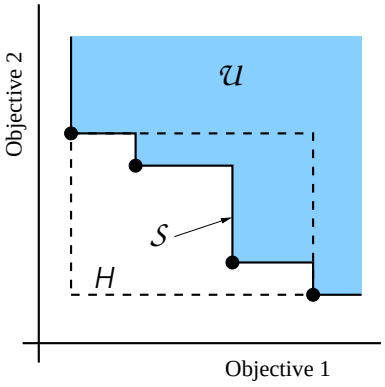

Figure 5: Attainment surface $\mathcal{S}_F$ is the boundary of the region, $\mathcal{U}$, dominated by the non-dominated set F, whose elements are marked as dots. Dashed lines denote H the minimum rectangle containing F.

that they can affect the energy of solutions generated near to the current estimated Pareto front.

2. They must be evenly distributed across the currently estimated Pareto front so as to not discourage the algorithm from accepting proposals in poorly populated regions of the front.

3. They must not dominate any proposal which is not dominated by any member of F, so that potential entrants to the archive are not discarded. A consequence of this is that they must all be dominated by at least one member of F.

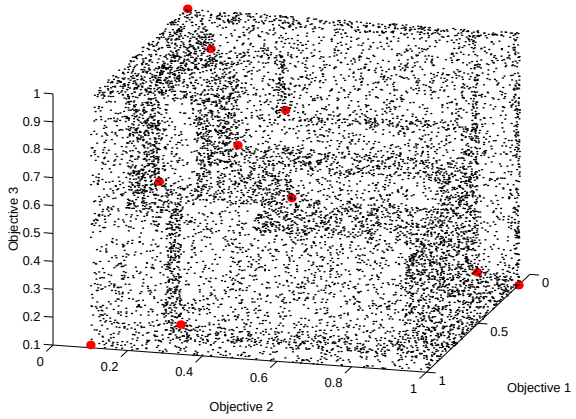

Figure 6: 10000 samples from the attainment surface for an archive of 10 points, which are marked with heavy dots.

The *attainment surface*, which has previously been used for estimated Pareto front visualisation [Zitzler, 1999] and is closely related to the attainment function [da Fonseca et al., 2001], is an interpolating surface between the elements of F that has the requisite properties. The attainment surface, $\mathcal{S}_F$, corresponding to F is a conservative interpolation of the elements of F so that every point of $\mathcal{S}_F$ is dominated by an element of F. The attainment surface for an F comprising four two-dimensional elements is sketched in Figure 5. More formally, the attainment surface is the boundary of the region in objective space which is dominated by elements of F. If $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$, it can be said that $\mathbf{u}$ *properly dominates* $\mathbf{v}$ (denoted $\mathbf{u} \lhd \mathbf{v}$) iff $u_i < v_i \ \forall i = 1, \ldots, D$. Then if

$$\mathcal{F} = \{\mathbf{y} \,|\, \mathbf{u} \prec \mathbf{y} \text{ for some } \mathbf{u} \in F\} \tag{16}$$

$$\mathcal{U} = \{\mathbf{y} \,|\, \mathbf{u} \lhd \mathbf{y} \text{ for some } \mathbf{u} \in F\} \tag{17}$$

the attainment surface is $\mathcal{S}_F = \mathcal{F} \setminus \mathcal{U} = \partial\mathcal{U}$.

To increase energy resolution F is interpolated with random samples uniformly distributed on $\mathcal{S}_F \cap H_F$, the attainment surface restricted to the minimum axis-parallel hyper-rectangle containing F (see Figure 5). From the definition of $\mathcal{S}_F$ it is apparent that interpolated points are dominated by an element of F, thus satisfying the third criterion. Uniform random sampling ensures that the second criterion is met, as is the first criterion because $\mathcal{S}_F$ interpolates F.

Algorithm 4 Multi-objective simulated annealing

Inputs:

$\{L_k\}_{k=1}^K$	*Sequence of epoch durations*
$\{T_k\}_{k=1}^K$	*Sequence of temperatures,* $T_{k+1} < T_k$
$\mathbf{x}$	*Initial feasible solution*

```
 1:  F := {x}                                              Initialise archive
 2:  for k := 1, ..., K
 3:    for i := 1, ..., L_k
 4:      x' := perturb(x)
 5:      if |F| < S                 If F is small construct attainment surface
 6:        S_F := interpolate(F)
 7:        F~ := S_F ∪ F ∪ {x} ∪ {x'}
 8:      else
 9:        F~ := F ∪ {x} ∪ {x'}
10:      end
11:      δE(x', x) := E(x') − E(x)          Energy difference based on F~
12:      u := rand(0, 1)
13:      if u < min(1, exp(−δE(x', x)/T_k))
14:        x := x'                                      Accept new current point
15:        if z ⊀ x  ∀z ∈ F   If x is not dominated by any element of F
16:          F := {z ∈ F | x ⊀ z}      Remove dominated points from F
17:          F := F ∪ x                                        Add x to F
18:        end
19:      end
20:    end
21:  end
```

Sampling from $\mathcal{S}_F$ may be performed using Algorithm 3, which works by sampling a point from a uniform distribution on the surface of H_F and then restricting one coordinate so that the point is dominated by an element of F. This is facilitated by the use of lists L_d, $d = 1, \ldots, D$ which comprise the elements of F sorted in increasing order of coordinate d. Determining whether an element of F dominates $\mathbf{v}$ on line 8 may be efficiently implemented using binary searches of the lists L_d, in which case the algorithm requires $\mathcal{O}(|F| \log(|F|))$ time for the generation of each sample. Figure 6 illustrates the sampled attainment surface for a set of ten 3-dimensional points; 10000 samples are shown for visualisation. In the experiments reported in section 3.5 F was augmented with 100 samples from $\mathcal{S}_F$ before calculating the energy of the proposal.

3.3.4 Multi-objective simulated annealing algorithm

Having discussed sampling from the attainment surface to increase the energy resolution, it is now possible to summarise the main points of the proposed multi-objective simulated annealing algorithm. As shown in Algorithm 4, the multi-objective algorithm differs from the uni-objective algorithm in that an archive F of non-dominated solutions found so far is maintained, and the energy difference between the proposed and current solution is calculated based on the current archive or its attainment surface.

The archive is initialised with the initial feasible point (line 1 of Algorithm 4). At each stage the current solution $\mathbf{x}$ is perturbed to form the proposed solution $\mathbf{x}'$. In the work reported here, in which the parameters $\mathbf{x}$ are continuous and real valued, each element of $\mathbf{x}$ is perturbed singly, drawing the perturbations from a Laplacian distribution centred on the current value.

If there are sufficiently many solutions in F, the augmented archive $\tilde{F}$ is constructed by adding $\mathbf{x}$ and $\mathbf{x}'$ (line 9) to F and the energy difference between $\mathbf{x}'$ and $\mathbf{x}$ is calculated using (14). If there are fewer than S solutions then additional samples are drawn from the attainment surface $\mathcal{S}_F$ using Algorithm 3 (line 6); the energy difference is then calculated based on the sampled attainment surface, $\mathbf{x}$ and $\mathbf{x}'$. In the work reported here, S is infinitely large so that F is always augmented with 100 samples from $\mathcal{S}_F$, as even when there are a large number of solutions in the archive of the estimated Pareto front it is worthwhile sampling from $\mathcal{S}_F$ since this samples evenly across the front, providing greater resolution in sparsely populated areas of the front.

If the proposal is accepted (line 14), the archive must be updated. If $\mathbf{x}$ is not dominated by any of the archival solutions, all archival solutions that are dominated by $\mathbf{x}$ are deleted from the archive (line 16) and $\mathbf{x}$ is added to the archive (line 17). Clearly F is always a non-dominated set, although note that $\mathbf{x}'$ may be dominated by members of F.

3.4 Run-time Algorithm Parameter Optimisation

The performance of this algorithm, in common with other simulated annealing systems, depends upon parameters for the initial temperature, the annealing schedule and the size of perturbations made to solutions when generating new proposals. Methods are described here which permit automatic setting of the initial temperature, and which adjust the scale of perturbations made to

maximise the quality of proposed solutions.

3.4.1 Annealing Schedule

If the initial computational temperature is set too high, all proposed solutions will be accepted, irrespective of their relative energies, and if set too low proposals with a higher energy than the current solution will never be accepted, transforming the algorithm into a greedy search. As a reasonable starting point, the initial temperature is set to achieve an initial acceptance rate of approximately 50% on exploratory proposals. This initial temperature, T_0, can be easily calculated by using a short 'burn-in' period during which all solutions are accepted and setting the temperature equal to the average positive change of energy divided by $\ln(2)$.

In the work reported here all epochs L_k are of equal length, $L_k = 100$ and the temperature is adjusted according to $T_k = \beta^k T_0$, where β is chosen so that T_k is 10^{-5} after two thirds of the evaluations are completed. This means that the final third of the search is effectively a greedy search.

3.4.2 Perturbation Scalings

For simplicity a proposal is generated from $\mathbf{x}$ by perturbing only one parameter or decision variable of $\mathbf{x}$. The parameter to be perturbed is chosen at random and perturbed with a random variable ϵ drawn from a Laplacian distribution, $p(\epsilon) \propto e^{-|\sigma\epsilon|}$, where the scaling factor σ controls the magnitude of the perturbation. The Laplacian distribution has tails that decay relatively slowly, thus increasing the probability of exploring regions distant from the current solutions [Yao et al., 1999].

Two sets of scaling factors are maintained, since the perturbations generating moves to a non-dominated proposal within a front (these can be considered *traversal* moves) may potentially be very different from those required to locate a front closer to $\mathcal{P}$, considered *location* moves. A scaling factor for each dimension of parameter space for each of the location perturbations and the traversal perturbations is maintained, and these are adjusted independently to increase the probability of such a move being generated. When perturbing a solution, it is chosen randomly with equal probability whether the location scaling set will be used, or the traversal scaling set. Statistics are kept on perturbations generating traversal and location moves; clearly these can be updated only after the proposal has been generated and evaluated so that the type of move is known. The scalings are adjusted throughout the optimisation, whenever a suitably large statistic set is available to reliably

calculate an appropriate scaling factor. These scalings are initially set large enough to sample from the entire feasible space.

Traversal Scaling

The traversal rescaling for a particular decision variable x_j is performed whenever approximately 50 traversal perturbations have been made to x_j since the last rescaling.

In order to ensure wide coverage of the front it is desirable to maximise the distance (in objective space) covered by the traversals. Generating traversals travelling a small distance will concentrate the estimated front around the point at which the current front was discovered, an undesirable effect which the larger perturbations aim to avoid.

The aim of the traversal scale calculations is to generate proposals on approximately the scale that has previously been successful in generating wide-ranging traversals. To achieve this, the perturbations are sorted by absolute size of perturbation in parameter space, and then trisected in order, giving three groups, one of the smallest third of perturbations, the largest third of perturbations, and the remaining perturbations. For each group the mean traversal size caused by the perturbations is calculated. The traversal size is measured as the Euclidean distance travelled in objective space when the current solution and the proposed solution are mutually non-dominating.

If a perturbation and the current solution are not mutually non-dominating, the traversal size is counted as being 0.

The traversal perturbation scaling for this decision variable is then set to the average perturbation of the group which generated the largest average traversal.

This heuristic is open to the criticism that it depends upon measuring distances in objective space while the relative weighting of the D objective functions is unknown. To alleviate this difficulty, however, the objectives may be renormalised during optimisation so the front has approximately the same extent in each objective. Unlike the generation of a composite function, where the relative weights of the objectives are crucial as they determine the final location of the optima, for these perturbation scales it is only necessary that they result in the coverage of large areas of objective space. It is emphasised that, of course, the use of metric information for setting the approximate scale of perturbations does not affect the dominance-based energy.

Location Scaling

Drawing from methods widely used in evolutionary algorithms (see Laumanns et al. [2001], Sbalzarini et al. [2001], Büche et al. [2003] for recent work in this area), it is desired that the adjustments to the scale of location perturbations keep the acceptance rate for $\mathbf{x}'$ that have a higher energy than $\mathbf{x}$ to approximately 1/3, so that exploratory proposals are made and accepted at all temperatures.

The location perturbation scaling is recalculated for each parameter for which 20 proposals having $\delta E(\omega, \omega') > 0$ have been generated, after which the count is reset. Location perturbation rescaling is omitted in two cases: firstly, when the archive of the estimated Pareto front F has fewer than 10 members. Secondly, when the combined size of F augmented by the samples from the attainment surface when multiplied by the temperature does not exceed 1. This is because the scalings are adjusted to attempt to keep the acceptance rate of exploratory moves approximately a third; when this value is too small, it becomes impossible to generate such a scaling, and in this case the scalings are kept at the most recent valid value.

Counting only moves generated from perturbations to a particular dimension of parameter space, the acceptance rate of exploratory moves α is the fraction of proposals to a greater energy which are accepted. If σ denotes the location perturbation scaling for a particular dimension, the new σ is set as:

$$\sigma := \begin{cases} \sigma(1 + 2(\alpha - 0.4)/0.6) & \text{if } \alpha > 0.4 \\ \sigma & \text{if } 0.3 \leq \alpha \leq 0.4 \\ \sigma/(1 + 2(0.3 - \alpha)/0.3) & \text{if } \alpha < 0.3 \end{cases} \tag{18}$$

This update works because, in general, smaller perturbations in parameter space are more likely to generate small changes in objective space, resulting in smaller changes in energy.

3.5 Experiments

The performance of this annealer is illustrated through its application to some well-known test functions from the literature, namely the DTLZ test functions of Deb et al. [2001, 2002c], and results are compared to the performance of the well established NSGA-II evolutionary algorithm [Deb et al., 2000] (using the PISA reference implementation [Bleuler et al., 2003]) and Nam & Park's multi-objective simulated annealer [Nam and Park, 2000] which are discussed in Chapter 2.

Table 2: Annealing Schedules

Problem	Run length	Time to $T_k = 10^{-5}$
DTLZ1	5000	3000
DTLZ2	1000	500
DTLZ3	15000	10000
DTLZ4	5000	3000
DTLZ5	1000	500
DTLZ6	5000	3000
DTLZ7	9000	6000

The benefit of using the DTLZ test functions is that the true Pareto front, $\mathcal{P}$, is known, so it can be calculated how close the estimated archive F is to $\mathcal{P}$, as well as making it possible to compare results from each algorithm. The DTLZ problems were discussed in Chapter 2 where the problem definitions are given in Table 1; in all the experiments here, $D = 3$ objectives are used.

In the work reported here all epochs L_k are of equal length for the annealers, $L_k = 100$ and the temperature is adjusted according to $T_k = \beta^k T_0$, where β is chosen so that T_k is 10^{-5} after approximately two thirds of the evaluations are completed; run lengths and the exact number of evaluations before T_k is 10^{-5} are given in Table 2. For MOSA, the parameter perturbations are controlled using the scheme described in Section 3.4.2, which are chosen to promote both exploration of the search space and convergence to $\mathcal{P}$. The perturbations for Nam & Park's annealer are performed using a scheme similar to that for MOSA but without the automatic rescaling feature novel to MOSA; the perturbation scalings are fixed at 0.1 (determined from a small empirical study which demonstrated that the optimal value for fixed scales was approximately 10% of search space). The parameters for the NSGA-II algorithm used were those suggested as the default values in the PISA [Bleuler et al., 2003] package[1]. A population size of 100 for NSGA-II was selected from the PISA suggested values and 100 simultaneous chains for the Nam & Park implementation were used.

The performance of the algorithms on each of the DTLZ test problems is first discussed, after which statistical results summarising the performance over 20 runs are presented. Each of the 20 runs was initialised with a different random seed; the use of 20 runs for each experiment ensures that

[1]The values for the PISA variator parameters are: individual_mutation_probability=1, individual_recombination_probability=1, variable_mutation_probability=1, variable_swap_probability=0.5, variable_recombination_probability=1, eta_mutation=15, eta_recombination=5. A binary tournament is used, and the SBX (simulated binary crossover) and the polynomial mutation operator are used, with the probability of an individual being affected by mutation or crossover being 1.

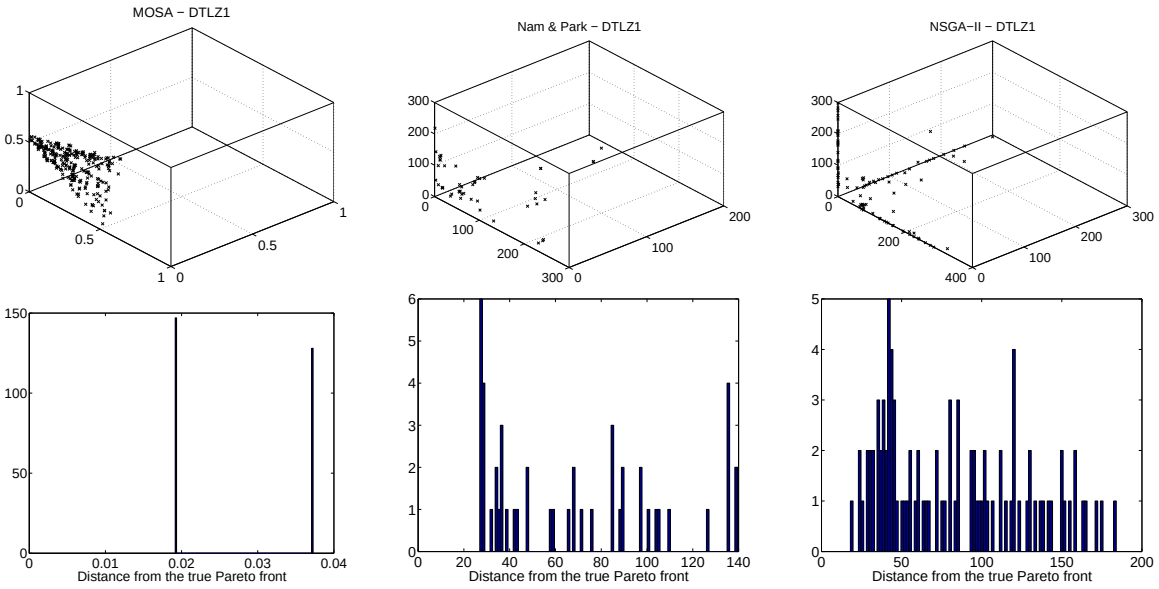

Figure 7: **Top:** Archives on test problem DTLZ1 after 5000 function evaluations for each of the three algorithms. **Bottom:** Histograms of the distances from the true Pareto front of the archive members (the 5% most distant have been omitted to aid visualisation in all six figures).

the results are representative of the algorithms' performance. The non-parametric Mann-Whitney rank-sum test (at the 0.05 level) is used to test for significant differences between the algorithms in the hypervolume and true front distance comparison measures.

3.5.1 DTLZ 1

Figure 7 shows views in objective space of the archive obtained from a single run of each of the algorithms on test problem DTLZ1 after 5000 objective evaluations, together with plots showing the distance of the members of each set to the true Pareto front. For each algorithm, the plotted results are those which have the median distance of solutions to the true front out of a series of 20 runs; this ensures that the results presented are representative of the series. The true front for DTLZ1 is the segment of the plane passing through 0.5 on each of the objective space coordinate axes, and it can be seen that the majority of solutions generated by MOSA lie very close to the front. This test problem has a large number ($\approx 11^5$) of local fronts which lie as planes parallel to and further from the origin than $\mathcal{P}$; the existence of these fronts is evident from the histogram of the distances

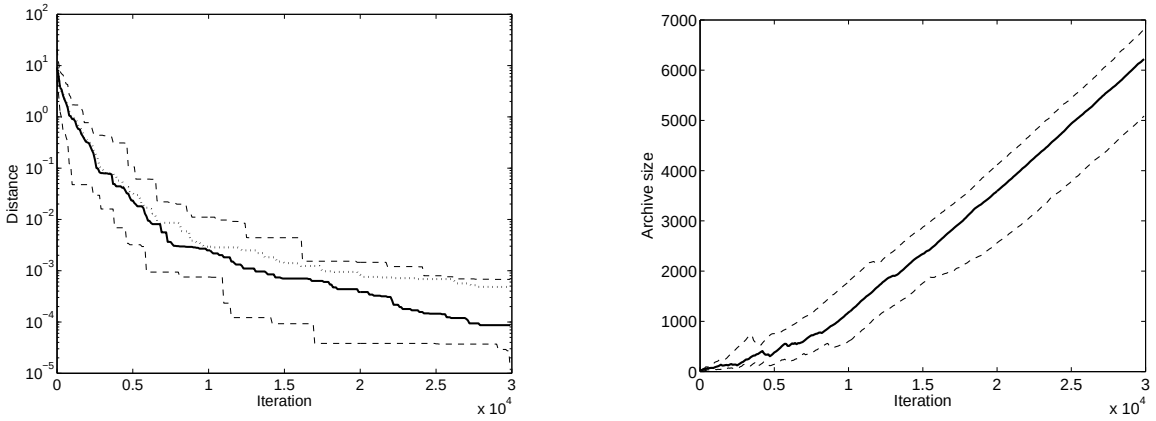

Figure 8: **Left:** Distance of current point, $\mathbf{x}$, and archive F from the true Pareto front, $\mathcal{P}$, versus iteration for DTLZ1. The dotted line shows median over 20 runs of distance of $\mathbf{x}$ from $\mathcal{P}$; dashed lines show maximum and minimum (over the 20 runs) distances at each iteration. The thick line shows the median (over 20 runs) of the median distance of archive members to $\mathcal{P}$. **Right:** Archive growth versus iteration. Thick line shows median (over 20 runs) archive size and dashed lines show maximum and minimum.

from $\mathcal{P}$ which shows solutions clustered at two distinct distances for MOSA and several for NSGA-II (this effect is less marked on the Nam & Park front, where the solutions are distributed more evenly across many fronts which are close in objective space). It seems likely that it is these local fronts which prevent Nam & Park's annealer and NSGA-II from converging on the true front, since in later problems without this feature the difference in performance between the three algorithms is, while still significant, much less extreme. Figure 16 provides, for each test problem, box plots comparing the average distance of the archive to the true front, the volume measure of the archive and the size of the archive (which is a fixed value for NSGA-II due to the constrained nature of the algorithm). For this DTLZ1 problem, the figure clearly illustrates that MOSA has not only converged to a set very close to the true front but that the front is also well covered as shown by the volume measure results; since the MOSA archive is unconstrained in size it has been able to generate a large archive close to, and covering well, the true front.

It was observed that the annealer on this problem converges to a local front, spreads across it until a perturbation 'breaks through' to a front closer to $\mathcal{P}$ after which the annealer explores the nearer local front, adding solutions on this front to the archive and removing solutions on the previous local front as they become dominated during the exploration. Figure 8 shows the median,

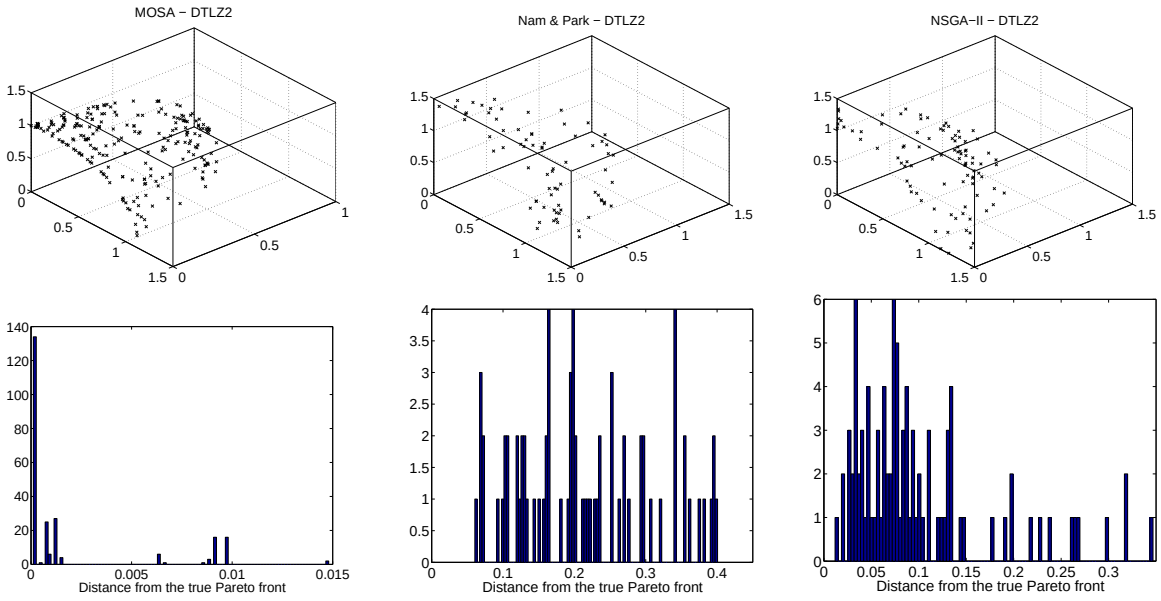

Figure 9: **Top:** Archives on test problem DTLZ2 after 1000 function evaluations. **Bottom:** Histograms of archive member distances from the true Pareto front (the 5% most distant have been omitted to aid visualisation).

maximum and minimum (over 20 runs) of the distance of the current point $\mathbf{x}$ to the true front $\mathcal{P}$ versus iteration, together with the median (over 20 runs) of the median distance of members of the archive F from $\mathcal{P}$ on a much longer set of runs. The presence of local fronts is apparent from the 'steps' in the median archive distance. The current solution clearly leads the archive, particularly at later iterations when the computational temperature is low and the search is effectively a greedy search.

3.5.2 DTLZ 2

Figure 9 presents the archive resulting from a representative run of the algorithms on problem DTLZ2 for 1000 function evaluations and a plot of the distances from the true front, which is the eighth of a spherical shell of radius 1, centred on the origin, lying in the positive octant. As the figure shows, the archive lies close to the optimal front for each of the algorithms, with MOSA significantly closer than the other algorithms.

It is worth remarking that this problem, and several others of the DTLZ suite without a plethora

of local fronts, can be successfully treated with a rapid cooling schedule, as used here. Due to the ease of convergence to the true front on this problem, it seems probable that any multi-objective optimiser will be able to produce a set of solutions close to the true front although the density and coverage may vary significantly, as is the case here. Figure 16 illustrates that, while all three algorithms have converged close to the true front, MOSA is significantly closer than NSGA-II or Nam & Parks annealer. The volume measure plot shows that the archive produced by MOSA also has a greater coverage/density of solutions; even after only 1000 evaluations, the archive size plot clearly illustrates that MOSA has already converged very close to the true front and is searching across the front improving the coverage and density.

While knowledge about the applicability of a short annealing schedule would not be initially available for typical real-world problems, it is anticipated that, for real-world problems, the annealer would be run with a very rapid annealing schedule initially to discover if the problem were searchable in this manner.

3.5.3 DTLZ 3

A striking example of the annealer's performance is provided in Figure 10, where its evaluation on DTLZ3 is shown for 15000 function evaluations. The Pareto front here is again an eighth of a spherical shell, preceded by multiple local fronts, of the same order as DTLZ1. The MOSA computational archive is converged to within 0.01 of the true front, as supported by the histogram of solution distances from the front in Figure 10. Consistent with the findings by Deb et al. [2001] NSGA-II had failed to converge [2] and Nam and Park's annealer yields performance similar to NSGA-II (as illustrated in Figure 16). Consistent with the previous problems, MOSA's archive is shown to be large, dense and well covering in Figure 16.

3.5.4 DTLZ 4

The true Pareto front for this problem is again an eighth of a spherical shell, but the solutions are unevenly distributed across it. Figure 11 shows the algorithms' archives after 5000 function evaluations, showing that solutions are concentrated close to the $f_1 - f_3$ and $f_1 - f_2$ planes together with a less dense covering of the shell between them for MOSA and Nam & Park's algorithm, while

[2] Deb *et al.* comment that in their experiments NSGA-II had still failed to converge after 50000 function evaluations

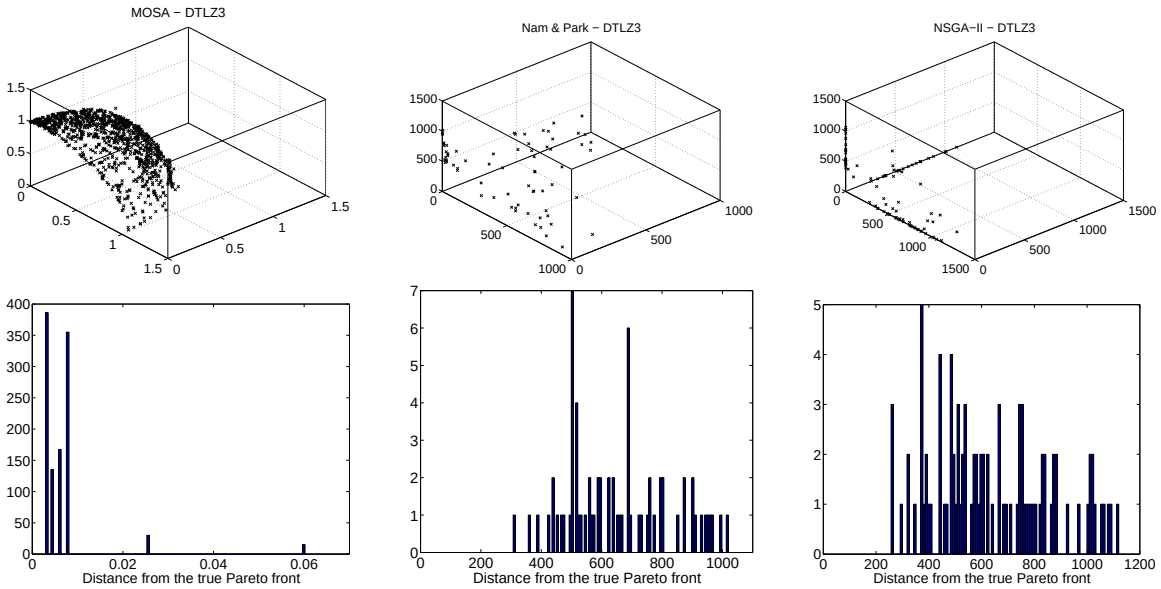

Figure 10: **Top:** Archives on test problem DTLZ3 after 15000 function evaluations. **Bottom:** Histograms of the distance from the true Pareto front of the archive members (the 5% most distant have been omitted to aid visualisation).

NSGA-II achieves an even coverage. Though the distribution of points across the front is more even with NSGA-II than MOSA, MOSA produced solutions which were far closer to the true front. Figure 16 shows that the solutions generated by MOSA have a much lower volume measure; although visually the solutions from the NSGA-II runs seem superior to MOSA's, the performance metrics suggest that MOSA has produced a better estimation of the true front. Deb et al. [2002c] observe that each run of NSGA-II in their experiments converged to a different part of the Pareto front; either to the f_1-f_2 plane, the f_3-f_1 plane, or distributed across the curved region of the front between these planes. The reason for the improved coverage of the PISA NSGA-II implementation is that the clustering close to the rims characteristic of the problem increases as solutions approach the true front. It is much more likely for solutions situated increasingly far from the true Pareto front to lie behind the central region of the front, although also to be dominated by the rims.

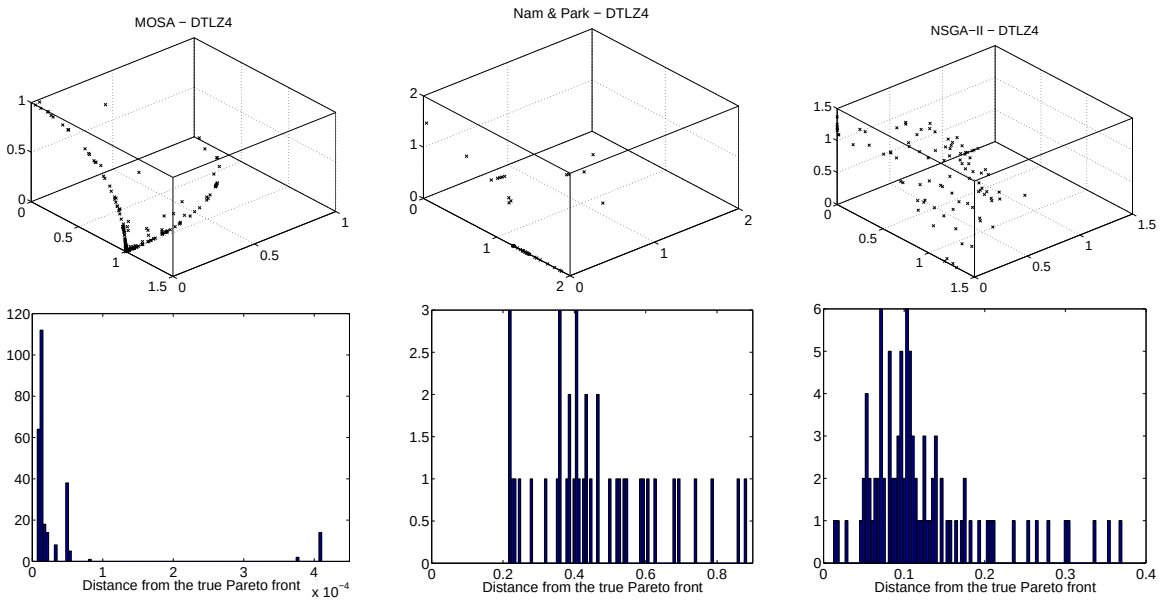

Figure 11: **Top:** Archives on test problem DTLZ4 after 5000 function evaluations. **Bottom:** Histograms of the distance from the true Pareto front of the archive members (the 5% most distant have been omitted to aid visualisation).

3.5.5 Density of solutions on the front

MOSA solutions on the front located by the annealer for problem DTLZ4 are close to the true Pareto front, but they are clearly inhomogeneously distributed across the front. Likewise, it is apparent from Figures 7, 9 and 10, for problems DTLZ1, DTLZ2 and DTLZ3, that the density of solutions is greater close to the $f_1 - f_2$ plane than distant from it. Here the reasons for this inhomogeneity are discussed in some detail. Other investigations into test problem properties have been performed by Okabe et al. [2002, 2004].

As alluded to in section 3.2, when $\mathbf{x}$ and $\mathbf{x}'$ both lie on or very close to $\mathcal{P}$ then $\delta E(\omega, \omega') = 0$ and all proposals lying on the front are accepted, so that the trajectory of the current solution is a random walk in *parameter* space. The density of solutions on this front in *objective* space is governed by the mapping of area or volume from parameter space to objective space. Assuming that the $f_i(\mathbf{x})$ are continuous in a neighbourhood of $\mathbf{x}$, the mapping is locally linear and is described by the D by

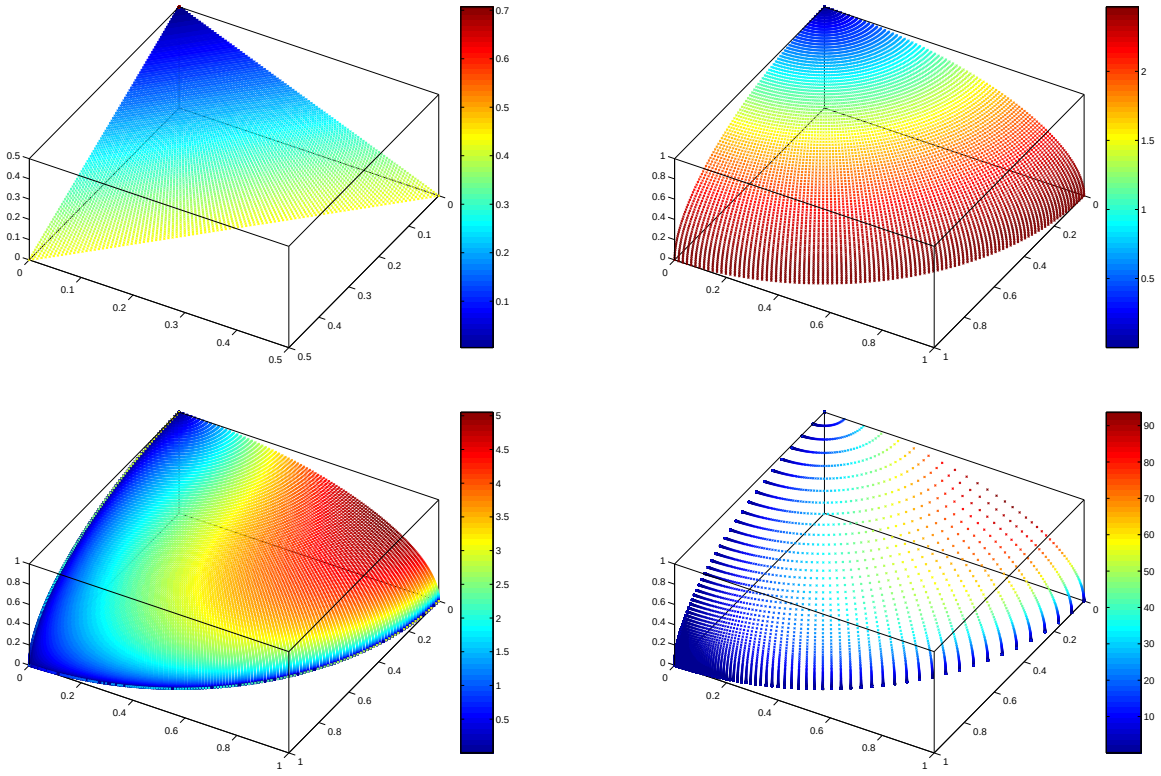

Figure 12: Magnification factors on the Pareto front. **Top left:** DTLZ1; **Top right:** DTLZ3; **Bottom left:** DTLZ4 with $\alpha = 2$; **Bottom right:** DTLZ4 with $\alpha = 10$. Colour indicates the local volume magnification factor from parameter space to objective space.

N Jacobian matrix of partial derivatives:[3]

$$\mathbf{J}_{ij}(\mathbf{x}) = \frac{\partial f_i}{\partial x_j}(\mathbf{x}). \tag{19}$$

It is useful to write $\mathbf{J}$ in terms of its singular value decomposition (SVD; see, for example, Golub and Loan [1983]):

$$\mathbf{J} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{20}$$

Here $\mathbf{U}$ is a D by D matrix whose orthonormal columns $\mathbf{u}_i$ $(i = 1, \ldots, D)$ form a local basis for

[3]In real problems the Jacobian matrix may be estimated by finite differences or computer-aided differentiation packages, e.g. Berz et al. [1996].

objective space at $\mathbf{f}(\mathbf{x})$. Likewise, the D columns of $\mathbf{V} \in \mathbb{R}^{P\times D}$, denoted $\mathbf{v}_i$, $(i = 1, \ldots, D)$ are orthonormal P-dimensional vectors forming a local basis for the D-dimensional subspace of parameter space that locally maps to objective space. The matrix $\mathbf{\Sigma} \in \mathbb{R}^{D\times D}$ is diagonal, whose diagonal elements $s_i \geq 0$ are known as singular values and are conventionally listed in descending order so that $s_1 \geq s_2 \geq \ldots s_D \geq 0$. The singular value s_i quantifies the magnification of a perturbation in direction $\mathbf{v}_i$ in parameter space: thus a small perturbation about $\mathbf{x}$ of $\epsilon\mathbf{v}_i$ in parameter space yields a change in objective space from $\mathbf{f}(\mathbf{x})$ to $\mathbf{f}(\mathbf{x}) + \epsilon s_i \mathbf{u}_i$.

If $\mathbf{x}$ lies on the Pareto front no parameter space perturbation can result in a change in objectives normal to the front, implying that one of the singular values is zero and the rank of $\mathbf{J}$ is at most $(D-1)$. Assuming for simplicity that the Pareto front is $(D-1)$-dimensional, the direction normal to the front corresponds to $\mathbf{u}_D$ and $\mathbf{v}_D$ in objective and parameter spaces respectively, and $s_D = 0$. Perturbations lying in the span of $\mathbf{v}_1, \ldots, \mathbf{v}_{D-1}$ result in traversal movements along the front and the (infinitesimal) volume in parameter space ν_p lying in span$(\mathbf{v}_1, \ldots, \mathbf{v}_{D-1})$ is magnified to volume

$$\nu_o = \nu_p \prod_{i=1}^{D-1} s_i. \tag{21}$$

on the Pareto front.

These ideas are illustrated in Figure 12, which shows the volume magnification factor on the front for DTLZ1, DTLZ3 and DTLZ4. These were calculated by evaluating the Jacobian matrix at a large number of points in parameter space using a symbolic algebra package and then numerically finding the singular values. Comparison with Figures 7 and 10 for DTLZ1 and DTLZ3 makes it apparent that the magnification factors correspond to the density of solutions generated by the simulated annealer. If $\mathcal{X}_\mathcal{P} = \mathbf{f}^{-1}(\mathcal{P})$ is the $(D-1)$-dimensional manifold in parameter space that maps to the Pareto front, then this may be understood in terms of the annealer performing a random walk on $\mathcal{X}_\mathcal{P}$ which it covers fairly uniformly, producing a high density of solutions in objective space where the magnification factor is low, but a low density of solutions where the magnification factor is high because here solutions in parameter space are spread more thinly in objective space.

The bottom panels of Figure 12 show the local volume magnification factors for DTLZ4, but with $\alpha = 2$ and $\alpha = 10$, rather than $\alpha = 100$ as recommended by Deb et al. [2001, 2002c]. As the figure indicates, the magnification factor at points on the front even for $\alpha = 10$ is almost two orders of magnitude greater than the magnification factors for DTLZ1 and DTLZ3; when $\alpha = 100$ the

pattern of magnification factors is similar but the range of magnifications is too great for sensible visualisation. The magnification is least close to the $f_1 - f_2$ and $f_1 - f_3$ planes, corresponding precisely to the regions in which plenty of solutions are located by the annealer (Figure 11) and greatest on the section of the front close to the $f_2 - f_3$ plane where few solutions are located. As such, it can be inferred that the annealer is locating and exploring $\mathcal{X}_\mathcal{P}$ in this case, but few solutions are seen on parts of the front because the magnification factors are extremely high.

These deliberations lead to further consideration of what is an appropriate natural measure on the Pareto front. In this formulation of a multi-objective simulated annealer an approximation to the Lebesgue measure is used, namely the number of solutions in the archive, to evaluate the energy of a solution (11). However, this measure is defined in objective space and it might be argued that a more natural measure in objective space is the one induced by Lebesgue measure on $\mathcal{X}_\mathcal{P}$. In fact, as these experiments show, once the vicinity of the Pareto front has been located it is (approximately) this induced measure that governs the density of solutions located. One may envisage that the singular value decomposition of $\mathbf{J}$ may be used to counteract the inhomogeneity produced in objective space by the magnification factor by biasing the perturbations along the singular vectors $\mathbf{v}_i$ associated with large singular values s_i. Another method for encouraging diversity in the sets is discussed in Chapter 4, and further study on this topic would be interesting for future research.

3.5.6 DTLZ 5

Figure 13 shows the archives generated by the algorithms after 1000 function evaluations on test problem DTLZ5 for which the front is a one-dimensional curve rather than a full two-dimensional surface. As the distance plots show, the annealer has successfully located the one-dimensional front while the other two algorithms generate sets which reside some distance behind this front; Deb et al. [2002c] also report that NSGA-II had not fully located the curve and yields a surface a little above the curve even after 20000 function evaluations in their experiments. Figure 16 shows that there is very little space dominated by the true front but not by the MOSA archive; the true front is almost completely covered by the archive. This is the only test problem in which MOSA's archive does not grow larger (in the allowed iteration count) than NSGA-II's (enforced) set of 100 results; this is not especially significant however, as the NSGA-II set is significantly less well converged than MOSA's archive.

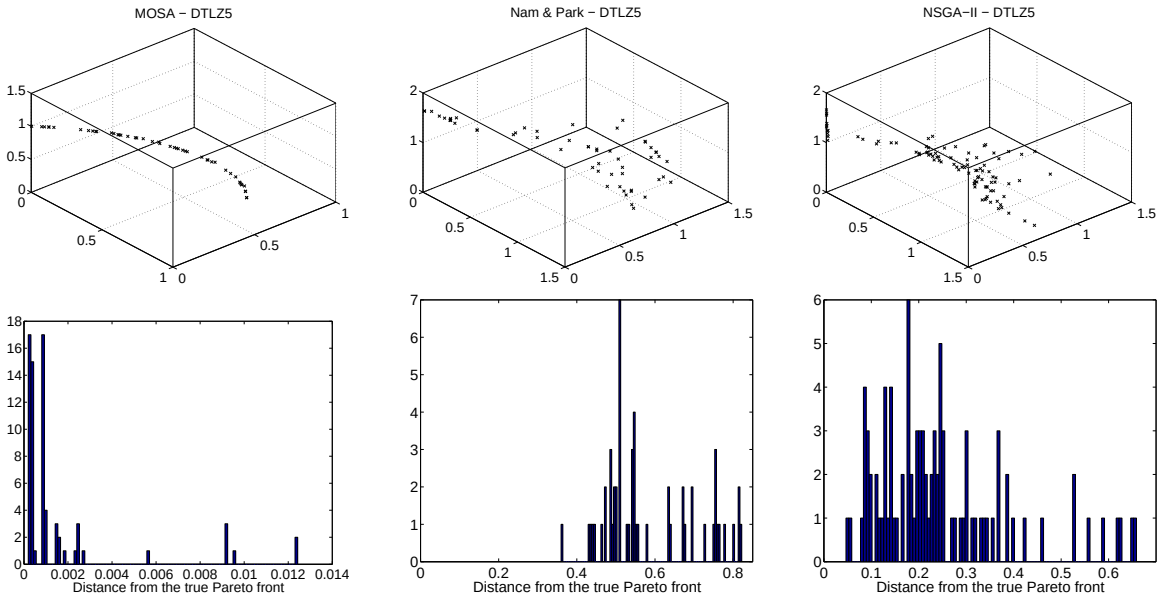

Figure 13: **Top:** Archives on test problem DTLZ5 after 1000 function evaluations. **Bottom:** Histograms of the distance from the true Pareto front of the archive members (the 5% most distant have been omitted to aid visualisation).

3.5.7 DTLZ 6

The front for DTLZ6 consists of four disjoint components.[4] As Figure 14 shows, the annealer is able to successfully locate each of these components during a single run. NSGA-II is able to generate solutions close to each front and Nam & Park's annealer does not converge in the allowed number of evaluations. Figure 16 shows that, again, MOSA's coverage of the front, as well as the distance from the true front, dominates almost all the feasible search space. During optimisation (and once the archive is close to the true Pareto front) it is observed that the current solution $\mathbf{x}$ of MOSA explores one component of the front for a few proposals before 'jumping' to another component. If the regions of parameter space corresponding to each of the components of the front were widely separated then it might be considerably more difficult for the annealer to simultaneously locate all components.

[4]The formula given in Deb et al. [2001, 2002c] is used; the figures in these publications appear to have been generated with the f_3 objective scaled by a factor of 2.

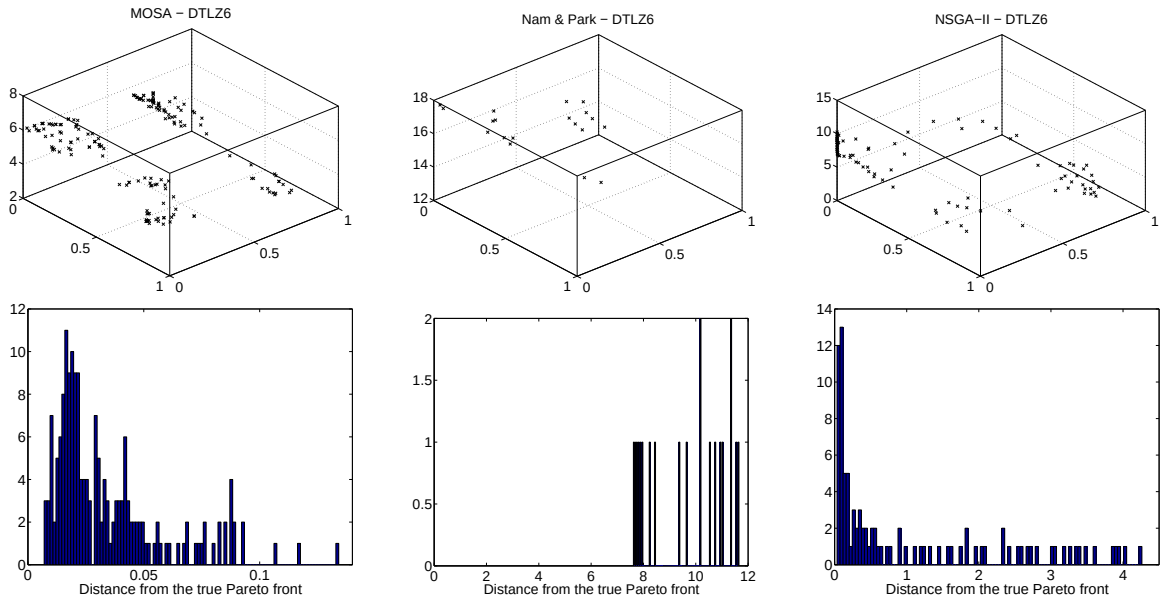

Figure 14: **Top:** Archives on test problem DTLZ6 after 5000 function evaluations for each of the three algorithms. **Bottom:** Histograms of the distance from the true Pareto front of the archive members (the 5% most distant have been omitted in each of the 6 figures to aid visualisation).

3.5.8 DTLZ 7

The DTLZ7 test problem is constructed using multiple constraint surfaces to yield a Pareto front consisting of a triangular planar section and a line segment. Figure 15 shows the algorithm archives after 9000 function evaluations. The particular way in which DTLZ7 is constructed means that a perturbation of a single parameter of a solution lying on the front makes the perturbed parameter vector infeasible because it violates one of the constraints. The schemes described in section 3.4.2, for adjusting the perturbation scalings rely on perturbing a single parameter at a time in order to keep track of the effect of the perturbation. However, this renders them ineffective for this problem: a single solution on the front is rapidly located, but the annealer is unable to explore the front because all perturbations result in infeasible proposals. For this reason the archive shown in Figure 15 was generated by perturbing a randomly chosen number of parameters for each proposal; for simplicity the perturbation scales were kept constant at 0.1 of the feasible region throughout the optimisation. While more efficient perturbation schemes could probably be devised, the figure shows that the annealer is reasonably successful in locating the central portion of the front, although the

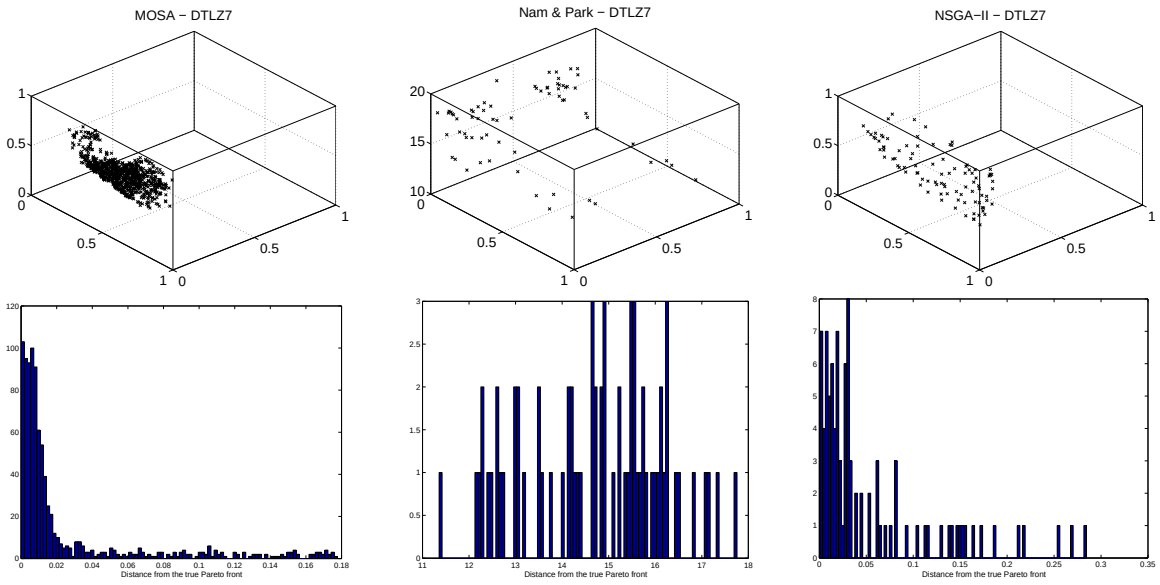

Figure 15: **Top:** Archives on test problem DTLZ7 after 9000 function evaluations for each of the three algorithms. **Bottom:** Histograms of the distance from the true Pareto front of the archive members (the 5% most distant have been omitted to aid visualisation in all 6 figures).

extremities of the front have not been explored and there remain some extraneous solutions close to constraint surfaces bounding the front, but still quite distant from $\mathcal{P}$ itself. The single parameter perturbation scheme used in this implementation of Nam & Park's annealer was also modified to perform the same multiple point perturbations as MOSA. NSGA-II, the PISA implementation of which already used a (more advanced) multiple parameter perturbation, did not need to be modified for this problem. Figures 15 and 16 show that, while MOSA has again converged well, and generates the solutions closest to, the true front, NSGA-II demonstrates the best coverage of solutions over the front towards the extremes of the constraints. It should be noted that the need to adapt to a multiple parameter perturbation scheme will be present for all algorithms which employ a specialised single parameter perturbation scheme (and conversely, problems can be constructed that would prevent a multiple parameter perturbation scheme from converging to the true front).

3.5.9 Statistical performance measures

Unlike single objective problems, solutions to multi-objective optimisation problems can be assessed in several different ways. Therefore, in order to quantify the convergence of the algorithms, two distinct properties are measured. Firstly, the average distance of the archived solutions discovered from the true front is calculated to ascertain how close, on average, solutions found are to the true front. Rather than using the root mean square distance which is susceptible to outliers, here the median distance of solutions in the archive is used, as discussed in Chapter 2:

$$\bar{d}(F, \mathcal{P}) = \underset{\mathbf{x} \in F}{\text{median}}[d(\mathbf{x}, \mathcal{P})] \tag{22}$$

where $d(\mathbf{x}, \mathcal{P})$ is the minimum Euclidean distance between $\mathbf{x}$ and the true front $\mathcal{P}$. Clearly, this measure depends on the relative scaling of the objective functions, however, it yields a fair comparison here because the objectives for the DTLZ test functions have similar ranges.

Secondly, since the algorithm is concerned with finding solutions spread across the true Pareto front, a variant of the volume $\mathcal{V}$ measure [Fieldsend et al., 2003] which is conceptually similar to the performance measure used by Laumanns et al. [2000] is used. The idea is to calculate the amount of objective space that is dominated by the true front, but not by the calculated archive. The $\mathcal{V}(\mathcal{P}, F)$ measure is discussed further in Chapter 2.

Figure 16 shows box plots over 20 runs, from different randomly-selected initial solutions, of the median Euclidean distance, $\bar{d}(F, \mathcal{P})$, fractional volume measures and archive size of the results for each algorithm on each test problem.

The distance of $\mathcal{P}$ to the objective space origin is $\mathcal{O}(1)$ for all of these problems, so it can be seen from Figure 16 that the annealer is able to converge very close to the front for all seven problems. In fact, MOSA is significantly closer to the front than both NSGA-II and Nam & Park's annealer. NSGA-II was able to converge to a set near to the true front for five of the problems (with two of those being very near) and Nam & Park's annealer was able to generate an archive near the true front on one of the problems.

The middle row of Figure 16 shows $\mathcal{V}(\mathcal{P}, F)$, the fractional volume dominated by $\mathcal{P}$ and not by F. As the figure indicates the annealer both converges well to $\mathcal{P}$ and also covers it reasonably well for all the problems. MOSA dominates significantly more volume than NSGA-II for 6 of the

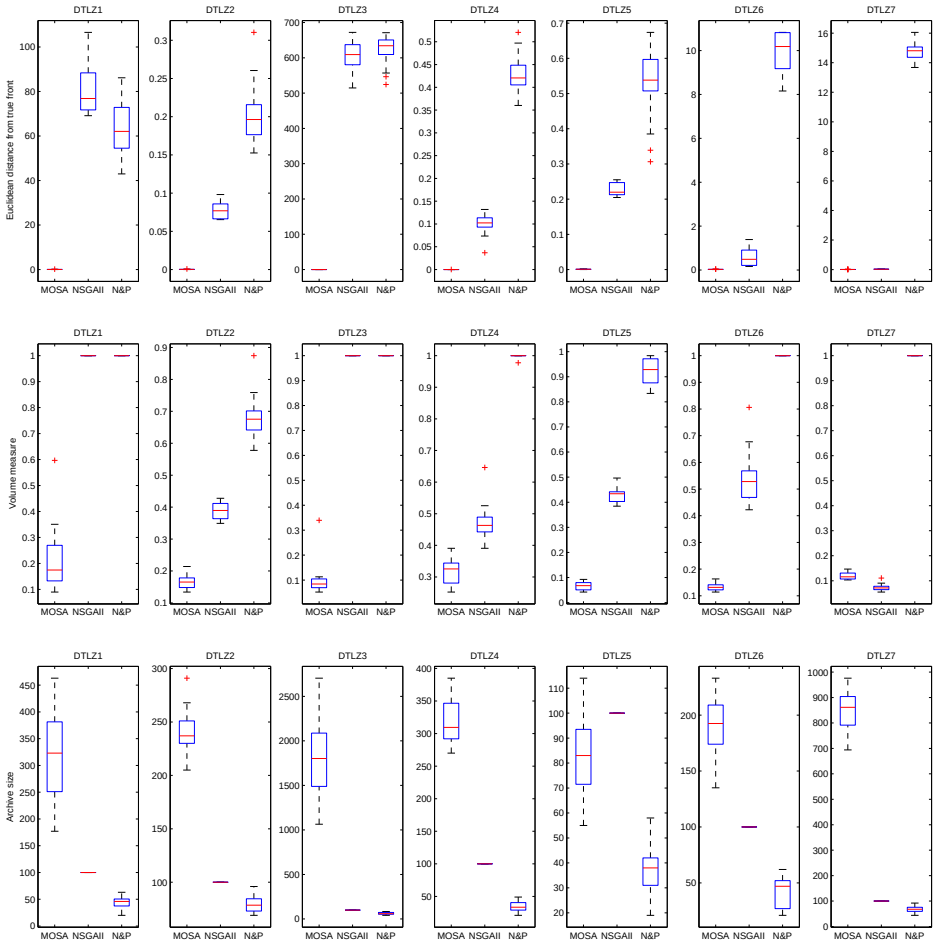

Figure 16: **Top:** Box plots of the average distance $d,\bar{\mathcal{P}}(F)$ of the archive from the true Pareto front for 20 runs of each of the DTLZ test problems, using the documented run lengths. **Middle:** Box plots of the volume measure $\mathcal{V}(\mathcal{P}, F)$ of the archive for each run. **Bottom:** Box plots of the size of the archive for each run. Each figure shows the results for MOSA, NSGA-II and Nam & Park's annealer.

7 cases although NSGA-II is significantly better on DTLZ7. NSGA-II achieved a good coverage on those problems for which it could converge near to the true front; the diversity maintenance in the algorithm encourages this. NSGA-II performed particularly well on DTLZ7 where the coverage was better than MOSA's. Nam & Park's algorithm was unable to effectively cover the true front for any problem.

The results for DTLZ4 effectively demonstrate why it is necessary to measure convergence in

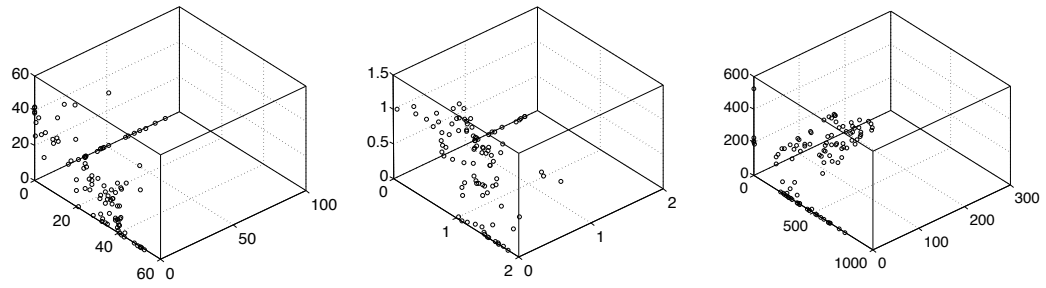

Figure 17: Archives from the NSGA-II algorithm using 100000 function evaluations on test problems DTLZ1 (left), DTLZ2 (centre) and DTLZ3 (right). The run with the median (over 20 runs) distance to the true Pareto front is shown.

terms of both distance and coverage, with MOSA having converged close to $\mathcal{P}$, but yielding a poor coverage of the front (in objective space), an artifact of the large range of volume magnification factors, as discussed earlier, also demonstrating that the visually appealing NSGA-II results were less well converged than it seems upon inspection. Confirming the impression given by the single run depicted in Figure 15, on average the annealer does not completely cover the true front for DTLZ7. As discussed above this could probably be improved by designing particular perturbation strategies for this particular problem; the NSGA-II implementation has a multiple point mutation scheme which performs very well on this problem (but could potentially hinder it in other problem constructions).

Figure 16 also shows how the final archive size varies across the 20 runs for each of the DTLZ problems used here. For the MOSA results it is clear that even the fronts generated by the least well-covered runs for each problem contain a large quantity of solutions relative to the run length. Furthermore the number of solutions generated for each problem is consistent across runs, although, as may be expected, problems with multiple local fronts (DLTZ1 and DTLZ3) have a larger spread. The NSGA-II algorithm is constrained to a predefined size (100 solutions in the work presented here) and Nam & Park's annealer does not generate large sets of solutions as it does not converge close to the true front. While it has traditionally been considered detrimental to maintain large archives due to the computational expense, MOSA's energy calculation benefits from large archives and limited archives may inhibit convergence as discussed by Fieldsend et al. [2003]. Using modern data structures, the computational expense of maintaining large archives is no longer a significant problem, as again shown by Fieldsend et al. [2003].

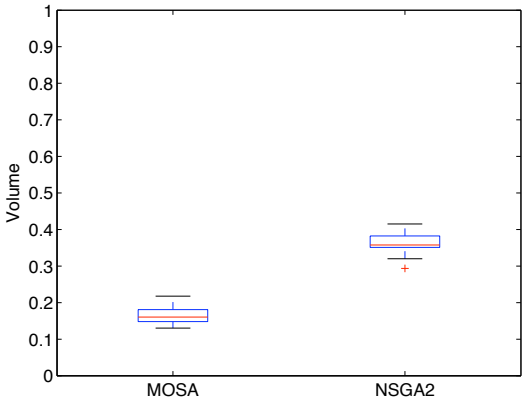

Figure 18: Boxplots comparing the volume measure $\mathcal{V}(\mathcal{P}, F)$ of estimated Pareto fronts for the MOSA (1000 function evaluations) and NSGA-II (100000 function evaluations) algorithms on DTLZ2. The MOSA results represent 100 solution samples from the generated archives.

In these comparisons a relatively small numbers of evaluations have been allowed for each algorithm in order to test rapid convergence, which is desirable in many industrial problems where function evaluations are expensive. It could be claimed, however, that this prejudices the results against the population based search of NSGA-II and in favour of MOSA, as it might be expected that MOSA would demonstrate rapid convergence and slow coverage, while NSGA-II would converge slowly but demonstrate superior coverage subsequent to convergence. While the results presented earlier show that MOSA does not demonstrate this behaviour, additional experiments, allowing NSGA-II 100,000 function evaluations for each of DTLZ1, DTLZ2 and DTLZ3, address this concern. Figure 17 shows the median archive, as calculated earlier, of NSGA-II on the DTLZ1, DTLZ2 and DTLZ3 problems after 100,000 function evaluations. DTLZ1 and DTLZ3 present the most difficult problems for NSGA-II, since they have many local fronts, and DTLZ2 presents the least barrier to convergence. It is clear from these figures that the fronts produced by NSGA-II, while closer than those produced in the shorter runs, are still distant from the true front for DTLZ1 and DTLZ3. The central plot of Figure 17, however, shows a front produced by NSGA-II after 100,000 evaluations which is close to the true front (as was achieved in the shorter runs presented earlier). To further analyse this result, a boxplot is given in Figure 18 comparing the dominated volumes of NSGA-II on these long runs to the MOSA results presented earlier on short runs. To ensure that MOSA's unconstrained archives do not influence these results (although this is considered a strength of the

MOSA algorithm over constrained archive algorithms), each of the MOSA results was randomly sampled three times, each for a front of 100 solutions, to match NSGA-II's constrained population. Even after far fewer evaluations, the sampled arhives generated by MOSA show significantly greater dominated volumes, demonstrating that MOSA is able to generate archives rapidly which are both close to, and well spread across, the true front which are competitive with NSGA-II over a great number of evaluations.

3.6 CDMA network optimization

Mobile telephone subscribers are allocated to one of a number of distinct *cells* or *sectors* comprising the telephone network. Cells may vary in extent from a few tens of metres (in a large office building) to several kilometres (in rural areas). Each cell is served by a single antenna and as the phone subscriber moves to a new location a 'handover' is made to a new cell in which the radio signal is stronger. The performance of the network as a whole and the quality of service enjoyed by individual subscribers is dependent upon a large number of operating parameters (many hundreds for a small network), some associated with the antenna and radio interface itself (such as the antenna azimuth and downtilt) and others associated with the network as a whole, such as the handover policy [Korhonen, 2001]. In addition performance itself may be evaluated in terms of several different metrics, for example: the network capacity (number of simultaneous calls); coverage (area served); and mean cell traffic channel power. The simultaneous optimisation of all these competing objectives is generally impossible and here the MOSA is used to investigate the trade-offs between them. Recent work using multi-objective optimisation in the mobile telecommunications domain has been undertaken: Ben Jamaa et al. [2004] have used multi-objective genetic algorithms for cell planning in order to optimise the cost and coverage of a network, and Szabó et al. [2003] have used multi-objective evolutionary algorithms for discovering the cost-interference trade-off when allocating transmitter placement and assigning transmission frequencies in time division multiple access (TDMA) networks.

The parameters to be optimised in this study are taken to be the pilot powers of a CDMA (Code Division Multiple Access) network for this study. The pilot power may be loosely thought of as the power with which the cell transmits to establish initial communication with phones in its cell. Pilot power is a particularly important parameter in CDMA networks because cells transmit continuously and if the pilot power is too great a cell may drown out its neighbours, but will not

be heard if the pilot power is too low. In this study there were 94 pilot powers as parameters and three objectives were optimised: the pilot pollution factor, defined as the number of pilots that each subscriber receives within 5dB of the dominant pilot; the mean downlink traffic channel (TCH) outage factor defined as the number of subscribers attempting to exceed their TCH power limit; and the mean reverse link server penalty which quantifies the unbalancing of the reverse link. This is done by comparing the propagation loss between each subscriber and its serving cell and the smallest propagation loss between that subscriber and any cell—the average difference between these values across all subscribers is defined as the reverse link penalty.

Unlike optimisation of test problems, as in section 3.5, the properties of the CDMA search space are not known in advance. Particularly, it is not known if the problem exhibits local front behaviour, where an optimiser must make several successive movements out of a locally optimal region of parameter space in order to locate the globally optimal region which corresponds to the Pareto front in objective space.

The results reported here are for an operational CDMA-1X network consisting of 94 sectors. Computational optimisation is feasible for this system due to the employment of a proprietary mathematical model of the downlink air interface which permits rapid evaluation of new configurations. In this study the pilot power of each sector was allowed to vary over the range from 1.0W to 3W in 0.5W intervals. Initially the pilot powers were set to their minimum feasible values and the MOSA was initialised to a temperature that yielded a 50% acceptance rate for derogatory moves, as described in section 3.4.1. The computational temperature was then reduced every $L_k = 100$ proposals by a factor of $\beta_k = 0.958$. The annealer was run for 100000 evaluations of the objective function.

Figure 19 shows the estimated Pareto front obtained, which consists of 965 solutions. Previous work by Motorola using a standard genetic algorithm optimising a composite objective function locates after 100000 function evaluations a single solution which is dominated by almost all of the non-dominated archive. The GA solution is distant from the archive (and therefore not shown in Figure 19) but it is likely that the genetic algorithm would have located a point in the vicinity of the front generated by the simulated annealer if it had been permitted a greater number of objective evaluations. The unavailability of the industrial code precluded making a comparison of MOSA and another single or multi-objective optimiser (for example NSGA-II), however the principal advantage

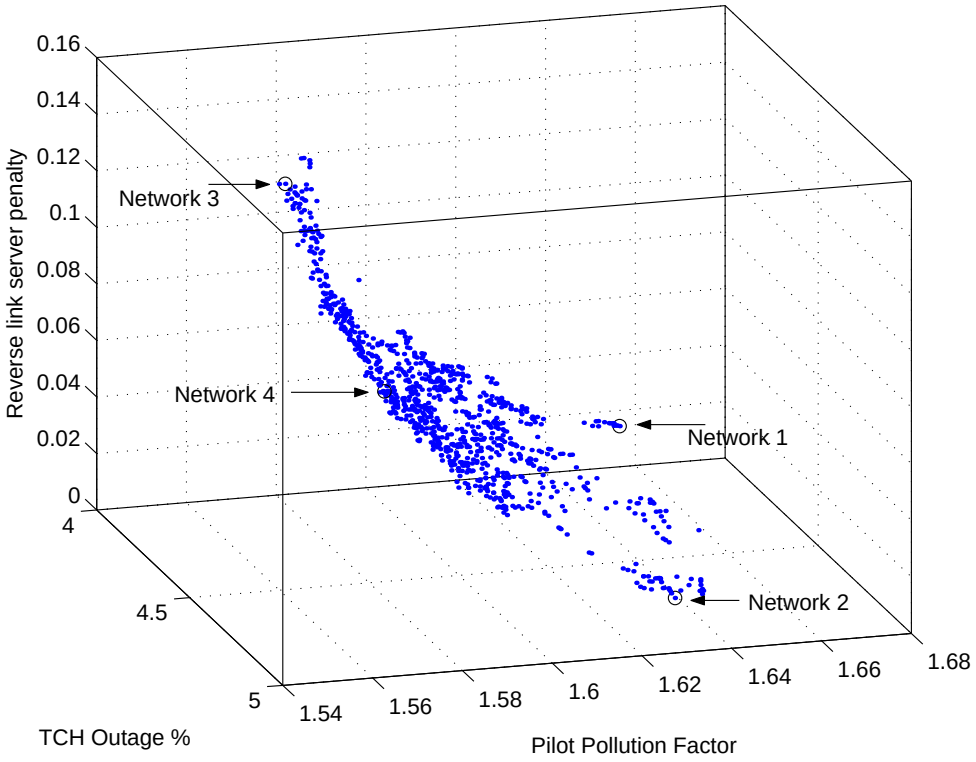

Figure 19: Estimated Pareto front for network pilot power optimisation.

of the annealer is not the reduction in the time taken to find a desirable solution (although this is considerable), but the frontal nature of the results generated by the simulated annealer. The front which has been located is clearly curved in objective space and displays to the network engineer the range of trade-offs which may be made in configuring the network.

The central portion of the network configuration corresponding to each of the solutions circled in Figure 19 is shown in Figure 20. In this figure, the pilot power for a sector is indicated by the length of an arrow rooted at the antenna location (antennae masts frequently support two or three antennae serving different sectors). It is interesting to note that each of the network configurations is very similar, despite their extreme relative frontal locations. This figure provides important information about the network to a network engineer, since some pilot powers seem to have single optimal values, particularly those in the less populated areas of the network with fewer interactions. However, as might be expected, it can be seen that the configuration with low pilot pollution (network 3) restricts pilot powers in sectors that face each other. The interaction between pilot powers and the other

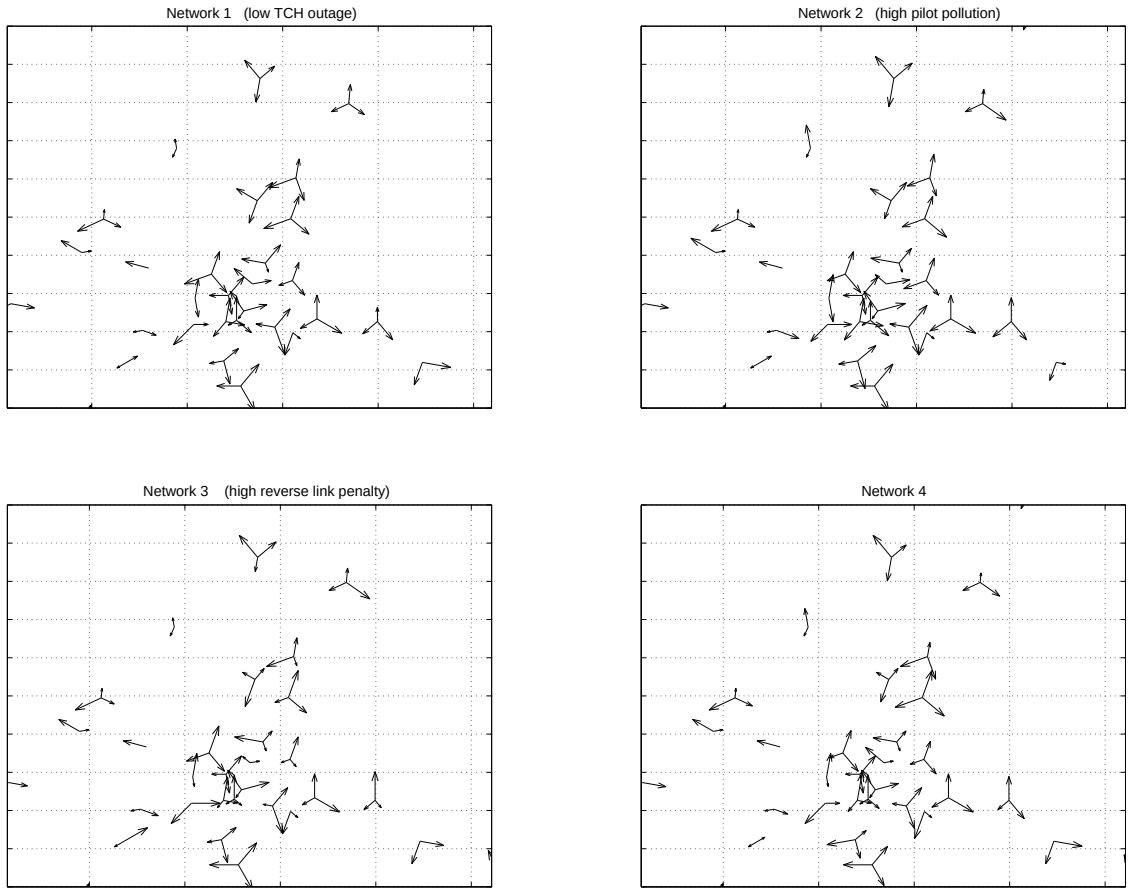

Figure 20: Pareto-optimal central network configurations corresponding to the labelled points on the Pareto front in Figure 19.

objectives is less clear, but the MOSA methodology provides a mechanism for locating these optimal configurations. Current work involves applying this methodology to the optimisation of antennae pilot powers, azimuths and downtilts in larger networks.

3.7 Conclusions

In this chapter, an energy measure for use in multi-objective SA has been presented which is based on the fundamental notion of dominance, rather than employing a weighted combination of the objectives. Simulated annealers employing this measure were shown to have good convergence properties on the first seven DTLZ test functions [Deb et al., 2002c, 2001]. An extensive comparison

with the evolutionary algorithm NSGA-II and the multi-objective annealer proposed by Nam and Park [2000] on these problems shows that the annealer consistently generates archives closer to the true front than NSGA-II and Nam & Park's annealer and that, in all but one case, produces a significantly better coverage (on DTLZ7 NSGA-II generates fronts with a fuller coverage of the front, possibly due to the more specialised multiple point perturbations used).

It should be emphasised that the MOSA algorithm was not 'tuned' for each of these problems, but run from a randomly chosen initial condition. More rapid convergence on many of these problems can be achieved by careful tuning of the annealing schedule but, of course, this requires many runs to discover an optimal schedule; while tuning the annealing schedule is important in industrial applications where the annealer is to be run many times on similar problems (e.g., CDMA network optimisation), for fairness of comparison no tuning of the annealer to particular *test* problems has been performed; here all the algorithms were simply limited to an evaluation count approximately at which the first algorithm converges on the true front. The only instance in which it was necessary to alter the annealers was for DTLZ7, for which single point perturbations of solutions close to the Pareto front result in infeasible proposals, however, it should be noted that the problem with single point perturbations will afflict all stochastic searches (evolutionary algorithms, GAs, etc) that perturb a solution to generate a new candidate solution and that in this problem, MOSA performed almost as well with a very basic multiple point perturbation scheme as NSGA-II did, which uses a more advanced scheme.

An advantage of the dominance based energy measure for multi-objective optimisation is that it is not *a priori* biased towards any part of the front. Weighted sum optimisers implicitly use distance information in objective space, which renders them sensitive to the relative scalings of the objectives, whereas the MOSA algorithm is robust to rescalings of the objectives. Indeed, if the relative importance or scales of the objectives were known in advance it might be more straightforward to optimise a single, appropriately weighted, sum of the objectives. Notions of dominance and Pareto optimality are well suited to handling competing objectives whose relative importance is *a priori* unknown and it is therefore natural to eschew metric information in favour of dominance concepts in order to guide the search. Indeed, it has been argued here that the dominance based energy tends to promote exploration in sparsely populated regions and in practice it has been shown that estimated fronts evenly and widely cover the true front. An area of future investigation

which has arisen from this work is to use the singular values and vectors of the Jacobian matrix to guide the search on the front towards areas that would otherwise be sparsely populated.

Determining an efficient scale on which to make proposals is more complicated in the multi-objective case than the uni-objective case, because some proposals work to advance the front, while others traverse the front. Here simple heuristics have been proposed to adapt the perturbation scales and future work involves applying machine learning techniques to learn the local mapping between parameter and objective space in order to more sensitively control the search direction.

When applied to the optimisation of a CDMA network the annealer is successful in generating a front with a large number of mutually non-dominating solutions, the vast majority of which are superior to the single solution located by a genetic algorithm optimising a composite objective function. This allows a network engineer to make an informed decision regarding network configurations with additional knowledge of the costs of the trade-offs involved. Further work in this direction covers optimisations involving both more parameters, such as antenna azimuths and downtilts, and additional objectives relating to the quality of service for subscribers.

The $E(\omega)$ presented here is a measure of a portion of the dominating set, namely $\mu(\tilde{F}_{\mathbf{x}})$, which is a close relation to Fleischer's recently proposed measure [Fleischer, 2003]; loosely, this measure deals with the area of the dominating surface—the attainment surface—while Fleischer's considers the dominated volume. Many questions are raised in the construction of the MOSA algorithm, such as the merits of a dominance rather than a volume based energy based on Fleischer's measure, whether it is best to optimise a single solution like traditional simulated annealers, or a set of solutions concurrently as an evolution strategy, or genetic algorithm and whether a more uniform coverage of non-uniform fronts, such as in DTLZ4, can be attained; these issues are investigated in Chapter 4.

Although a proof of convergence for simulated annealers based on the dominance measure remains to be completed, this would be an interesting area of future work, together with the application of the annealer to other large scale problems.

Chapter 4

Alternative Strategies for Multi-Objective Simulated Annealing

4.1 Introduction

Chapter 2 discusses a variety of techniques for locating approximations to the Pareto front, including genetic algorithms and evolution strategies. Evolutionary algorithms usually maintain a population of putative solutions to the optimisation problem, which allows them to be adapted to the multi-objective case in which a set of solutions is sought. Simulated annealing (SA) [Kirkpatrick et al., 1983], a well-known algorithm for solving single-objective optimisation problems, however, has received relatively little attention from the multi-objective optimisation community. Given a current approximate solution to the optimum, the optimiser *state,* a *greedy* search adopts as the new current solution a perturbation to the current solution that improves the objective. SA on the other hand may be regarded as a greedy search in which non-greedy perturbations (that is those that degrade the objective) are permitted because these, seemingly *detrimental*, *exploratory* perturbations may lead to eventual improvements in the objective. The magnitude of permitted exploratory perturbations,

measured as the difference in *energy* between the current and perturbed solution, is decreased during the optimisation by decreasing a computational temperature according to an annealing schedule; in the low temperature limit the search becomes entirely greedy. A particular attraction of SA is the existence of the proof of Geman and Geman [1984] that guarantees convergence to the global minimum for single objective problems provided that the annealing rate is sufficiently slow.

The majority of schemes that adapt simulated annealing to more than one objective [Serafini, 1994, Ulungu et al., 1999, Czyżak and Jaszkiewicz, 1998, Nam and Park, 2000, Hapke et al., 2000, Suppapitnarm et al., 2000, Tuyttens et al., 2003] have focused on a *single* solution, and determining the quality of perturbations to that single solution using a weighted sum of objectives or an archive of the best mutually non-dominating solutions discovered thus far in the optimisation. The multi-objective simulated annealing, MOSA, scheme proposed in Chapter 3 generally outperforms the popular and effective genetic algorithm NSGA-II [Deb et al., 2002a] on standard test problems [Deb et al., 2001, 2002c] but also focusses exclusively on the use of a single solution for a state.

An alternative to focusing on a single solution, used in many evolutionary algorithms, is to regard the mutually non-dominating *set* of solutions discovered thus far as the quantity to be annealed. Algorithms based on this idea have not yet been explored in the multi-objective simulated annealing literature. In light of the close connection with evolutionary schemes that may be regarded as set-based multi-objective greedy searchers or set-based simulated annealers at temperature zero, this chapter explores the connections and relative efficiencies of set-based and single-solution-based simulated annealing schemes together with their zero temperature or greedy counterparts. It also explores alternative formulations of the energy change and presents a new algorithm for calculating the incremental energy change based on dominated volume when an element is added to the set, even when this element dominates set members.

Following this introduction, the components of a general search algorithm are discussed in the context of a general multi-objective simulated annealing algorithm. Well known algorithms are shown to be specializations of this general algorithm to greedy searching, depending on the choice of energy and whether a single solution or a set of solutions is regarded as the current state. Subsequently, in section 4.3 single solution state and set-based state simulated annealing algorithms are described, together with their greedy counterparts. These algorithms are empirically compared on standard test problems in section 4.4. Somewhat surprisingly it is found that the greedy versions of the algorithms,

which exclude exploratory perturbations, are more efficient than versions that permit exploratory moves. This suggests the conclusion that, in contrast to single objective problems, the optimum of many multi-objective problems can be located by a greedy algorithm. The fitness landscape of extant test problems is discussed and new test problems, which require exploratory perturbations to locate the optimum, are introduced in section 4.5.

4.2 Search Techniques

Many optimisation procedures can be decomposed into the following components: a *state*, denoted by ω, which is a set of one or more solutions $\mathbf{x}$; a method of generating a new or *perturbed* state ω' from ω; an *energy* of a state to be minimised $E(\omega)$ together with a method of comparing the quality of two states, which is called an *energy change* $\delta E(\omega, \omega')$ in the simulated annealing literature or the *fitness* in the genetic algorithms community; and finally a rule to determine whether to accept the perturbed state as the new current state based on $\delta E(\omega, \omega')$. A search then proceeds iteratively from an initial state by perturbing the state and determining whether the energy of the perturbed state is lower (fitter) than the original; perturbed states with lower energies are generally adopted as the new state, but some algorithms, particularly simulated annealing, permit *exploratory* moves in which a state with a higher energy is accepted. Having accepted the new state as the current state, the cycle repeats until some convergence criterion is met. Since the state need not contain the best solution visited so far, and for multi-objective problems there are usually several mutually non-dominating optimal solutions, many algorithms maintain an additional *archive* or record of the optimal solution visited thus far in the search.

As will become apparent, many optimisation procedures may be regarded as particular cases of a general multi-objective simulated annealing algorithm, which is summarised in Algorithm 5.

As discussed in Chapter 2, uni-objective problem simulated annealing, proposed by Kirkpatrick et al. [1983], can be regarded as the computational analogue of slowly cooling a metal so that it adopts a minimum-energy crystalline state. In order to minimise a computational energy $E(\omega)$ a computational temperature T is progressively lowered during the optimisation according to an annealing schedule. At high temperature the state changes freely, whereas as the temperature is lowered the state is increasingly confined due to the high energy cost of rearrangement. At each T the SA algorithm aims to draw samples from the equilibrium distribution $\pi_T(\omega) \propto \exp\{-E(\omega)/T\}$.

Algorithm 5 Multi-objective simulated annealing

Inputs:

$\{L_k\}_{k=1}^K$	*Sequence of epoch durations*
$\{T_k\}_{k=1}^K$	*Sequence of temperatures,* $T_{k+1} < T_k$
ω	*Initial state*

```
1:  F := ω                                                  Initialise archive
2:  for k := 1, ..., K
3:    for i := 1, ..., L_k
4:      ω' := perturb(ω)
5:      δE(ω, ω') = energyChange(ω, ω')
6:      u := rand(0, 1)
7:      if u < min(1, exp(−δE(ω, ω')/T_k))
8:        ω := ω'                                           Accept new current state
9:        foreach y ∈ ω'                                    Maintain the archive
10:         if z ⊀ y  ∀z ∈ F
11:           F := {z ∈ F | y ⊀ z}                          Remove dominated points from F
12:           F := F ∪ {y}                                  Add y to F
13:         end
14:       end
15:     end
16:   end
17: end
```

As $T \to 0$ the probability mass of π_T is increasingly concentrated in the region of the global minimum of E, so eventually any sample from π_T almost surely lies at the minimum of E.

The computational temperature is fixed at T_k during each of the $k = 1, \ldots, K$ epochs; during each epoch L_k samples are drawn from the equilibrium distribution $\pi_T(\omega)$. This is achieved by Metropolis-Hastings sampling [Metropolis et al., 1953], which involves making proposals ω' (line 4 of Algorithm 5) that are accepted with probability

$$A = \min\left(1, \exp\{-\delta E(\omega, \omega')/T\}\right) \tag{23}$$

where the energy difference between the states ω and ω' is

$$\delta E(\omega, \omega') \equiv E(\omega') - E(\omega). \tag{24}$$

(Lines 6-8 of Algorithm 5.) Initially, when T is high, perturbations from ω to ω' which increase the energy are likely to be accepted (in addition to perturbations which decrease the energy, which are

always accepted). This allows the annealer to explore the search space, so as not to become trapped in local minima. As T is reduced only perturbations leading to small increases in E are accepted, so that only limited exploration is possible as the system settles on the global minimum. As discussed in Chapter 2 convergence to the global minimum of uni-objective problems is guaranteed if the cooling schedule is sufficiently gradual [Geman and Geman, 1984], but experience has shown SA to be a very effective optimisation technique even with relatively rapid cooling schedules.

For multi-objective problems the usual uni-objective algorithm is augmented by the addition of an archive F that represents the best estimate of the Pareto front found thus far; that is, it is the non-dominated set whose members are not dominated by any solution visited during the optimisation. Lines 9-14 of Algorithm 5 maintain this archive, which is initialised from the initial state (line 1). Each element of the state that is not dominated by an existing element of the archive is added to the archive (line 12) and any members of F that are dominated by the new entrant are deleted from F (line 11).

4.2.1 States and perturbations

In many algorithms, such as conventional uni-objective simulated annealing and the multi-objective simulated annealer presented in Chapter 3, the state consists of a single solution: $\omega = \{\mathbf{x}\}$. Perturbations are made to the single solution: $\mathbf{x} \mapsto \mathbf{x}'$; and the energy change $\delta E(\omega, \omega') = \delta E(\mathbf{x}, \mathbf{x}')$ is just the difference in energy between the current solution and the perturbation to it.

An alternative is to regard a set of non-dominating solutions as the state. In this case perturbations are made by selecting, μ solutions $\mathbf{x}_i$, $i = 1, \ldots, \mu$, from ω, perturbing each one in turn, $\mathbf{x}_i \mapsto \mathbf{x}'_i$, and forming the perturbed state from ω and the perturbed solutions.

A superficially attractive method of deriving the new state would be to form the union of ω and the perturbations, removing those elements of ω' that are dominated by other elements:

$$\omega' = \text{nondom}\left(\omega \bigcup_{i=1}^{\mu} \{\mathbf{x}'_i\}\right). \tag{25}$$

This scheme ensures that ω' is a mutually non-dominated set. However, it is unsuitable for a new state because, by construction, no element of ω' can be dominated by an element of ω, which implies that ω' is never inferior to ω so that no exploratory perturbations can be made. In fact, this scheme

Figure 21: Set perturbation method. *Left:* $\mathbf{x}'$ dominates members of ω *Right:* $\mathbf{x}'$ is dominated by members of ω. The large blue point represents $\mathbf{x}'$, the green dots represent those members of ω which, together with $\mathbf{x}'$, form ω', and the red points are the members of ω which are not present in ω'.

leads directly to elitist algorithms, as exemplified by SPEA [Zitzler and Thiele, 1999] or that of Fieldsend and Singh [2002a]. The *nondom* function is used to return all the members of the set which are not dominated by any other member of the set.

Exploratory perturbations to the state are permitted by retaining all the perturbations which are mutually non-dominating, but removing any elements of ω that either dominate or are dominated by the perturbations:

$$X' = \text{nondom}\left(\bigcup_{i=1}^{\mu}\{\mathbf{x}'_i\}\right) \tag{26}$$

$$\omega' = X' \bigcup \{\mathbf{x} \in \omega \,|\, (\mathbf{x} \not\prec \mathbf{x}' \wedge \mathbf{x}' \not\prec \mathbf{x}) \forall \mathbf{x}' \in X'\}. \tag{27}$$

This scheme is illustrated in Figure 21 and ensures that ω' is itself a mutually non-dominating set as well as allowing the inferior states to be visited by the search. The efficacy of set-based annealers based on this scheme is investigated in section 4.3.3.

The energy change, on the basis of which the perturbed set is accepted, is then the difference in energies between the two sets ω and ω'. Methods of calculating the energy change for particular algorithms are discussed below. It should also be noted that the performance of an optimiser can be affected by the manner in which solutions to be perturbed are selected from ω; particular methods are discussed subsequently.

For multi-objective problems, the set-based state is appealing because the Pareto front itself is

generally a set. Since perturbations are made to solutions across the entire estimated front, one may expect the search to proceed more efficiently as it will be less likely to become 'stuck' at a single solution. On the other hand, it might be expected that a single solution state might be efficient because it can rapidly reach the vicinity of the Pareto front (without the need to perturb all the elements of a set-based state to the Pareto front), after which the front is 'filled out' by perturbations transverse to the front; indeed the multi-objective simulated annealing scheme presented in Chapter 3 has been observed to work in this fashion. Finally, it appears that theoretical results for uni-objective SA are more easily adapted to the multi-objective situation using a set-based formulation.

4.2.2 Greedy Search

Perhaps the simplest search method is a greedy search. Greedy methods are defined as those which always make the locally optimal choice: that is, ω' is always accepted if it has a lower energy than ω; but perturbations with higher energies are never accepted. Simulated annealing with $T = 0$ may be recognised as a greedy search because there is zero probability that a state with $\delta E(\omega, \omega') > 0$ will be accepted.

Various greedy-type evolutionary algorithms have been developed for multi-objective optimisation, perhaps most prominently those based on $(\mu + \lambda)$-evolution strategies [Knowles and Corne, 1999, 2000, Fieldsend and Singh, 2002a, Everson et al., 2002, Fieldsend and Singh, 2002b, Fieldsend et al., 2003, Laumanns et al., 2002, Everson and Fieldsend, 2006, Laumanns et al., 2004, eg]. These incorporate algorithms in which the state is a single solution, for example the well-known PAES algorithm [Knowles and Corne, 1999], and those in which the state is comprised of the non-dominated archive from which a single solution is selected at each iteration [Fieldsend and Singh, 2002a]. In both cases the basis of the acceptance criteria is simple: if a perturbation is better than, or non-dominated by archive members the new state is accepted, if it is worse the state is rejected. (This approach is extended with a more advanced gridding applied over this in PAES to promote uniform exploration of $\mathcal{P}$).

The strictness of the greedy search acceptance criterion means that the perturbation operator(s) used can greatly influence the results. Since the algorithm can never make moves to regions of higher energy, it can become stuck on a local optimum, relying on a single perturbation to carry the state to a region of lower energy. Such perturbations may be rare, particularly when the dimension of

the search space is large, or indeed non-existent, meaning that the global optimum is not found in finite time. The attraction of methods such as simulated annealing is that they permit exploratory perturbations to high energy regions that may subsequently lead to low energy states.

4.2.3 Energy Functions

Choice of an energy function to compare the quality of two states is fundamental to the operation of simulated annealing and related algorithms. Energy functions may be classified into three major groups as follows.

Weighted sum

As discussed in Chapter 2, the most obvious method of applying a single-objective optimiser such as simulated annealing to a multi-objective problem is to optimise a weighted sum of the objectives. For simulated annealing an energy is thus formed as

$$E(\mathbf{x}) = \sum_{i=1}^{D} w_i f_i(\mathbf{x}). \tag{28}$$

Several works have investigated this approach using a single solution state [Serafini, 1994, Czyżak and Jaszkiewicz, 1998, Ulungu et al., 1999, Nam and Park, 2000, Hapke et al., 2000, Suppapitnarm et al., 2000, Tuyttens et al., 2003]. Generally these procedures always accept a perturbation if it is not dominated by the current archive, but use the composite energy (28) to calculate the acceptance probability if the perturbation is dominated by an element of F.

While convergence to a single point on the Pareto front is guaranteed (as a consequence of Geman & Geman's proof for a scalar SA [Geman and Geman, 1984]), there are significant drawbacks to this approach. It is not clear how the relative weights of the objectives should be determined in advance, and these will determine which point on the Pareto front is eventually located. In addition, as noted in Chapter 2, it has been shown that parts of the Pareto front are inaccessible when fixed weights are employed [Das and Dennis, 1997]. Recognising this, investigators have proposed a variety of schemes for adapting the w_i during the annealing process to encourage exploration along the front.

Dominance

Since the dominance relation is used for comparison of individual solutions in multi-objective problems and is widely used in multi-objective evolutionary and genetic algorithms, it is desirable to use dominance to compare the quality of solutions. A further property recommending its use is that dominance does not use metric information in the objective space, rendering a dominance based energy function invariant to rescalings of the objectives.

Dominance based energy functions utilising only the relative dominance of a solution compared to previously located solutions have been investigated in Chapter 3, and shown to significantly out-perform a simulated annealer utilising a weighted sum energy and to generally out-perform the NSGA-II multi-objective genetic algorithm. Specific dominance based energy functions for single solution and set states are described in section 4.3 and empirically compared in section 4.4.

Volume

Fleischer [2003] proposes that a multi-objective simulated annealer can be trivially constructed by defining the energy of a state to be the negative of the volume of objective space dominated by the state.[1] It is envisaged that the state is a set of non-dominated solutions and the volume dominated by this set is clearly maximised when the set equals the Pareto front. Note that there is no single solution state analogue using the dominated volume, because, while it is possible to evaluate the volume dominated by a single solution, this is equivalent to optimising an unweighted product composite objective function.

While the dominated volume is attractive from a purely theoretical viewpoint, its practical computation turns out to be prohibitive for more than two objectives. Its performance for two objective problems is examined in section 4.4.

4.3 Algorithms

As described in the previous section, different multi-objective optimisation algorithms related to simulated annealing, and fitting into the general framework of Algorithm 5, may be obtained through different choices of state (single solution or set of solutions), energy (weighted sum, dominance

[1] Practically, the volume of some hyper-rectangle in objective space dominated by the state must be used to avoid infinite volumes.

Table 3: Multi-objective optimisation algorithms based on simulated annealing.

	Dominance energy		Volume energy
	Single $\mathbf{x}$ state	Set state	
Exploratory: $T > 0$	MOSA (Smith et al. [2006])	SAMOSA	VOLMOSA
Greedy: $T = 0$	MOSA0 PAES (Knowles and Corne [1999])	SAMOSA0 Set-sampled (1+1)-ES (Fieldsend and Singh [2002a])	VOLMOSA0 -

based or volume based) and whether the search is exploratory (computational temperature $T > 0$) or greedy ($T = 0$). Table 3 summarises greedy and exploratory algorithms using dominance and volume energies, together with single solution and set states, which are described in this section; their performance on standard test problems is compared in section 4.4. Greedy multi-objective optimisers have appeared in the literature using single solution states (e.g. PAES [Knowles and Corne, 1999]) and set states (e.g. Fieldsend and Singh [2002a]); here direct comparison with exploratory optimisers is facilitated by setting to zero the computational temperature of the exploratory algorithms MOSA (single solution state, dominance energy), SAMOSA (set state, dominance energy) and VOLMOSA (set state, dominated volume energy). The zero temperature algorithms are dubbed MOSA0, SAMOSA0 and VOLMOSA0.

4.3.1 MOSA

The multi-objective simulated annealing algorithm used here was first proposed in Smith et al. [2004] and is discussed at length in Chapter 3; for completeness it is briefly summarised here. It maintains a single current solution $\omega = \{\mathbf{x}\}$ which is perturbed to create a new state, acceptance of which is determined using an energy based upon the proportion of an archive F of previously located non-dominated solutions that is dominated by $\mathbf{x}$ and $\mathbf{x}'$. The energy change is defined as:

$$\delta E(\omega, \omega') = \frac{1}{|\tilde{F}|}\left(|\tilde{F}_{\mathbf{x}'}| - |\tilde{F}_{\mathbf{x}}|\right) \tag{29}$$

where $\tilde{F} = F \cup \{\mathbf{x}\} \cup \{\mathbf{x}'\}$ is the archive augmented by the state and the perturbation, and $|\tilde{F}_{\mathbf{x}}|$ and $|\tilde{F}_{\mathbf{x}'}|$ are the number of solutions in $\tilde{F}$ dominated by $\mathbf{x}$ and $\mathbf{x}'$ respectively. If $\tilde{F}$ is a non-dominating set the energy difference between any two of its elements is zero. Note also that $\delta E(\omega', \omega) = -\delta E(\omega, \omega')$. The inclusion of the current solution and the proposal in $\tilde{F}$ means that $\delta E(\omega, \omega') < 0$

if $\mathbf{x}' \prec \mathbf{x}$, which ensures that perturbations that move the estimated front towards the true front are always accepted. Proposals that are dominated by one or more members of the current archive are accepted with a probability depending upon the difference in the number of solutions in the archive that dominate $\mathbf{x}'$ and $\mathbf{x}$. In practice when there are few elements in $\tilde{F}$, it is augmented by interpolating from the attainment surface in order to increase the energy resolution, as described in Section 3.3.3.

Although sophisticated perturbation schemes that adjust the perturbation scale during optimisation may be used with MOSA (Section 3.4.2), a straightforward method, applicable also to SAMOSA and VOLMOSA, is employed in this chapter. The perturbation scheme is a single-point method in which one of the P decision variables of $\mathbf{x}$ is selected at random and perturbed by the addition of a random variable ϵ drawn from a heavy-tailed Laplacian distribution, $p(\epsilon) \propto e^{-|\sigma\epsilon|}$, where the scale factor σ sets the magnitude of the perturbation. In this work $\sigma = 0.1$, corresponding to 1/10th of the range of the decision variables; experiments suggest that the performance of these algorithms is not strongly dependent on the magnitude of σ.

4.3.2 SAMOSA

In contrast to the MOSA algorithm which operates with a state which is a single solution, resulting in behaviour which locates the Pareto front and then moves across it, discovering the extent of the front as it travels, the SAMOSA algorithm presented here aims to converge a set of solutions covering the front simultaneously.

The state, ω, of the SAMOSA is a set of mutually non-dominating solutions. This state is not of a fixed size and, due to the requirement of simulated annealing of an ability to move away from, as well as towards, the Pareto front, solutions in ω, while mutually non-dominating, may be dominated by previously discovered solutions. The state is initialised with a single, randomly generated solution.

SAMOSA perturbs the state by perturbing a single member of the state, $\mathbf{x} \mapsto \mathbf{x}'$, using the one-point scheme described previously for MOSA. In order that ω' is a mutually non-dominating set and that exploratory movements (those that result in the perturbed solution being dominated by element(s) of ω) are possible, the perturbed state is formed by adding to $\mathbf{x}'$ those elements of ω

Algorithm 6 SAMOSA perturbation.

```
1:  x := Uniselect(ω)
2:  x' := perturb(x)
3:  ω' := {x'}
4:  foreach u ∈ ω
5:     if (x' ⊀ u) ∧ (u ⊀ x')
6:        ω' := ω' ∪ {u}
7:     end
8:  end
```

that neither dominate $\mathbf{x}'$ or are dominated by $\mathbf{x}'$:

$$\omega' = \mathbf{x}' \bigcup \{\mathbf{x} \in \omega \,|\, (\mathbf{x} \not\prec \mathbf{x}' \wedge \mathbf{x}' \not\prec \mathbf{x})\}. \tag{30}$$

It is important that the member of ω which is perturbed is not itself removed from ω unless it is either dominated by, or dominates, the new solution; perturbing the selected member, instead of a copy, would fix the size of the state set at its initial size, one.

Creation of the perturbed state is straightforwardly accomplished as shown in Algorithm 6. The particular member of the ω to perturb is selected using a simple scheme dubbed 'Uniselect'. In the same spirit as the gridding scheme used by PAES [Knowles and Corne, 2000], the aim of Uniselect is to prevent clustering of solutions in a particular region of the front ω biasing the search because they are selected more frequently. To achieve this each time a solution is to be selected from ω an objective, i, is chosen at random (with equal probability), a random number u is drawn uniformly in the range $[\min_{\mathbf{x}\in\omega}(f_i(\mathbf{x})), \max_{\mathbf{x}\in\omega}(f_i(\mathbf{x}))]$, and the element of ω to be perturbed is the element with i^{th} coordinate closest to u: $\mathbf{x} = \arg\min_{\mathbf{x}\in\omega} |f_i(\mathbf{x}) - u|$. Although Uniselect has been used here, similar methods (e.g., PQRS [Fieldsend et al., 2003]) that counteract the effects of clustering yield comparable results.

The energy difference between ω and ω' is defined as the signed proportion of the state set ω that is removed in the state set ω':

$$\delta E(\omega, \omega') = \frac{1}{|\omega|}\left[|\{\mathbf{u} \in \omega \,|\, \mathbf{u} \prec \mathbf{x}'\}| \;-\; |\{\mathbf{u} \in \omega \,|\, \mathbf{x}' \prec \mathbf{u}\}|\right]. \tag{31}$$

If the solutions are removed from ω due to the insertion of a dominating solution as in the left image of Figure 21, a negative value is assigned, whereas if the solutions are removed due to the

Algorithm 7 SAMOSA Energy change calculation

```
1: δE := 0
2: foreach u ∈ ω
3:    if x' ≺ u
4:       δE := δE + 1
5:    else if u ≺ x'
6:       δE := δE − 1
7:    end
8: end
9: δE(ω, ω') := δE/|ω|
```

insertion of a dominated solution as in the right image of Figure 21, the assigned value is positive. Note that one or other of the two terms on the r.h.s. of (31) is always zero; if the perturbation does not dominate and is not dominated by any member of ω, then $\delta E(\omega, \omega') = 0$ so that this non-dominating perturbation is accepted. When members of ω' dominate a large proportion of ω, this is considered to be a large improvement, and a large negative $\delta E(\omega, \omega')$ is assigned. Similarly, when ω' is dominated by much of ω, the new state is significantly worse and so is assigned a large positive $\delta E(\omega, \omega')$. The magnitude of the energy change is correctly assigned to be small when there are small differences between ω and ω', and large when one set dominates much of the other.

Note that if F is a subset of the Pareto front $F \subseteq \mathcal{P}$ then $\delta E(F, \omega) \geq 0$; furthermore $\delta E(F, \omega) = 0$ if and only if all elements of ω are not dominated by (and are therefore members of) $\mathcal{P}$, so that the global optimum has minimum energy among all feasible sets.

As shown in Algorithm 7, the energy difference is simply calculated by comparing the perturbation with each of the elements in ω and for efficiency this procedure may be combined with the formation of ω' (Algorithm 6).

4.3.3 VOLMOSA

Although as noted, Fleischer [2003] suggested basing a multi-objective simulated annealer on the dominated volume, no report on the performance of such an annealer has appeared in the literature. An annealer with a set state is straightforwardly defined. Here the state, initialisation and formation of the perturbed state are identical to those of SAMOSA; the only difference in the algorithms is in the assessment of the energy difference. The energy $E(\omega)$ of a state ω is defined to be the volume of the hyper-rectangle defined by the origin and some reference point $\mathbf{r} \in \mathbb{R}^D$ that is dominated by elements of ω. The energy difference is then just the difference in energies of the states: $\delta E(\omega, \omega') =$

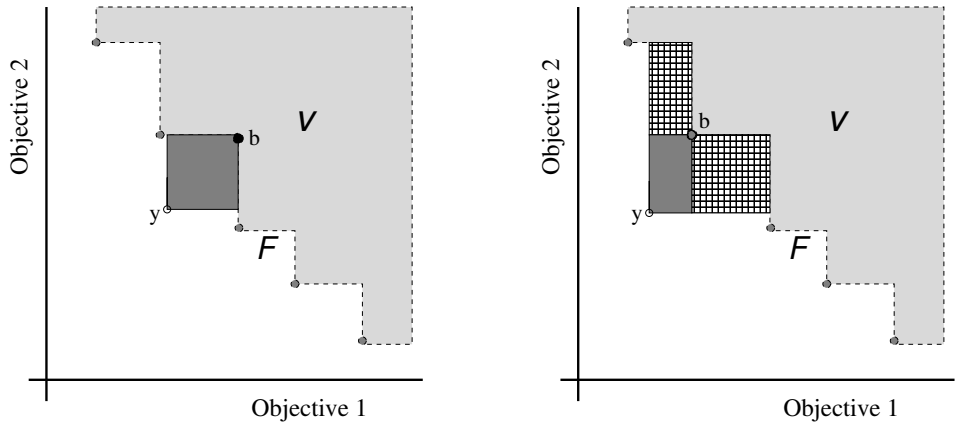

Figure 22: Incremental volume calculation examples. **Left:** The additional volume from **y** is a single rectangle. **Right:** Several additional rectangles contribute additional volume.

$E(\omega) - E(\omega')$.

Unfortunately, computational methods to exactly calculate the dominated volume for ω have a time complexity of $\mathcal{O}(|\omega|^D)$, rendering them prohibitively expensive even for small numbers of objectives [Fleischer, 2003, Huband et al., 2005]. Monte Carlo sampling, an approximate method, has been used in this context for comparing fronts [Fieldsend et al., 2003], but when the difference in volume between ω and ω' is small it is again prohibitively expensive to achieve the necessary accuracy.

Although the cost of calculating the volume dominated by a complete set ω is high, in evolutionary algorithms and VOLMOSA in particular the set of interest ω' commonly differs from ω only by a single solution. Taking advantage of this, an incremental method for dominated volume calculation is presented. This method is a logical extension of Fleischer's calculation [Fleischer, 2003], presented as a recursive method instead of explicitly stack-based, and extended such that the set does not need to be mutually non-dominating. Using this method, when a perturbation to ω generates a solution which is not dominated by the members of ω (that is: when the volume is not decreased), only the difference in volume between the two sets need be calculated, which is considerably simpler than the evaluation of the volume dominated by the entire set.

Algorithm 8 Incremental dominated volume change.

Inputs:
F *Non-dominated set with volume* v
$\{L_d\}_{d=1}^{D}$ *Elements of* F*, sorted by increasing coordinate* d
$\mathbf{y}$ *Point to be added to* F

```
1:  define addPoint(y) :
2:    if ∃u ∈ F|u ≺ y exit function; end
3:    for d := 1, ..., |D|
4:      for i := 1, ..., |F|          Find smallest u_d s.t. y_i < u_i
5:        u := L_{d,i}
6:        if u_d > y_d
7:          b_d := u_d
8:          exit loop
9:        end
10:     end
11:   for i := 2^D, ..., 1                  Find bounding vertices
12:     k := i
13:     for i := D, ..., 1
14:       if k ≥ 2^{j-1}
15:         k := k − 2^{j-1}
16:         v_j := b_j
17:       else
18:         v_j := y_j
19:       end
20:     end
21:     B := B ∪ v
22:   end
23:   B :=sort(B) \ y              Sort B by increasing dominance
24:   foreach u ∈ B
25:     addPoint(u)
26:   end
27:   v := v + ∏_{i=1}^{D}(b_i − y_i)       Add contribution to the volume
28: end define
```

Calculating incremental volume change

The basis of this method is to simply calculate the smallest addition of volume possible through the insertion of $\mathbf{y}$, and to recursively 'fill in' the volumes until the existing dominated volume is reached, as can be seen in Figure 22. The left image in Figure 22 shows the simplest situation for the addition of a new point $\mathbf{y}$ to an existing set F. In this case, the hyper-rectangle is constructed, bounded at $\mathbf{b}$, from the neighbouring points in each dimension and each vertex of the hyper-rectangle (excluding $\mathbf{y}$) is dominated by a member of F. As such, the additional volume dominated by $\mathbf{y}$ but not F, shown as the dark shaded region, is added to the existing dominated volume v, shown as a light

shaded region. The right half of Figure 22 shows the (recursive) general case. In this case, some vertices of the hyper-rectangle bounded by $\mathbf{y}$ and the attainment surface of F are not dominated by members of F and so each of these non-dominated vertices is added using the same method, resulting in the the addition of the volumes shown hatched, before the $\mathbf{y}$ is added, finally resulting in the dark shaded volume being added.

More formally, this incremental calculation, shown in Algorithm 8, recursively calculates and adds the volumes of space between the front for which the volume is known, and the newly added point. When a new point $\mathbf{y}$ is added, the existing set is checked and if the point is already in the set, or dominated by a member of the set, no action is taken as the new solution does not increase the volume dominated by the set. If $\mathbf{y}$ is not in the set then an axis-parallel hyper-rectangle is constructed; the lower bounding vertex is $\mathbf{y}$, and the opposite bounding vertex is calculated with the use of D lists, each containing the set sorted in one dimension. As shown in lines 4 to 10 of Algorithm 8, these ordered lists, L, are used to locate the nearest neighbour $\mathbf{u}$ of $\mathbf{y}$ in each dimension, i, such that $\mathbf{u}_i > \mathbf{y}_i$; the nearest value which is greater than $\mathbf{y}$ in this dimension. The bound, $\mathbf{b}$, of the hyper-rectangle is constructed from these nearest (greater) neighbours in each dimension. Each vertex of the hyper-rectangle is then calculated, and this can be performed simply, as in lines 11 to 22 of Algorithm 8, through a combination analogous to binary representations; here each combination of the vertices of both $\mathbf{y}$ and $\mathbf{b}$ is generated, which produces all the vertices. The new solution, $\mathbf{y}$, is then removed from the list of vertices, B. The vertices are then sorted in order of dominance (this is not a strict total order; those vertices within a group of mutual non-dominance may be ordered arbitrarily). Each vertex, starting with the least dominating, is then added to the set using the same recursive function. Once the function has recursed over each of the vertices in this manner, the volume of the bounding box is trivially calculated and added to the dominated volume of the set and the point itself is stored in the set.

In the best case (where the initial hyper-rectangle is the only additional volume dominated by the new point), the addition of a point with this method is only $\mathcal{O}(2^D \log(|F|))$ (the nearest neighbour in each dimension can be determined to be dominated in $\mathcal{O}(\log(|F|))$ time using data structures such as PQRS trees). The worst case complexity is $\mathcal{O}(n^D 2^D \log(|F|))$, where the entire set is dominated by the new point as the upper bound on the number of possible recursive subvolumes is n^D; as such, it is worth recalculating the volume completely and not incrementally when a large proportion of

the set is dominated by the new point.

4.4 Comparisons

In this section the performance of MOSA and SAMOSA are compared with each other and with their greedy ($T = 0$) versions. Results on MOSA and SAMOSA give a direct comparison of single solution states against set states, while dominance based and volume based energy measures are compared via the SAMOSA and VOLMOSA algorithms. As displayed in Table 3, the temperature zero versions of the algorithms are denoted by MOSA0 and SAMOSA0.

Performance is evaluated on well-known test functions from the literature, namely the DTLZ test suite problems 1-6 [Deb et al., 2001, 2002c][2]. Results with $D = 3$ objectives are shown for the MOSA and SAMOSA comparisons, but computational expense limits SAMOSA–VOLMOSA comparisons to two objective problems. The benefit of using these test functions is that the true Pareto front $\mathcal{P}$ is known, so the proximity of the estimated Pareto front, F, can be measured.

All annealers used a common annealing schedule. All epochs L_k were of equal length, $L_k = 100$ and the temperature was reduced according to $T_k = \beta^k T_0$, where β was chosen so that T_k is 10^{-5} after approximately two thirds of the function evaluations are completed. 50000 function evaluations were used for each of the problems, except the DTLZ2 problem for which 5000 evaluations were used, due to the ease of convergence of the algorithms on this problem. The MOSA initial temperature was set so that approximately half of the perturbations are accepted, as described in Chapter 3, while for SAMOSA $T_0 = 4$ which results in an initial acceptance rate of roughly 0.5 [3].

Twenty runs from different (random) initialisations were made for each algorithm. Before statistical summaries of the performance on all the test problems are given, results on the individual problems are discussed. In order to present representative results, the median (to reduce the influence of outliers) distance of solutions in F from a single run is calculated, as discussed in Chapter

[2]The formulations of the DTLZ functions presented in Smith et al. [2006] are used because these correct typographical errors in the formulae presented by Deb et al. [2001, 2002c]. The number of parameters used are those recommended by Deb et al. [2002c], namely: DTLZ1 7; DTLZ2 7; DTLZ3 12; DTLZ4 12; DTLZ5 12; DTLZ6 22.

[3]Since the energy change is based on a proportion, the range for detrimental solutions is $0 < \delta E(\omega, \omega') \leqslant 1$. Assuming an average energy change of 0.5, setting T to 4 yields an average acceptance probability of approximately 0.5 for these exploratory proposals at the start of a run.

2:

$$\bar{d}(F, \mathcal{P}) = \underset{\mathbf{x} \in F}{\text{median}}[d(\mathbf{x}, \mathcal{P})] \tag{32}$$

where $d(\mathbf{x}, \mathcal{P})$ is the minimum Euclidean distance between $\mathbf{x}$ and the true front $\mathcal{P}$. The representative run is then chosen as the run with the median $\bar{d}(F, \mathcal{P})$ over the 20 runs. This measure depends on the relative scalings of the objective functions, but here it gives a fair comparison because the objectives for the DTLZ functions all have similar ranges.

4.4.1 DTLZ 1

Figure 23 presents the results for both the greedy and non-greedy versions of both the MOSA and SAMOSA algorithms on the DTLZ1 problem. The true front for the DTLZ1 problem is the plane intersecting the axes at $(0, 0, 0.5)$, $(0, 0.5, 0)$ and $(0.5, 0, 0)$, and with $f_i \geq 0$.

Both the MOSA (non-greedy) and MOSA0 (greedy) algorithms converge well to the true front and provide good coverage of the front. SAMOSA and SAMOSA0 both perform less well than the MOSA variations, not coming so close to the front and not covering it so uniformly. The formulation of the problem means that it is easy to make perturbations which traverse the front, so there is no gain from SAMOSA's approach of simultaneously optimising a set of solutions by making perturbations to many solutions, bringing them all to the true front together. MOSA's approach of optimising a single solution, in this case, results in the front being located more quickly, allowing time to traverse the front providing the good coverage exhibited. Since the estimated front is still converging, it can be expected that, if given sufficient time, SAMOSA would also converge on the true front but that it will require significantly more evaluations. Particularly interesting is that, despite $\approx 10^{11}$ local fronts [Deb et al., 2001, 2002c] arranged parallel to $\mathcal{P}$ in the DTLZ1 problem, the greedy algorithms are able to converge to, or near to, the true front; MOSA0 is only slightly out-performed by MOSA and SAMOSA0 outperforms SAMOSA. This demonstrates that local fronts in this form are not necessarily a barrier to the convergence of greedy algorithms — this is discussed further in section 4.5.

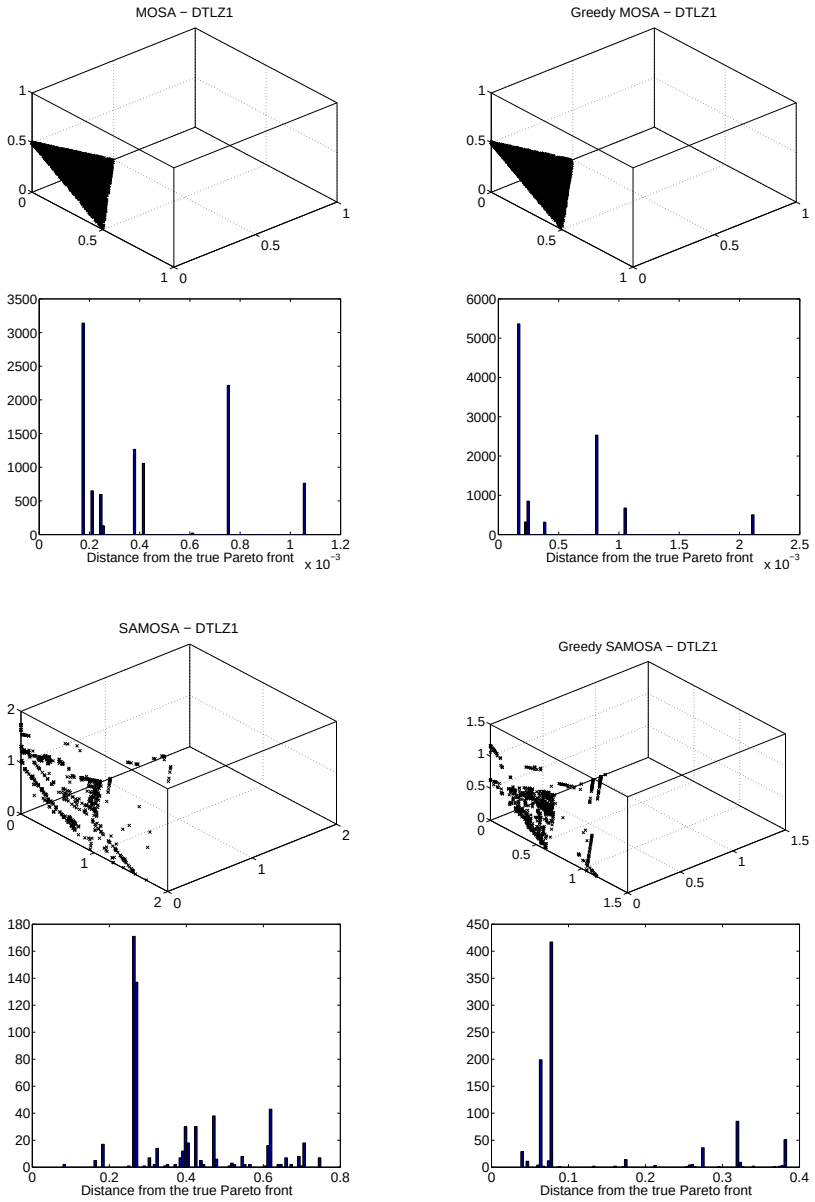

Figure 23: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on DTLZ1. Archives of the median annealer run after 50000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

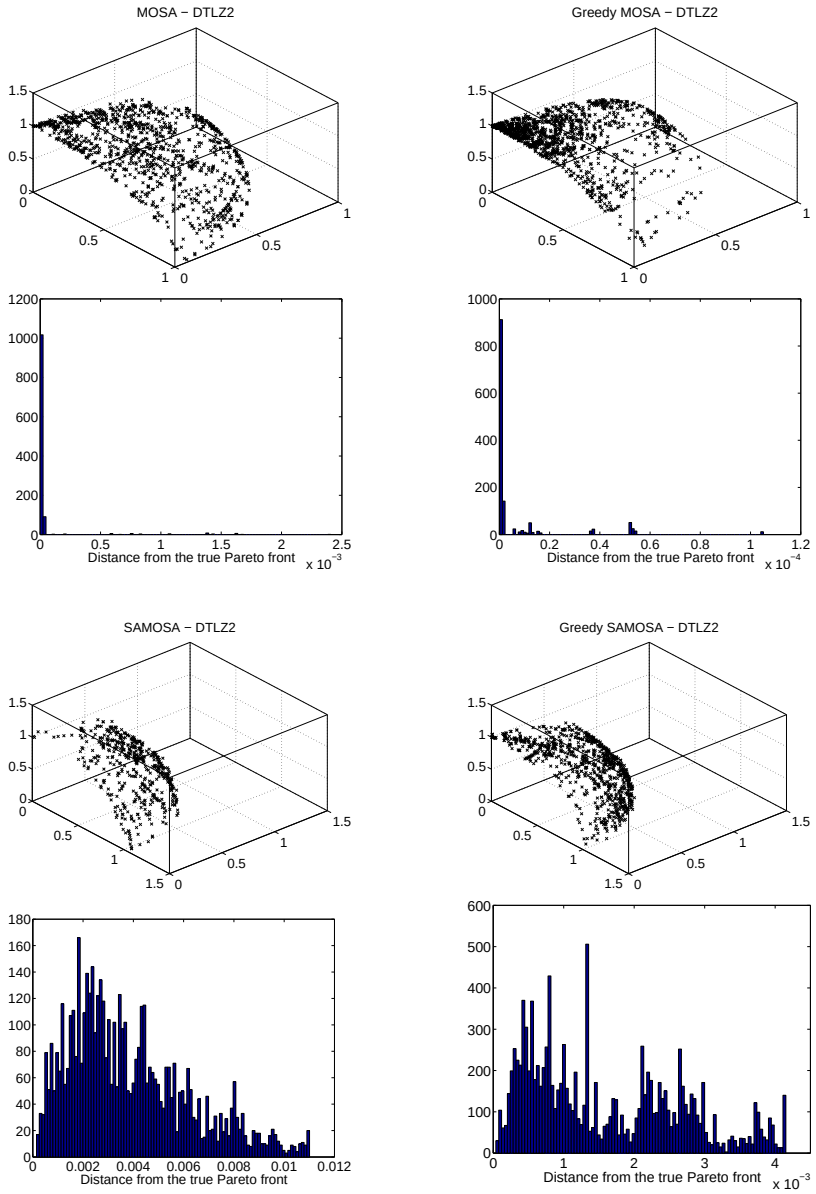

Figure 24: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on DTLZ2. Archives of the median annealer run after 5000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

4.4.2 DTLZ 2

Results for DTLZ2 are presented in Figure 24. Here $\mathcal{P}$ is the positive octant of a sphere of radius 1, centred on the origin. As may be expected on this easy problem, the greedy variants significantly outperform their respective non-greedy counterparts (note the different scales on the histograms). Again the MOSAs outperform the SAMOSAs as the ease of traversal across the front results in redundant perturbations of state members. Put another way: the set-based annealer's potential advantage of being able to perturb solutions in diverse parts of parameter space in order to move the front forward if a single solution is 'stuck' on a local front is of no advantage, and in fact perturbing solutions across the entire front is a waste of computational resources on this problem.

4.4.3 DTLZ 3

The DTLZ3 results, presented in Figure 25, show that even on this problem, with many local fronts on which a greedy optimiser might be expected to become stuck, the greedy algorithms are able to out-perform the corresponding non-greedy variants and converge on the true front, which is again a radius 1 octant of a sphere centred on the origin. Due to the large number of similar perturbations that must be made to move an entire set state SAMOSA is again out-performed by MOSA. Note, however, that the perturbation scheme is local in the sense that only a single parameter is perturbed at a time, but global in the sense that there is a finite probability that a perturbation can move the selected parameter across its entire feasible range.

4.4.4 DTLZ 4

Figure 26 presents the results for DTLZ4. The true front for DTLZ4 is, as for DTLZ2 and DTLZ3, an octant of unit sphere centred on the origin; the formulation of DTLZ4 is such, however, that it is easy for an optimiser to reach one corner of the front, but that the rims represent a very small region of parameter space and that the central region of the front is mapped to yet a smaller region of parameter space (see Chapter 3 for a more detailed analysis). Here the Uniselect method for selecting solutions for perturbation, combined with the exploratory or non-greedy ability to move away from the front, allows SAMOSA to cover the central region of the front well, a small distance away from $\mathcal{P}$, providing far better coverage of, although less close to, the true front than the other algorithms. SAMOSA0 also provides a reasonable coverage of the front, due to the Uniselect selection

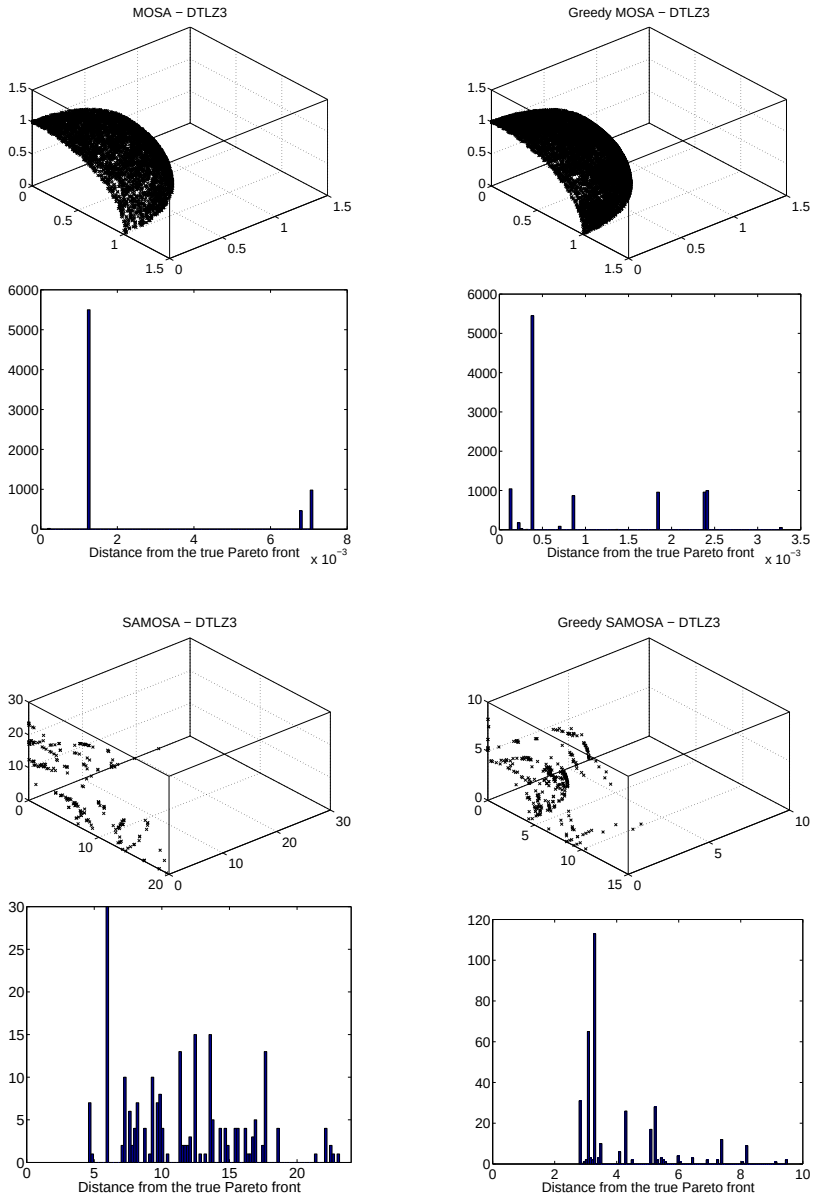

Figure 25: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on DTLZ3. Archives of the median annealer run after 50000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

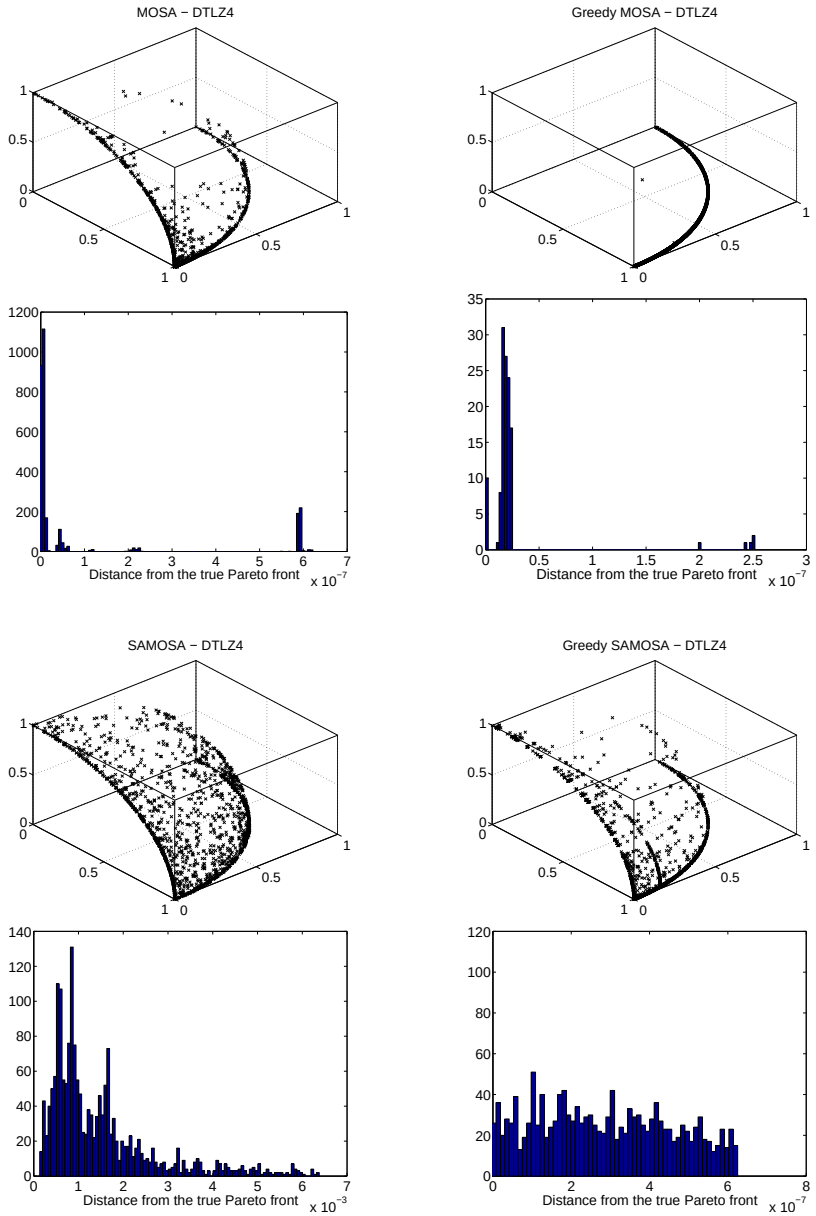

Figure 26: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on DTLZ4. Archives of the median annealer run after 50000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

method. As MOSA uses a single solution for the state, there is no possibility to use a Uniselect method for selecting solutions to perturb evenly across the front but it has still managed to generate a small covering of the front, possibly through non-greedy moves away from the corner. The MOSA0 variant, having neither Uniselect nor the ability to make non-greedy moves, was unable to leave one of the rims of the front but was still able to converge to within a small distance from the true front for the regions it covered.

4.4.5 DTLZ 5

All four algorithms were able to converge well to the true front for DTLZ5, the results of which are presented in Figure 27. The true front in this case is a one-dimensional arc, rather than a two-dimensional surface. Uniselect allowed SAMOSA and SAMOSA0 to generate a more even coverage of the front while MOSA and MOSA0 converged closer to the true front.

4.4.6 DTLZ 6

Figure 28 presents the results for the DTLZ6 problem, the true front for which consists of four disjoint regions or 'cushions' in objective space, to which all four algorithms were able to converge well. Note that MOSA and MOSA0 whose single solution states might be expected to limit them to a single cushion are able to locate all four regions; if the regions in parameter space corresponding to each of the objective space cushions were more widely separated then it might be more difficult for the MOSA and MOSA0 annealers to locate all the cushions. The Uniselect selection seems to have enabled SAMOSA and SAMOSA0 to generate a more uniform coverage of the four cushions, while MOSA and MOSA0 converged closer to the true front, and generated more solutions in their archives possibly due to the 'wasteful' effect of making similar perturbations across a range of solutions, as observed on DTLZ2.

4.4.7 Summary

Statistical summaries of these results are presented in Figure 29 as box plots showing the distributions over 20 runs of the median frontal distance $\bar{d}(F, \mathcal{P})$ from the true Pareto front and the proportion of the volume of the bounding box of $\mathcal{P}$ that is dominated by $\mathcal{P}$ but not by F. Note that the volume measure is denoted by $\mathcal{V}(\mathcal{P}, F)$ in order to make connection with other work, but it may

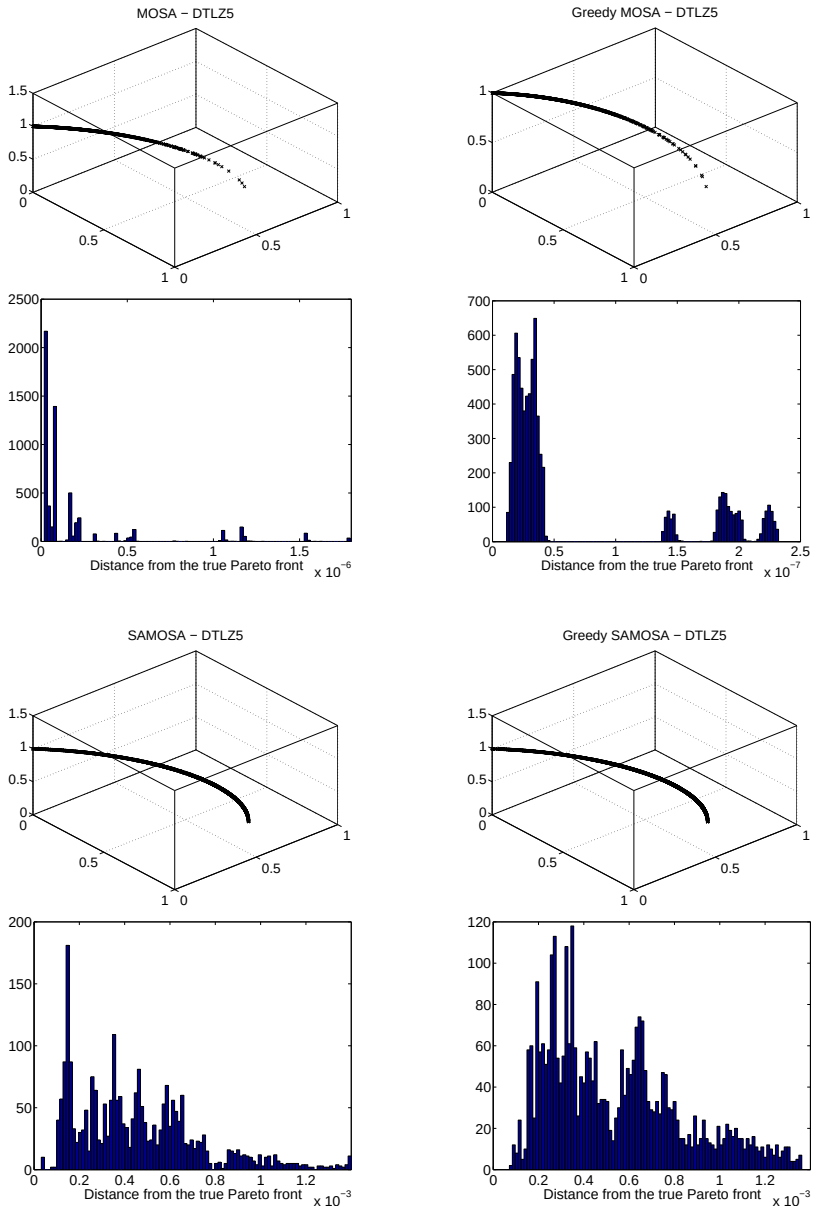

Figure 27: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on DTLZ5. Archives of the median annealer run after 50000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

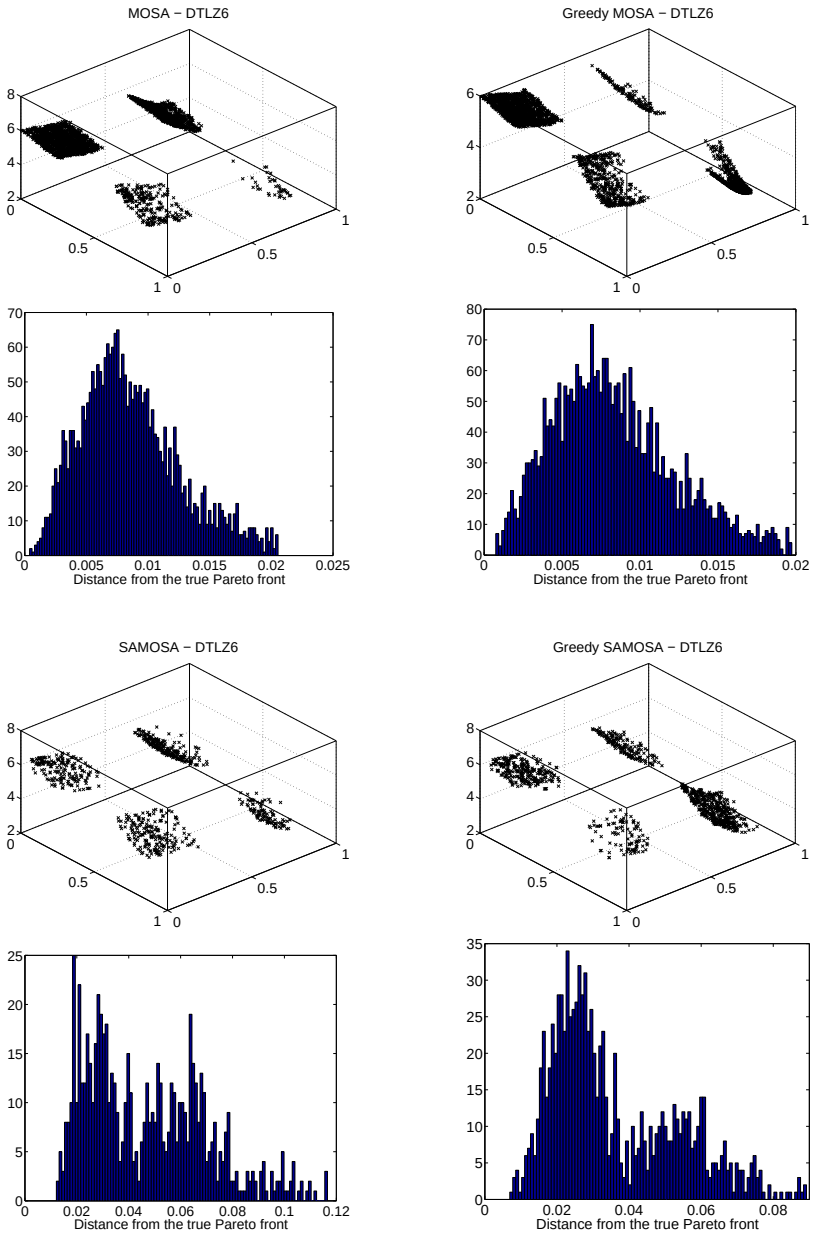

Figure 28: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on DTLZ6. Archives of the median annealer run after 50000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

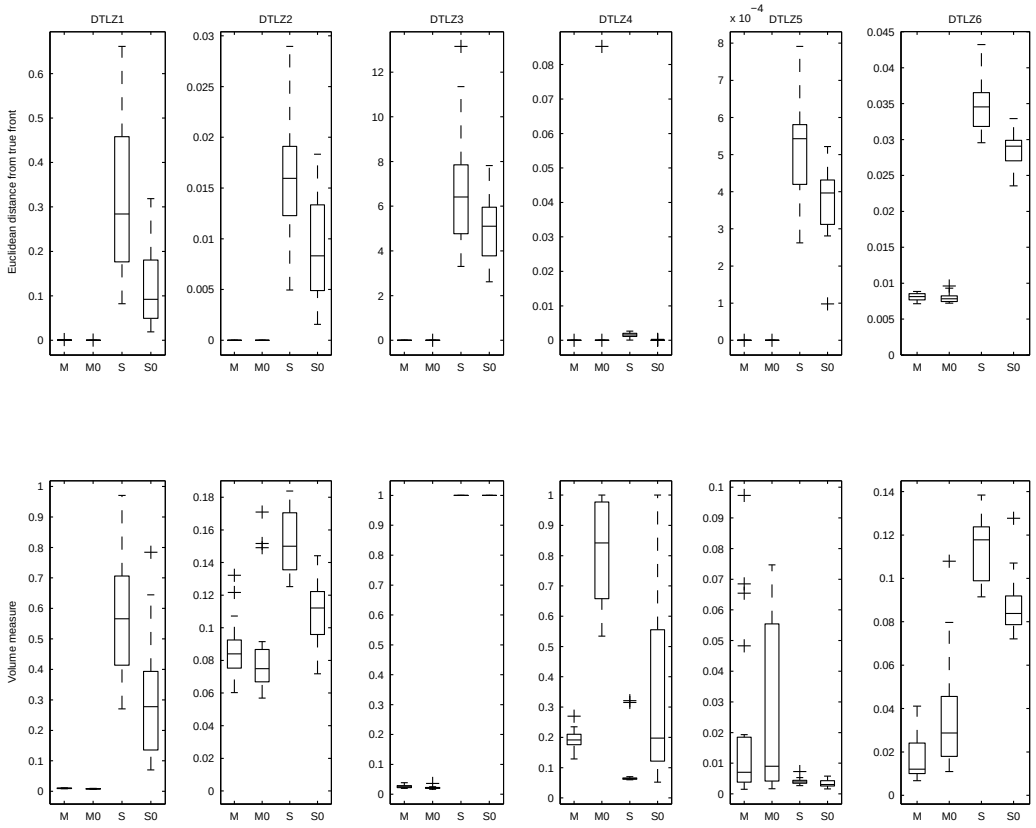

Figure 29: **Top:** Box plots showing the distribution over 20 runs of the median distances $\bar{d}(F, \mathcal{P})$ of the front from the true Pareto front. 'M', 'M0', 'S' and 'S0' denote the MOSA, MOSA0, SAMOSA and SAMOSA0 algorithms respectively. **Bottom:** Box plots showing the distribution over 20 runs of the proportion of the bounding box of the Pareto front dominated by the true front but not the archives.

be recognised as being proportional to the energy $E(\mathcal{P}, F)$, used in the VOLMOSA annealer. It was calculated by Monte Carlo sampling (10^5 samples) rather than via the incremental algorithm (section 4.3.3).

These results in Figure 29 primarily reveal three interesting traits of the algorithms and problems. The first result, common across these problems, is that the MOSA and MOSA0 algorithms outperform SAMOSA and SAMOSA0 in their ability to generate solutions close to the Pareto front. This result may be understood as follows: SAMOSA may advance the state by perturbing each of the constituent solutions to move the ω incrementally forward, however, if a subsequent perturbation to

a solution dominates several other solutions then these solutions are replaced in ω by the dominating solution and the perturbations that have advanced them thus far are wasted; this does not occur for MOSA which focuses on a single solution. It appears from these results that SAMOSA's advantage of being able to perturb solutions in several locations is outweighed by these wasted perturbations. This implies that, for algorithms similar to those used here, it takes an algorithm which perturbs solutions selected from a set longer to reach the Pareto front than an algorithm which repeatedly perturbs a single solution. In contrast to this, however, the SAMOSA and SAMOSA0 algorithms were able to generate fronts with a more uniform coverage in objective space than MOSA and MOSA0, particularly on the DTLZ4 problem and observable to a lesser extent on the DTLZ5 and DTLZ6 problems. The Uniselect method which is used to select solutions for perturbations from regions of the front uniformly, without prejudice to the density of solutions in those regions, maintains a more uniform coverage within the estimated front.

The most surprising comparisons are those between the greedy and non-greedy algorithms. The DTLZ1 and DTLZ3 problems were specifically constructed with the characteristic of many local fronts [Deb et al., 2001, 2002c], designed such that greedy algorithms may become unable to escape them; nonetheless the greedy algorithms are able to outperform the non-greedy algorithms on these problems. It is clear that for many problems, even those with local fronts, greedy algorithms are sufficient; this behaviour is investigated further in section 4.5 and a problem where this is not true is investigated.

4.4.8 Comparing Volume and Dominance Measures

Due to the complexity of the volume calculations it is necessary to use two objective versions of the problem formulations[4] and a limited number of evaluations for the comparison of the dominance-based energy measures used in SAMOSA and the volume-based energy measures used in VOLMOSA. As the behaviour of the $T = 0$ algorithms are very similar, they are not shown here. Results are presented on the DTZL1-4 test problems, after 30000 function evaluations for each of DTLZ1, DTLZ3 and DTLZ4 and after 5000 evaluations on DTLZ2 due to the ease of convergence on that problem. The same methodology was used here as for the three-objective comparisons, with 20 runs of each configuration and selection of the median run. Again, an annealing schedule was chosen so as to

[4]The number of parameters P was chosen as recommended by Deb et al. [2002b] for two-objective formulations, namely: DTLZ1 6; DTLZ2 6; DTLZ3 11; DTLZ4 11.

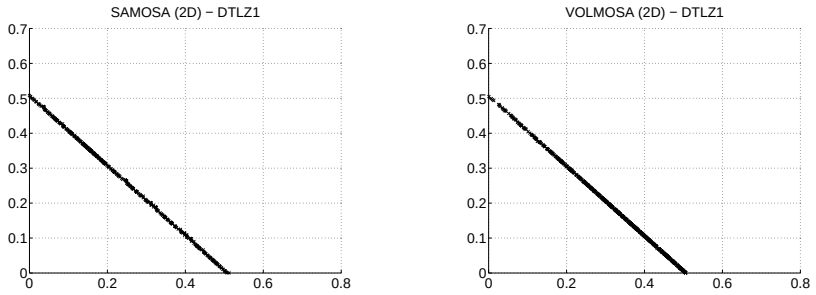

Figure 30: Archives of the DTLZ1 problem after 30000 function evaluations of the SAMOSA and VOLMOSA annealers.

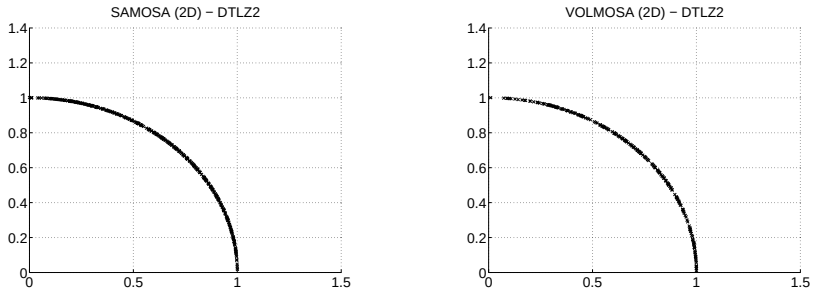

Figure 31: Archives of the DTLZ2 problem after 5000 function evaluations of the SAMOSA and VOLMOSA annealers.

reduce from $T_0 = 4$ initially to $T = 10^{-5}$ after approximately two thirds of the evaluations.

Figure 30 shows the results of SAMOSA and VOLMOSA on the two-objective formulation of DTLZ1. The results show that the two energy functions have a similar performance on this problem, both producing estimated fronts within 10^{-2} of $\mathcal{P}$ on average.

The results for VOLMOSA and SAMOSA on the DTLZ2 test problem are presented in Figure 31. Both algorithms converge close to $\mathcal{P}$ (each set resides approximately 10^{-3} from $\mathcal{P}$).

Figure 32 shows the archives produced by SAMOSA and VOLMOSA on DTLZ3. Both algorithms produced archive sets close to $\mathcal{P}$. This demonstrates that the DTLZ3 problem is significantly easier in its two-objective formulation than the usual three-objective one, as SAMOSA was previously shown to have difficulty converging to the true front in a greater number of function evaluations in the three-objective formulation.

The archives produced by SAMOSA and VOLMOSA for the DTLZ4 problem are presented

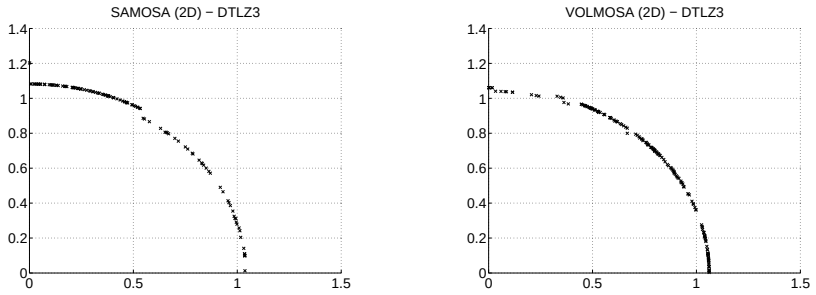

Figure 32: Archives of the DTLZ3 problem after 30000 function evaluations of the SAMOSA and VOLMOSA annealers.

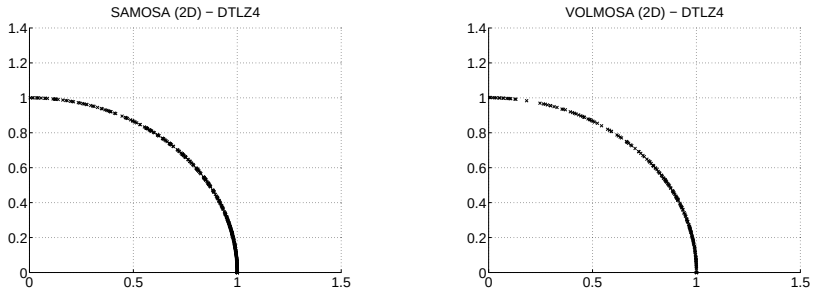

Figure 33: Archives of the DTLZ4 problem after 30000 function evaluations of the SAMOSA and VOLMOSA annealers.

in Figure 33. In common with the other test problems, both VOLMOSA and SAMOSA produce similar, well converged, archives.

As suggested by the results shown here, the VOLMOSA and SAMOSA energy functions yield very similar results. This is confirmed by the box plots shown in Figure 34, which show no significant difference between the performance of the two algorithms. It thus appears that either energy function is equally effective with a set-based multi-objective simulated annealer. Unfortunately the computational complexity of the dominated volume calculation is prohibitive and consequently a more thorough investigation cannot be performed; this also suggests that it may be unsuitable for real-world problems of more than two objectives unless the complexity of the objective evaluations is so great that only a very few evaluations are to be performed. Additionally, as the volume measure uses objective results directly, it may be sensitive to the objective scales in a similar, although less pronounced, way to composite objective functions; this will change the rate of convergence but not

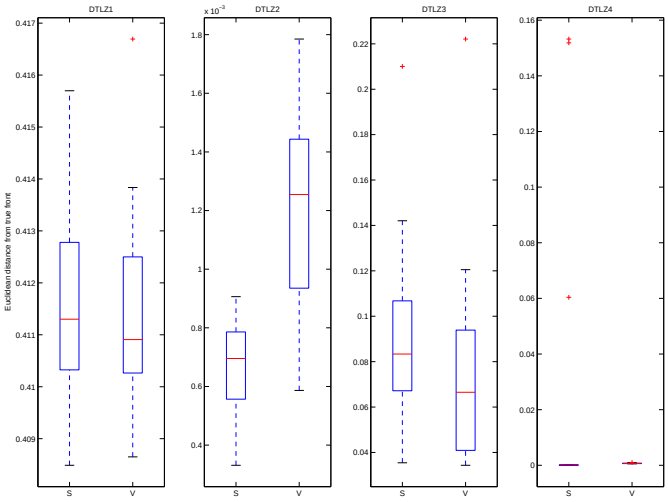

Figure 34: Box plots showing the distribution over 20 runs of the median distances $\bar{d}(F, \mathcal{P})$ of the front from the true Pareto front. 'S' and 'V' denote the SAMOSA and VOLMOSA algorithms respectively.

the ability to converge eventually.

4.5 Greedy Search and Local Fronts

4.5.1 Overview of Greedy Search

The DTLZ1 and DTLZ3 problems present an optimisation algorithm with many local fronts within which it is possible for a greedy search algorithm to become trapped, yet it is clear from the results presented earlier that the presence of local fronts is not sufficient to prevent a greedy search technique from optimising a problem. Further study with the algorithms presented previously in this chapter reveals that a similar situation is found in the WFG problems [Huband et al., 2005]; problems with many local fronts do not necessarily cause greedy algorithms to become 'stuck'. It is not immediately clear, however, why this should be; the presence of a local front implies that a solution cannot travel through the front (in objective space) by continuing on the same vector that caused it to reach the front (in parameter space). Indeed the notion of local fronts suggests that the search must

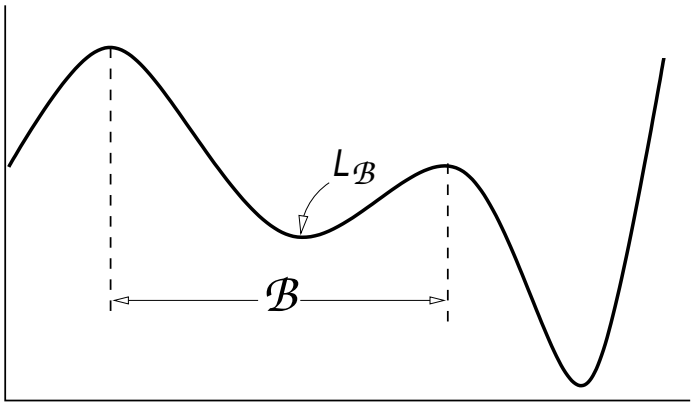

Figure 35: Basin $\mathcal{B}$ and local minimum $L_{\mathcal{B}}$ of a single objective, single parameter optimisation problem.

move away (in objective space) from the global Pareto front in order to eventually reach the global optimum. This section discusses the ability of a greedy optimiser to pass through these local fronts and presents new local-fronted problems on which greedy optimisers perform less well.

Uni-objective simulated annealing is a popular technique due to its ability to 'escape' from local minima; it is able to move to areas of less desirable solutions than that which it currently explores, so that it can eventually locate the global optimum. A greedy search algorithm is unable to make such movements, as it is only able to move to lower energy states. This is sometimes thought of as meaning that, given a simulated annealer and a greedy optimiser starting from the same local minimum, the simulated annealer will be able to escape by moving up the gradients which bound the minimum, while the greedy algorithm will not. This is not, however, necessarily the case, as optimisation algorithms are not restricted to infinitesimal perturbations. As such, where the basins surrounding minima are small, or the feasible perturbations are large, perturbations which escape the basin entirely are possible, falling within another region of parameter space. For sufficiently complex optimisation problems, with large basins relative to the possible perturbations, it could be that it is not possible, or is very unlikely, that perturbations covering a sufficiently large distance in parameter space will be generated.

Unlike uni-objective optimisation problems, where the concepts of local minima and basins are simple, the concepts of local fronts and the associated basins in multi-objective test problems are not necessarily intuitive. The search space in multi-objective problems, like uni-objective problems,

may be partitioned into a set of disjoint sets or basins. A basin B is defined as the maximal region of parameter space such that for every solution $\mathbf{x}$ within $\mathcal{B}$, infinitesimally close to $\mathbf{x}$ there is another solution $\mathbf{x}_\delta$ which is dominated by $\mathbf{x}$. Thus B is the maximal region such that $\forall \mathbf{x} \in \mathcal{B} \exists \mathbf{x}_\delta s.t. \mathbf{x} \prec \mathbf{x}_\delta$. The local front, $L_\mathcal{B}$, corresponding to this basin is the non-dominated subset of the basin. It can be seen that this extension of definition to the multi-objective spaces encompasses the single-objective definition, as illustrated in Figure 35. In single-objective problems, the dominates relation is equivalent to the less-than relation, such that $\mathbf{x} \prec \mathbf{x}_\delta$ and $\mathbf{x} < \mathbf{x}_\delta$ are equivalent, and this definition of basins clearly applies to the single-objective case. In single-objective problems, the non-dominated subset of a basin must be a single point and as such the local front is a single point, the local minima (unless B has a flat bottom, in which case several points share identical, minimal, energies).

Deb introduces similar definitions [Deb, 1999] for a 'Local Pareto-optimal Set'; the definitions used here differ slightly from those used by Deb in order to clarify the distinction between fronts and basins.

A problem is said to be *infinitesimally greedy-searchable* if, for any solution in parameter space, a series of infinitesimally small perturbations may be made to it such that the limit of these perturbations is a solution in the Pareto set and no solution in the sequence of perturbations is dominated by a predecessor. If the sequence of perturbations from a solution to a Pareto-optimal solution must contain a perturbation which is dominated by a predecessor then an infinitesimal greedy search would be unable to accept it, recognising it as an inferior solution.

It should be noted that a greedy-search algorithm (one which will not replace good solutions with inferior solutions) will make finite, possibly large, perturbations and the concepts of a greedy-search algorithm and infinitesimal greedy-search must be kept distinct. In the extreme case, a greedy-search algorithm allowed unbounded perturbations may make a single perturbation from any solution to the Pareto front. While this means that a real greedy optimiser may be able to locate the Pareto front even if the problem is not infinitesimally greedy-searchable, the computational effort required is equivalent to an exhaustive search. The MOSA0 and SAMOSA0 greedy search algorithms are able to locate the Pareto front on the DTLZ problems, despite the presence of local fronts, because the perturbations required to escape these local fronts are relatively small.

Table 4: INGS1 and INGS2 problem definitions. There are D objectives and P decision variables, $0 \leq x_i \leq 1$. The parameter Q sets the number of regions.

INGS1(Q)	
$r =$	$Q\left[\frac{1}{P-D}\sum_{i=D+1}^{P} x_i^2\right]^{\frac{1}{2}}$
$h =$	$r - \lfloor r \rfloor + 1 - \text{remainder}(\lfloor r \rfloor, 2) + 2\lfloor r/2 \rfloor$
$R =$	$h + \lfloor h \rfloor$
$f_i(\mathbf{x}) =$	$R + (1 - x_{D-i}) \prod_{i=1}^{D-i-1} x_i$

INGS2(Q)	
$s =$	$\sqrt{\sum_{i=1}^{D} x_i^2}$
$r =$	$2Q\left[\frac{1}{P-D}\sum_{i=D+1}^{P} x_i^2\right]^{\frac{1}{2}}$
$f_i(\mathbf{x}) =$	$\left[\frac{1}{2}\left(r - \lfloor r \rfloor + 1 - \text{remainder}(\lfloor r \rfloor, 2)\right) + 1 + \lfloor r/2 \rfloor\right] \frac{x_i}{s}$

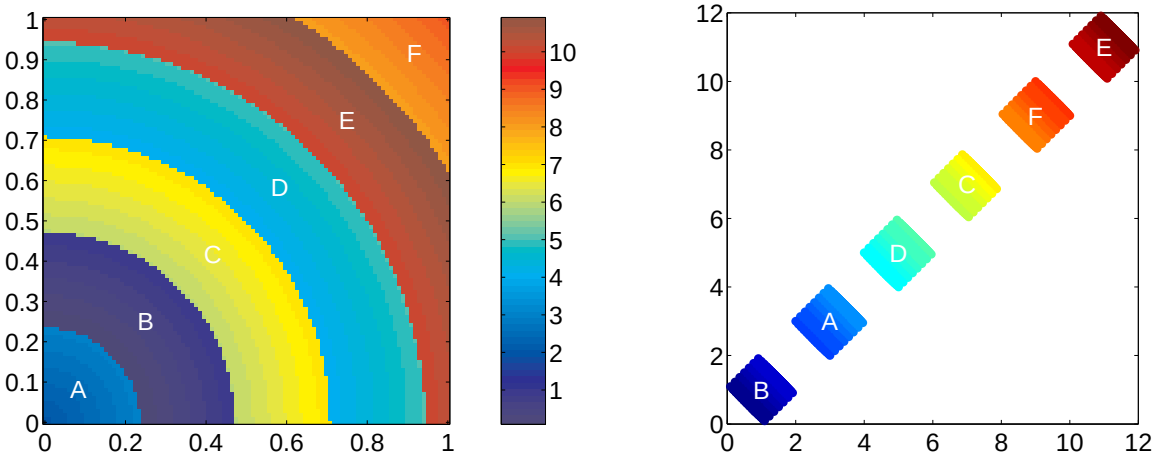

Figure 36: The INGS1 for $Q = 6$ problem mapping between parameter and objective space for four parameters and two objectives. **Left:** Parameter-space for the two parameters corresponding to the distance from the true front $\mathcal{P}$. **Right:** Objective space – the white regions are not feasible. In both images the colour indicates the distance from $\mathcal{P}$ in objective space.

4.5.2 A Non-Infinitesimally Greedy Searchable Problem (INGS1)

When constructing test problems demonstrating the ability of algorithms to escape local minima it is crucial to ensure that the basins are sufficiently large in parameter space that it is very unlikely,

or impossible, to make a single perturbation which escapes them.

The addition of multiple objectives makes the inclusion of local minima in a test problem more complex, as there must now be regions, rather than points, in parameter space which correspond to locally minimal fronts in objective space, rather than locally minimal points, as is the case in single-objective constructions. The creation of these regions can be trivially performed, however, through the use of separate 'distance' and 'location' parameters, such that the location parameters determine the position of a solution upon the current front, and the distance parameters determine which the local front is, and how far from it the solution lies. Using this technique, local minima may still be discrete points in the space of the distance parameters, making construction easier. It also allows separation of the front shapes from the complexity of the number and sparsity of the local fronts. This technique has been used in the DTLZ [Deb et al., 2001, 2002c] and the WFG [Huband et al., 2005] problems and this approach is used for the problem definitions in this chapter.

Table 4 defines an INGS1 problem, illustrated in Figure 36. This problem is constructed in the same manner as the DTLZ and WFG problems, with separate parameters determining the region of the Pareto front the solution lies above, and the distance to the Pareto front. The first D parameters are used for the location of solutions on the Pareto front, and the remaining $P-D$ decision variables are used to set the distance from the true front. Objective space consists of Q distinct regions, which are labelled in Figure 36 as A to F. The radius, r, in parameter space is calculated first. The fronts in objective space are then constructed by the simple R term. These fronts are then separated and rescaled appropriately (the reason for the separation is to ensure that each discrete region completely dominates, or is completely dominated by, each other region; in Figure 36, all members of region A are dominated by all members of region B. This is explained in the later description of the INGS2 problem). These fronts can be seen in Figure 36: one front is the image in objective space of the boundary of A and B in parameter space, another at the boundary of C and D *etc.*; the Pareto set is the boundary of A and B. The radius of each region in parameter space which maps to a single region of objective space is $1/Q$ in parameter space, and the corresponding objective space region has a width of 1. There are $\lceil Q/2 \rceil$ basins in which it is possible for a greedy search algorithm to become trapped on a local front (not every region corresponds to a complete basin). Q is treated as a parameter of the INGS1 problem and INGS1(6) is used to denote the INGS problem where $Q = 6$.

The INGS1 problem is not infinitesimally greedy searchable, as it is not possible to escape the

$\lceil Q/2 \rceil$ basins which contain only locally optimal fronts using infinitesimally small perturbations to a solution within a basin. Figure 36 illustrates the mapping from parameter space to objective space for the INGS1(6) problem. Here it can be seen that every region of parameter space (excluding the extremal regions), shown in the left image, is adjacent to two other regions but that it is only adjacent in objective space (shown in the right image) to one of these two regions; it is this feature of the mapping which provides local fronts for this problem. The local fronts for this problem are located in parameter space at the boundary of two regions which are not adjacent in objective space.

It should be emphasised that optimisation algorithms such as those discussed here do not make infinitesimally small perturbations and that, as such, a greedy optimiser may still converge upon such a problem which is not infinitesimally greedy searchable due to the presence of local fronts, as is the case with the DTLZ and WFG problems. The ability of a greedy optimiser to converge is determined by the size of perturbation possible and the size of perturbation required to escape a basin to a dominating basin. In Figure 36 for example, a perturbation from D to B brings the solution closer to $\mathcal{P}$, but a smaller perturbation from D to C is detrimental. Therefore by setting Q to be sufficiently high (and as such the size of the basins sufficiently small) it may be easy for greedy optimisers to converge upon the true front. It may be expected that the reason greedy optimisers converge for the DTLZ multi-fronted problems is that the $\approx 10^{11}$ fronts (for DTLZ1) correspond to regions of parameter space over which a perturbation may easily jump.

Despite algorithms using large perturbations, INGS1 is difficult to converge to for algorithms which perturb either a single parameter, or many parameters at a time. Assume, for simplicity, that parameters are constrained to $0 \leq \mathbf{x}_i \leq 1$. For parameter perturbation schemes able to make perturbations that cross the entire range of m parameters simultaneously, the largest possible movement, in parameter space, per perturbation is $\sqrt{m}$ and the movement required to escape a basin is at least $\frac{\sqrt{P-D}}{\lceil Q/2 \rceil}$. Consequently, setting $\sqrt{m} < \frac{\sqrt{P-D}}{\lceil Q/2 \rceil}$ ensures that non-greedy moves are needed for an m-parameter perturbation algorithm to converge on the true front. While it is initially possible to bypass fronts by performing several very large perturbations to some parameters, when each parameter becomes close to the optimal value it is necessary to make increasingly accurate moves to overcome the local fronts, because the annular regions of parameter space corresponding to basins decrease in area/volume as $\mathcal{P}$ is approached. A greedy algorithm is thus reduced to what is effectively an exhaustive search, while an exploratory algorithm may move between basins.

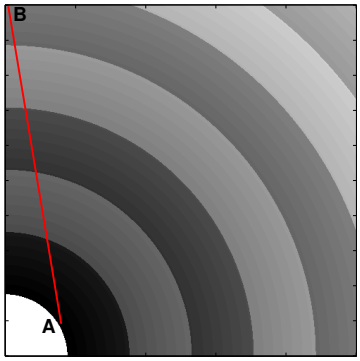

Figure 37: An example of multiple fronts in objectives space. The red line illustrates a sequence of infinitesimal movements, each of which is mutually non-dominated but which starts at a point on $\mathcal{P}$ (A) and ends on a local front far from $\mathcal{P}$ at point B.

4.5.3 A Similar, but Infinitesimally Greedy Searchable Problem (INGS2)

At first sight, the INGS2 problem, defined in Table 4, appears to not be infinitesimally greedy searchable, as it uses a very similar formulation to the earlier INGS1 problem but this is not, however, the case. The reason that INGS2 is susceptible to an infinitesimal greedy search is that, unlike INGS1, there exists a solution $\mathbf{x}$ within a basin from which a new solution $\mathbf{x}'$ may be generated through an infinitesimally small perturbation which leaves the basin, but where it is not true that $\mathbf{x} \prec \mathbf{x}'$, nor $\mathbf{x}' \prec \mathbf{x}$. This is because the basins in INGS2 are continuous in objective-space; this is not true in INGS1 which has discrete basins in objective space. In INGS1 when comparing two basins $\mathcal{B}_1$ and $\mathcal{B}_2$, either all members of $\mathcal{B}_1$ dominate all members of $\mathcal{B}_2$ or vice versa. When basins entirely dominate other basins, it is not possible to make a non-dominated move between them, as it is in the INGS2 problem. An example of this is illustrated in Figure 37. This figure shows several curved fronts in *objective space*, similar to those of the INGS2 problem. It is clear from the red line that it is possible to make a series of movements from the point A on $\mathcal{P}$ to the point B many fronts away from $\mathcal{P}$ without any point on this line being dominated by any other (for simplicity in this figure, a path is shown which requires non-infinitesimal perturbations – a similar path may be traced with infinitesimal perturbations but it is less intuitive). A similar path may also be traced for the INGS2 problem – although for it to be a single continuous path in *parameter* space it consists of

several discrete arcs in *objective* space, no point on any arc segment dominates any point on another arc segment. Please note that, despite visual similarities to the *parameter* space image in Figure 36, Figure 37 represents *objective* space and the shading is used purely to illustrate the presence of fronts (for this reason, the shading is in greyscale here and not in colour, as in Figure 36).

As it is anticipated that the INGS2 problem is difficult for greedy-searching algorithms to converge to, despite being infinitesimally greedy-searchable, this makes it particularly appealing for demonstrating the benefits of exploratory moves and as such, comparative results are presented here.

Since the number of fronts has a great effect upon the properties of the INGS2 problem, it is interesting to present results on two instances, both of which are configured such that it is possible (although not necessarily likely) for the algorithms to generate perturbations which escape basins : the first, INGS2(10^6), has $Q = 10^6$ while the second, INGS2(10), has $Q = 10$. Both INGS2(10^6) and INGS2(10) have $P = 20$ The INGS2(10^6) problem provides a landscape which is easy to traverse for a greedy optimiser, since even very small perturbations to a single parameter can move a solution across many basins without impediment. The INGS2(10) problem provides a landscape which is difficult (although not impossible, as previously noted) for an infinitesimally greedy optimiser to traverse, since very large perturbations are required to move a solution from a local minimum, over the surrounding region of greater energy, to a dominating point. A single parameter perturbation is used, and in both formulations $\sqrt{m} > \frac{\sqrt{P-D}}{\lceil Q/2 \rceil}$, meaning that it is feasible to optimise the problems without resorting to exploratory moves.

INGS2 results

Results for both MOSA and SAMOSA, with an annealing schedule and fixed at $T = 0$ are shown in Figures 38 and 39. In this case, all 20 runs of each algorithm were started from the same distant initial solution. The results for INGS2(10^6), in Figure 38 show that the greedy algorithms (the annealers set to $T = 0$) outperformed their corresponding non-greedy counterpart (all algorithms were initialised with a solution residing on the local front with radius approximately 10^6). MOSA and MOSA0 dramatically outperform SAMOSA and SAMOSA0, in common with the results for the DTLZ3 problem; this is again probably because the large number of solutions available in the state to perturb in SAMOSA means that it is likely for similar moves to be made to several solutions in

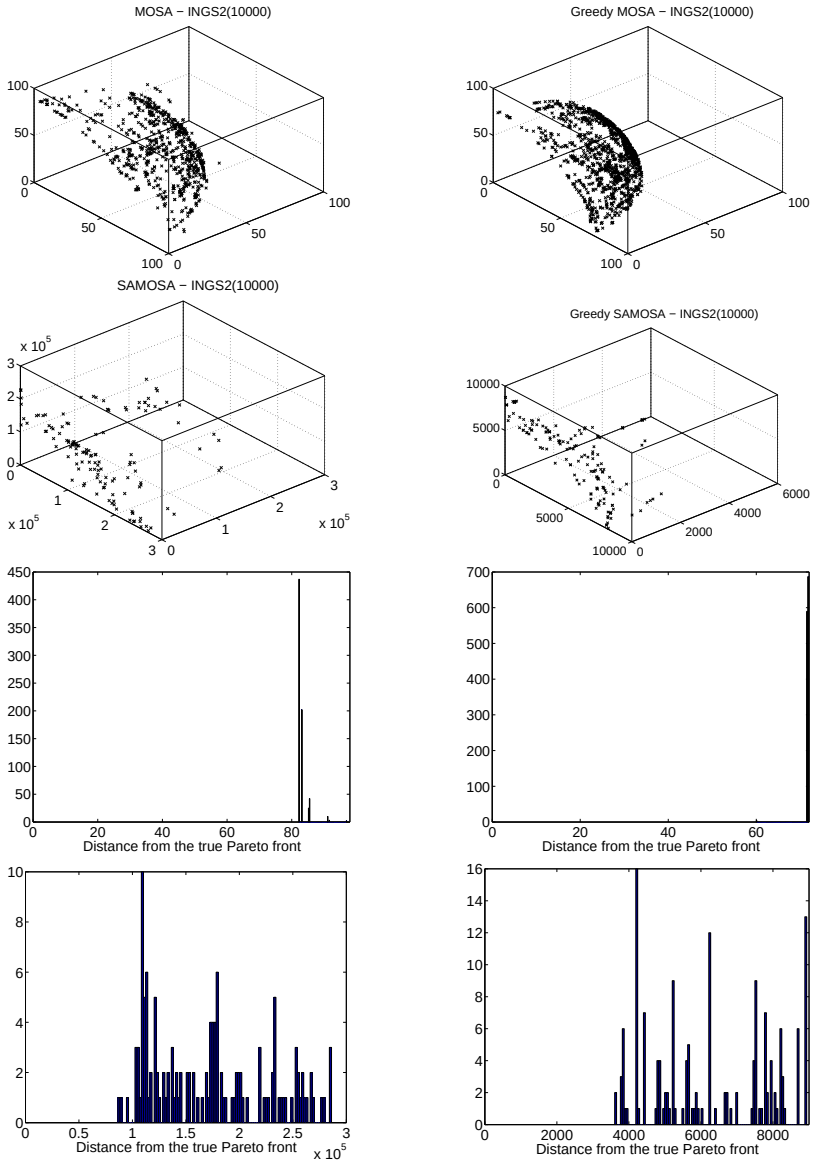

Figure 38: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on INGS2(10^6). Archives of the median annealer run after 50000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

the set, pushing individual solutions forwards while not advancing the state as a whole. Figure 39 shows the results for the algorithms on the INGS2(10) problem, for which the initial solution for

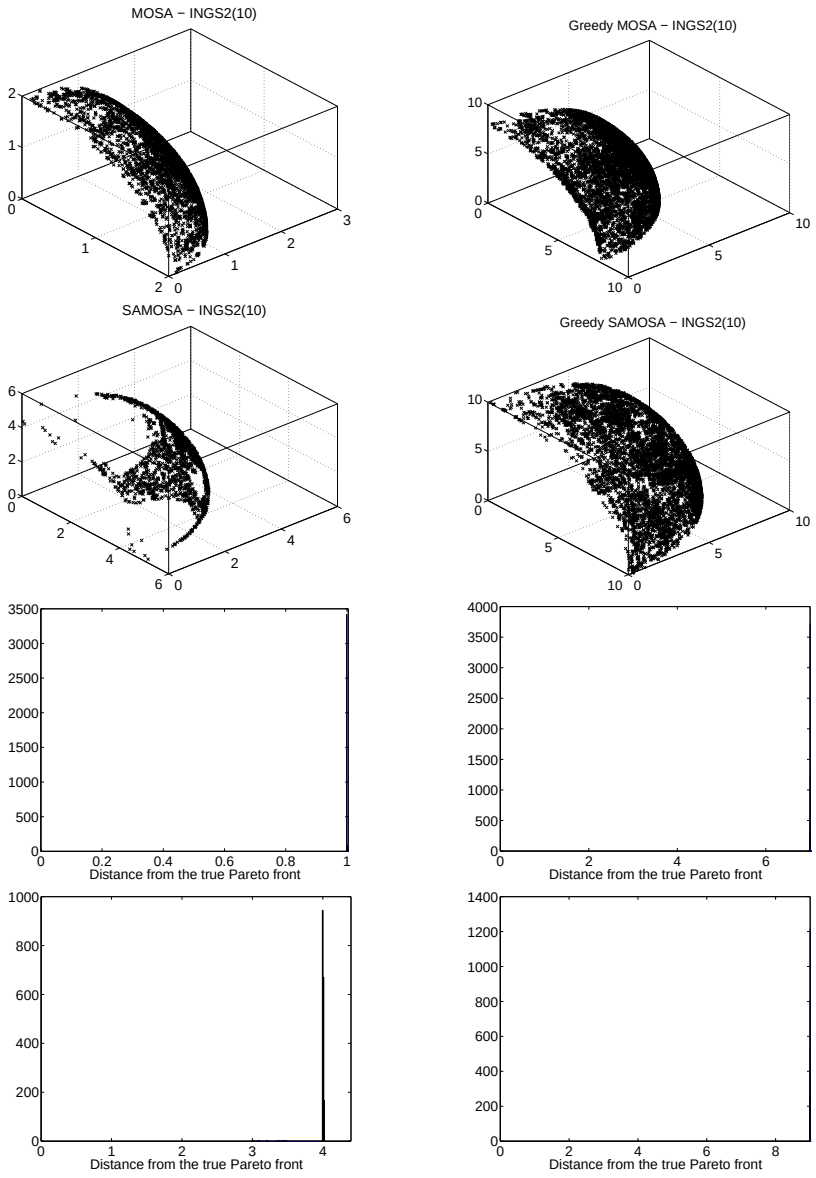

Figure 39: Performance of MOSA, MOSA0, SAMOSA and SAMOSA0 on INGS2(10). Archives of the median annealer run after 50000 function evaluations. Each pair of plots shows the median estimated archive and a histogram of the distances of members of F from $\mathcal{P}$ (the 5% most distant have been omitted to aid visualisation).

each algorithm resided upon a local front with radius 10. On this, more difficult, problem it can be seen that the greedy algorithms have difficulty leaving the initial front with (on the median run)

SAMOSA0 being unable to escape the first local front and MOSA0 being able to skip a single front to find a local front at radius 8. In contrast, the non-greedy exploratory algorithms were able to jump several fronts on the median run; indeed, on some runs, MOSA was also able to converge to the true front.

These results demonstrate that there exist problems for which it is preferable to use a non-greedy technique to converge upon the true front, even though it is possible to reach $\mathcal{P}$ through essentially exhaustive search with a greedy algorithm, and that, in the case of the INGS2 test problem proposed here, multi-objective simulated annealing techniques provide a method for optimisation. Additionally, the INGS2 problem is difficult for a greedy algorithm to converge upon when it is used with a small Q, despite being infinitesimally greedy-searchable. Although INGS2 is infinitesimally greedy searchable, the path of mutually non-dominated solutions which must be taken to escape fronts is sufficiently unlikely that any perturbation made by a greedy algorithm which escapes a front is likely to be through a single perturbation completely bypassing a basin (although these are also very unlikely, as can be seen in the results for the greedy algorithms) and similar results can be obtained on the INGS1 problem as those presented here for INGS2.

4.6 Conclusions

Several approaches to multi-objective simulated annealing have been presented, with a range of properties, optimising single solutions or sets of solutions, and evaluating the solution qualities through measures of dominance or volume. To enable the calculation of dominated volumes more efficiently, an iterative method was proposed, enabling the cost of evaluation to be drastically reduced when the volume dominated by a known subset has been previously calculated. An extensive comparison is presented of these algorithms which establishes the properties of each configuration and the traits of problems for which they are suited. As the DTLZ test problems [Deb et al., 2002c], currently predominant in the literature do not exhibit traits which fully distinguish between greedy and non-greedy algorithms and the newer WFG suite [Huband et al., 2005] behaves similarly, new test problems were presented, the configuration of which determines the relative difficulty of convergence for a greedy algorithm, when compared against non-greedy techniques.

Results on the DTLZ suite of test problems suggest that greedy searchable problems are a larger subset of all problems than might be thought and greedy algorithms similar to PAES or (1+1) ES

are thus suprisingly effective. While it has previously been thought [Deb et al., 2001, 2002c] that local fronts in a problem make it difficult for greedy optimisation techniques to converge to the true Pareto front, as they will become stuck upon the local fronts, it seems that this is not necessarily true. Additionally, while it is intuitively true that providing more local fronts makes it increasingly difficult for a greedy optimiser to converge, as it provides more locations in which one may become trapped, the opposite is true as increasing the number of fronts decreases their size in parameter space, thereby increasing the chance of a perturbation being made which escapes them. Depending upon the configuration of parameter space, it is even possible for local fronts in objective space to provide no obstacle to convergence at all – even for algorithms that make infinitesimal perturbations.

An investigation into the properties of problems has determined that there are two properties which determine whether a problem may be optimised by a greedy search algorithm. The first is whether a series of infinitesimally small perturbations can be made which move a solution through parameter space without being dominated by local fronts or earlier solutions in the series. The second is the size of perturbation required to escape a local front. When perturbations cannot avoid local fronts, and the perturbation required to escape is larger than the maximum perturbation possible with the applied scheme, it is impossible for a greedy algorithm to optimise the problem.

Results on the DTLZ suite show that even though exploratory (non-greedy) moves are not necessary on many problems (and will slow convergence on easy problems, such as DTLZ2), they may still be beneficial; even when it is possible for successive greedy perturbations to move a solution to another area of parameter space.

The introduction of non-greedy-searchable problems, INGS1 and INGS2, results in dramatically different results from those generated on the DTLZ problems, with the greedy algorithms performing considerably worse than their non-greedy counterparts. The INGS problems exhibit local fronts which must be traversed in order to reach the true front. The problems may be tuned in difficulty for greedy optimisers by adjusting the number (and therefore size) of these regions. Empirical results show that a configuration of the INGS2 problem with many local fronts, mapping from many small regions of parameter space, allows the greedy optimisers to outperform the corresponding non-greedy algorithms, as the non-greedy moves slow, rather than aid, convergence. However, greedy optimisers are ineffective when the local fronts are widely seperated.

A novel set based multi-objective simulated annealer (SAMOSA) which stores a non-dominated

set of solutions as the current state was introduced. While simulated annealing requires a current state, which ultimately approximates the results desired, it is not clear whether this state should represent a single solution (as with other multi-objective simulated annealing techniques, including MOSA) or a set of solutions; as the desired result of the optimisation is a set of solutions, maintaining a set of solutions as the state is appealing (and this approach is used in many other optimisation techniques such as genetic algorithms and many evolution strategies). Comparisons between SAMOSA and MOSA showed two traits; that the set slowed convergence to the true Pareto front as measured by the distance of points from the true front, but also that, for problems with highly non-linear mappings from the region representing the true front in parameter space to the front in objective space such as DTLZ4, methods such as Uniselect can be applied to set-based annealers which result in a considerably more uniform estimation of the true front. The slower convergence can be explained as being inherent to set-based methods of this nature, although profoundly different set-based approaches may not have this trait. An example of this can be easily constructed by imagining a series of perturbations: If $\mathbf{x}$ is perturbed and results in solution $\mathbf{x}'_1$ which is non-dominating, this has expanded the front, which is the motivation for using set states. If $\mathbf{x}$ is then chosen for a perturbation again and produces solution $\mathbf{x}'_2$ which dominates $\mathbf{x}$, part of the front is advanced. $\mathbf{x}_1$ is then perturbed, generating $\mathbf{x}_3$, which dominates $\mathbf{x}_1$ but not $\mathbf{x}_2$. If $\mathbf{x}_1$ is then perturbed to $\mathbf{x}_4$ and $\mathbf{x}_4$ dominates all previous solutions, the perturbation from $\mathbf{x}$ to $\mathbf{x}_2$ has clearly been wasted. If a single-solution state had been used, this wasted perturbation would never have been made.

Other researchers [Fleischer, 2003] have suggested the applicability of the dominated volume as an energy function for multi-objective simulating but have not suggested methods for the algorithm construction. Hence, a new volume-based multi-objective simulated annealing (VOLMOSA) algorithm has been investigated. Similar to the SAMOSA previously introduced, VOLMOSA maintains a set of solutions as the current state and perturbations are performed which generate new sets for the new state proposals. Unlike SAMOSA, which uses the relative difference in dominance between the solutions in two sets to calculate the energy change between two states, VOLMOSA calculates the dominated volume of each set and uses the difference between these values as the energy change.

The VOLMOSA algorithm was shown to exhibit performance comparable to SAMOSA on the

two-objective formulations of the DTLZ 1-4 problems. The performance of the SAMOSA is considerably better on the two-objective DTLZ problems than on the three-objective formulations, yielding results both converging close to and covering uniformly the true Pareto front. This suggests that the two-objective DTLZ formulations are significantly easier than the three-objective variants. Unfortunately, the $\mathcal{O}(N^D)$ computational complexity of the calculation of the dominated volume of a set means that it is not feasible, even for $D = 3$, to compare the performance of VOLMOSA and SAMOSA on the more difficult forms of the problems.

To reduce the computational cost of dominated volume calculations, a novel incremental method, suitable for use on dominated sets, was proposed. This can dramatically reduce the complexity of a volume calculation, because as a large proportion of state perturbations in the VOLMOSA algorithm result in the simple addition of a single non-dominated solution to the state it is usually faster in these cases to calculate the newly-dominated volume incrementally than to recalculate the total volume dominated by the entire set.

While a proof of convergence for SAMOSA or VOLMOSA is not directly a consequence of Geman & Geman's proof for single-objective simulated annealing, the primary barrier (that the energy change between two states is dependent upon the previously located estimate of the Pareto front) to the application of a proof in MOSA is not present in these algorithms. The remaining barrier to a simple application of Geman & Geman's proof for both SAMOSA and VOLMOSA is that the optimal set (that representing every globally non-dominated solution) is generally infinitely large, and, as such, unattainable, for problems with continuous fronts.

Chapter 5

Conclusions

This thesis presents an investigation into the development of simulated annealing techniques for multi-objective optimisation and their applicability. During the course of the work, a number of novel techniques have been developed, as well as a significant number of discoveries regarding the behaviour of both these simulated annealing techniques and of traditional greedy search such as (1+1) ES. These are outlined below, before a discussion of the possible future directions of research brought to light in this thesis, and finally a brief summary of the findings of the thesis.

- The identification of the proportion of the Pareto front dominating a point as a measure of the quality of a solution suitable for a simulated annealing energy measure.
- The development of MOSA, a multi-objective simulated annealing algorithm which utilises an approximation of this measure, using an estimate of the Pareto front.
- This new algorithm is shown to demonstrate performance superior to that of an existing multi-objective simulated annealing algorithm, and the NSGA-II genetic algorithm on the standard DTLZ test problems for multi-objective problems (many-objective problems having a dimensionality much greater than 3 are not tested).
- This new algorithm was applied to the optimisation of CDMA mobile telecommunications network configurations, providing superior results to existing single-objective genetic algorithms applied to this problem.

- Two additional multi-objective simulated annealing algorithms were developed, both of which use a set of solutions as the annealer state with new energy calculation methods:
  - SAMOSA implements an extension of the dominance measure used for the MOSA algorithm.
  - VOLMOSA implements a novel incremental volume calculation method for state energies.
- An investigation into the performance of the algorithms revealed that greedy algorithms out-perform the simulated annealers on the DTLZ test problems.
- It is established that in these multi-objective problems, local fronts do not prevent greedy algorithms from converging unless perturbations which 'jump' from one front to another are unlikely.
- Two new test problems were developed which provide more difficult landscapes for greedy algorithms to optimise, where the distance between parameterisations mapping to different local fronts is configurable, thus permitting the difficulty of the problem to be tuned.
- The simulated annealing algorithms are shown to out-perform greedy algorithms when it is unlikely or impossible for a perturbation to move a solution from one local front to a dominating front on multi-objective problems.

5.1 Future Directions

5.1.1 Proof of Convergence

While the proof of convergence for single-objective simulated annealing algorithms is a strong motivating factor in the development of multi-objective simulated annealing techniques, this proof has not been extended to any of the multi-objective simulated annealing algorithms presented here. As it has been shown that simulated annealing only out-performs greedy algorithms on problems with very difficult search landscapes, it is especially desirable to have a proof that it will, eventually, converge to the Pareto front on these problems.

5.1.2 Efficient Perturbation Generation

The investigation into the density of solutions on the Pareto front in Chapter 3 outlines a possible approach to generating perturbations either along the front, or away from the front for exploration. It may be possible, through an estimation of the Jacobian matrix, to guide the search process in producing perturbations of a desirable nature.

5.2 Final Summary

The work presented here provides several substantial contributions in the field of multi-objective optimisation (many-objective optimisation has not been addressed within this thesis). The first, and most obvious, contribution is the development of several multi-objective simulated annealing algorithms which may be applied to optimisation problems. Following this, comparisons of the algorithms show some interesting features; it seems that maintaining a single solution as the state yields faster convergence on the tested problems and it is thought that this is because set perturbations can wastefully perturb many solutions in similar ways but that utilising set perturbation methods which sample solutions uniformly generate more uniform estimated fronts than single-solution states. Surprisingly, it was discovered that problems with many local fronts do not prevent greedy algorithms from out-performing simulated annealing algorithms; this is discussed and new test problems are proposed which are more difficult for greedy algorithms, in which case the new simulated annealing algorithms perform better.

The final message of this thesis is therefore that simulated annealing is a viable, well-performing, technique for the optimisation of multi-objective problems and that while on some problems a greedy algorithm would perform faster, on more difficult problems the simulated annealing algorithms can significantly outperform their greedy-searching counterparts.

Bibliography

S. Ben Jamaa, Z. Altman, J.M. Picard, and B. Fourestié. Multi-objective strategies for automatic cell planning of UMTS networks. In *Vechicle Technology Conference, Milan*, 2004.

M. Berz, C. Bischof, G. Corliss, and A. Griewank, editors. *Computational Differentiation: Techniques, Applications, and Tools.* SIAM, Philadelphia, 1996.

S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler. PISA — a platform and programming language independent interface for search algorithms. In C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization (EMO 2003)*, Lecture Notes in Computer Science, pages 494 – 508, Berlin, 2003. Springer.

D. Büche, S.D. Müller, and P. Koumoutsakos. Self-adaptation for multi-objective evolutionary algorithms. In C.M. Fonseca, P.J. Fleming, E.Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO 2003, Faro, Portugal.*, volume 2632 of *Lecture Notes in Computer Science*, pages 267–281. Springer, 2003.

C.A. Coello Coello, D.A. van Veldhuizen, and G.B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems.* Kluwer Academic Publishers, Boston, 2002.

P. Czyżak and A. Jaszkiewicz. Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34–47, 1998.

V. Grunet da Fonseca, C.M. Fonseca, and A.O. Hall. Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. In E. Zitzler, K. Deb, L. Thiele, C.A.C. Coello, and D. Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 213–225. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

I. Das and J. Dennis. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14(1):63–69, 1997.

K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, 2001.

K. Deb. Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, 1999.

K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *Proceedings of Parallel Problem Solving from Nature - PPSN VI*, pages 849–858. Springer, 2000.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. Technical Report 112, Institute für Technische Informatik und Kommunikationsnetze, ETH Zurich, 2001.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002a.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830. IEEE Service Center, May 2002b.

K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830, 2002c.

P. Engrand. A multi-objective approach based on simulated annealing and its application to nuclear fuel management. In *5th International Conference on Nuclear Engineering*, pages 416–423, Nice, France, 1997.

R.M. Everson and J.E. Fieldsend. Multi-objective optimisation of safety related systems: An application to short term conflict alert. *IEEE Transactions on Evolutionary Computation*, 10(2): 187–198, 2006.

R.M. Everson, J.E. Fieldsend, and S. Singh. Full Elite Sets for Multi-Objective Optimisation. In I.C. Parmee, editor, *Adaptive Computing in Design and Manufacture V*, pages 343–354. Springer, 2002.

J.E. Fieldsend and S. Singh. Pareto Multi-Objective Non-Linear Regression Modelling to Aid CAPM Analogous Forecasting. In *Proceedings of the 2002 IEEE International Joint Conference on Neural Networks*, pages 388–393, Hawaii, May 12-17, 2002a.

J.E. Fieldsend and S. Singh. A Multi-Objective Algorithm based upon Particle Swarm Optimisation, an Efficient Data Structure and Turbulence. In *Proceedings of UK Workshop on Computational Intelligence (UKCI'02)*, pages 37–44, Birmingham, UK, Sept. 2-4, 2002b.

J.E. Fieldsend, R.M. Everson, and S. Singh. Using Unconstrained Elite Archives for Multi-Objective Optimisation. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, 2003.

M. Fleischer. The measure of Pareto optima: Applications to multi-objective metaheuristics. In C.M. Fonseca, P.J. Fleming, E. Zitzler, K. Deb, and L. Thiele, editors, *Evolutionary Multi-Criterion Optimization, Second International Conference, EMO2003*, volume 2632 of *Lecture Notes in Computer Science*, pages 519–533. Springer, 2003.

D. B. Fogel. *Evolutionary Computation: The Fossil Record.* IEEE Press, Piscataway, NJ, USA, 1998.

C.M. Fonseca and P.J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Genetic Algorithms: Proceedings of the Fifth International Conference*, pages 416–423. Morgan Kaufmann, 1993.

C.M. Fonseca, L. Paquete, and M. López-Ibáñez. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of 2006 Congress on Evolutionary Computation*, pages 3973–3979, 2006.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Kluwer Academic Publishers, Boston, 1989.

G.H. Golub and C.F. Van Loan. *Matrix Computations*. North Oxford Academic, Oxford, 1983.

D. Greening. Simulated annealing with inaccurate costs functions. In *Proceedings of the IMACS International Congress of Mathematics and Computer Science*, Trinity College, Dublin, 1993.

M. Hapke, A. Jaszkiewicz, and R. Slowinski. Pareto simulated annealing for fuzzy multi-objective combinatorial optimization. *Journal of Heuristics*, 6(3):329–345, 2000.

S. Huband, L. Barone, L. While, and P. Hingston. A scalable multi-objective test problem toolkit. In *Evolutionary Multi-Criterion Optimization: Third International Conference, EMO 2005*, volume 3410 of *Lecture Notes in Computer Science*, pages 280–294. Springer-Verlag, 2005. Corrected version at `http://www.wfg.csse.uwa.edu.au/publications/WFG2005a_corrected.pdf`.

EJ Hughes. Evolutionary many-objective optimisation: Many once or one many? *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, 1, 2005.

L. Ingber. Simulated annealing: Practice versus theory. *Mathematical Computation and Modelling*, 18:29–57, 1993.

J. Jahn, J. Klose, and A. Merkel. On the application of a method of reference point pproximation to bicriterial optimization problems in chemical engineering. *Advances in optimization, pp*, pages 478–491, 1991.

A Jaszkiewicz. Comparison of local search-based metaheuristics on the multiple objective knapsack problem. *Foundations of Computer and Decision Sciences*, 26(1):99–120, 2001.

M. T. Jensen. Reducing the Run-time Complexity of Multi-Objective EAs: The NSGA-II and other Algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5):502–515, 2003.

S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220: 671–680, 1983.

J. Knowles and D. Corne. The Pareto Archived Evolution Strategy: A new baseline algorithm for Pareto multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 98–105, 1999.

J.D. Knowles and D. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

J. Korhonen. *Introduction to 3G mobile communications*. Mobile communications series. Artech House Publishers, Norwood, MA, 2001.

J. Koski. Multicriteria truss optimization. *Multicriteria Optimization in Engineering and in the Sciences*, 37:263–307, 1988.

M. Laumanns, E. Zitzler, and L. Thiele. A Unified Model for Multi-Objective Evolutionary Algorithms with Elitism. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 46–53, 2000.

M. Laumanns, G. Rudolph, and H.P. Schwefel. Mutation control and convergence in evolutionary multi-objective optimisation. In R. Matousek and P. Osmera, editors, *Proceedings of the 7th International Mendel Conference on Soft Computing (MENDEL 2001)*, pages 24–29, Brno, Czech Republic, 2001.

M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running Time Analysis of Multi-objective Evolutionary Algorithms on a Simple Discrete Optimization Problem. In J.J. Merelo Guervós, P. Adamidis, H-G Beyer, J-L Fernández-Villacañas, and H-P Schwefel, editors, *Parallel Problem Solving from Nature—PPSN VII*, Lecture Notes in Computer Science, pages 44–53. Springer-Verlag, 2002.

M. Laumanns, L. Thiele, and E. Zitzler. Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-Boolean Functions. *IEEE Transactions on Evolutionary Computation*, 8 (2):170–182, 2004.

N. Metropolis, A.W. Rosenbluth, M. Rosenbluth, A. H. Teller, and E Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

M. Mitchell. *An Introduction to Genetic Algorithms*. The MIT Press, 1996.

D.K. Nam and C.H. Park. Multiobjective simulated annealing: a comparative study to evolutionary algorithms. *International Journal of Fuzzy Systems*, 2(2):87–97, 2000.

T. Okabe, Y. Jin, and B. Sendhoff. On The Dynamics Of Evolutionary Multi-objective Optimization. *Proceedings of the Genetic and Evolutionary Computation Conference table of contents*, pages 247–256, 2002.

T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff. On test functions for evolutionary multi-objective optimization. *Parallel Problem Solving from Nature VIII*, pages 792–802, 2004.

V. Pareto. *Cours d'économie politique*. Librairie Droz, 1964.

J.T. Richardson, M.R. Palmer, G. Liepins, and M.Hilliard. Some guidelines for genetic algorithms with penalty functions. In Morgan Kaufmann, editor, *Proceedings of Third International Conference on Genetic Algorithms*, pages 191–197, 1989.

P. Salamon, P. Sibani, and R. Frost. *Facts, Conjectures, and Improvements for Simulated Annealing*. Number 7 in Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics, Philadelphia, 2002.

I.F. Sbalzarini, S. Müller, and P. Koumoutsakos. Microchannel optimization using multiobjective evolution strategies. In E. Zitzler, K. Deb, L. Thiele, C.A. Coello Coello, and D. Corne, editors, *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO)*, Lecture Notes in Computer Science, pages 516–530. Springer, 2001.

J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In Lawrence Erlbaum, editor, *Proceedings of First International Conference on Genetic Algorithms*, pages 93–100, 1985.

P. Serafini. Simulated annealing for multiobjective optimization problems. In *Multiple criteria decision making. Expand and enrich the domains of thinking and application*, pages 283–292, 1994.

K.I. Smith, R.M. Everson, and J.E. Fieldsend. Dominance measures for multi-objective simulated annealing. In *Proceedings of Congress on Evolutionary Computation, CEC04*, pages 23–30, 2004.

K.I. Smith, R.M. Everson, J.E. Fieldsend, R. Misra, and C. Murphy. Dominance-based multi-objective simulated annealing. *IEEE Transactions on Evolutionary Computation*, 2006. (under review).

A. Suppapitnarm, K.A. Seffen, G.T. Parks, and P.J. Clarkson. A simulated annealing algorithm for multiobjective optimization. *Engineering Optimization*, 33:59–85, 2000.

G. Szabó, K. Weicker, N. Weicker, and P. Widmayer. Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment. *IEEE Transactions on Evolutionary Computation*, 7:189–203, 2003.

D. Tuyttens, J. Teghem, and N. El-Sherbeny. A particular multiobjective vehicle routing problem solved by simulated annealing. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for multiobjective optimisation*, volume 535 of *Lecture Notes in Economics and Mathematical Systems*, pages 133–152. Springer, 2003.

E.L. Ulungu, J. Teghaem, Ph. Fortemps, and D. Tuyttens. MOSA method: a tool for solving multiobjective combinatorial decision problems. *Journal of Multi-criteria Decision Analysis*, 8: 221–236, 1999.

X. Yao, Y. Liu, and G. Lin. Evolutionary Programming Made Faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.

E. Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications.* PhD thesis, Swiss Federal Institute of Technology Zurich (ETH), 1999. Diss ETH No. 13398.

E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms — A comparative case study. *Lecture Notes in Computer Science*, 1498, 1998.

E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 2(8):173–195, 2000.

E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation, and Control*, pages 95–100. CIMNE, Barcelona, Spain, 2002.