Chapter 1
Introduction

The management of Water Distribution Systems (WDSs) involves making decisions about various operations in the network, including the scheduling of pump operations or setting of disinfectant dosing rates. There are often conflicting objectives in making these operational decisions, such as minimising costs while maximising the quality of the water supplied. Hence, the operation of WDSs can be quite complex, and there is generally considerable scope to improve the operational efficiency of these systems by improving the associated decision making process.

The fact that Genetic Algorithms (GAs) are guided search methods that are capable of carrying out optimisation in conjunction with any simulation model has allowed them to be adopted in a wide range of disciplines, such as engineering, numerical optimisation, robotics, classifier systems, pattern recognition and design (Herrera et al., 1998). In the vast majority of these and many other applications, the results produced by the GA are very promising. The optimisation of WDSs is no exception, as identifying the best design and operation for these networks can be very difficult. Since the first application of a GA to a WDS optimisation (Simpson et al., 1994) this method has displayed great success on a range of design and operational problems (Al-Zahrani and Moied, 2003; Munavalli and Kumar, 2003; Ostfeld and Salomons, 2004b; Prasad et al., 2004; Tolson et al., 2004; Van Zyl et al., 2004).

While there has been extensive research demonstrating the potential of GAs for improving the design and operation of WDSs, generally the optimisation method has had limited adoption in practice. While in theory a GA can operate on any objective function and generally find good solutions, as no special function formulation is required, the process of selecting and applying a GA still requires expert knowledge to provide suitable results. There are a number of reasons that contribute to this lack of uptake, including difficulty in making decisions that are required to select the most appropriate variant of the algorithm, determining the most appropriate parameter settings for the algorithm, and

developing an appropriate fitness function to describe the objective of the optimisation including all constraints. Therefore, the use of GAs for WDS optimisation is still mainly limited to the research domain.

While these are all important considerations, the correct selection of GA parameter values is addressed in this thesis. Common parameters include population size, probability of crossover, and probability of mutation. Both the absolute value used of these parameters, as well as their relative values, will determine how the GA finds new solutions and, ultimately, the quality of the final solution found. The values adopted for the GA parameters will produce a searching behaviour between two possible extremes: *exploitation*, where the current best solution is used as a basis for finding better solutions; and *exploration*, where solutions are combined and random search is used to explore the entire search space (Herrera et al., 1998).

While certain operators will be beneficial to certain optimisation problems, in general it would be expected that a properly calibrated GA would perform better than a specialised GA with arbitrary parameter values. The setting of the GA parameters may be one of the reasons why GAs have not been adopted, as they must be selected on a problem by problem basis, and there is not a single set of parameter values that will be suitable for all problems (Wolpert and Macready, 1997). Hence, just because a certain parameter set has been found to perform well on a certain WDS problem, does not imply it will perform well on another. A poorly selected set of parameter values may lead the GA to converge prematurely to sub-optimal solutions, or direct it to search the fitness function almost randomly, without making maximum use of current knowledge to find better solutions.

As the most suitable GA parameters must be found for each individual optimisation problem, it might be expected that the best parameter values would be related to the characteristics of the associated fitness function. The goal of this research is to develop a GA calibration methodology based on the characteristics of the fitness function. It is proposed that fitness function statistics can be used to provide this information.

The success of GAs in the WDS optimisation field is not unfounded, as it has been proven using Markov chain modelling that a GA can be guaranteed to converge to the optimal solution of an optimisation problem (Suzuki, 1995). Generally, this convergence will not occur in a realistic time frame, and a GA can therefore only provide a good solution, rather than guarantee that the optimal solution will be found. The trade off between the time to find a solution and the quality of a solution found must be taken into consideration in the parameters used for a GA. If the focus is on finding good solutions quickly and repeatedly (such is the case in an operational situation) parameters may be

set to speed up the optimisation process, while reducing the likelihood of finding the very best solutions. However, if the problem is design-oriented and there are large costs involved, the optimisation process should be tailored to find the best solution, irrespective of how long it may take. Another goal of this research is to take into consideration the time available to find a solution in the GA calibration methodology.

The following section outlines the goals of this research in more detail. This is followed by the methodology that is proposed to obtain these goals, before the chapter is concluded with an outline of the layout of the remainder of this thesis.

1.1 GOALS OF THIS RESEARCH

The general goal of this research is to contribute to making GAs more user-friendly, and assist in their acceptance and application in water utilities. There are many factors that must be addressed to ultimately achieve this goal, including: how to formulate the objective function, the most suitable decision variable representation, the best algorithm operators to use and the parameter values to use for the operators selected. However, if an algorithm is calibrated correctly, its performance will be maximised for a given problem formulation, irrespective of operators or decision variable representation used. Therefore, this research has focused on the parameterisation problem, how to set the algorithm parameters for a given optimisation problem. The objective of this research is to develop a methodology to determine the most suitable GA parameter values for a given problem, with a focus on an application to WDS optimisation.

In order to meet this main objective, the secondary objectives of this research are to:

- Make use of genetic theory and dimensional analysis studies to identify relationships to assist in setting the most influential GA parameter, the population size;
- Determine if the characteristics of the optimisation problem have an influence on the most effective population size to solve the problem;
- Investigate methods to provide useful information about characteristics of optimisation problems;
- Quantify the relationship between the characteristics of an optimisation problem and the most effective population size to solve the problem;
- Determine relationships that can be used to determine the most suitable values for the remaining GA parameters;
- Combine the optimisation problem characterisation and parameter setting relationships into a step-by-step GA calibration methodology;

- Compare the performance of the proposed methodology against existing state of the art GA calibration methods; and
- Apply the proposed GA calibration methodology to realistic WDS optimisation problems.

1.2 Proposed Methodology

As outlined in Section 1.1, the aim of this work is to provide a GA calibration methodology that can be applied practically to WDS optimisation problems. In their work developing simple dimensional models to further the understanding of mixing in GAs, Goldberg et al. (1993) suggested that:

> "There seems to be a growing call for more complex analyses [of GA behaviour], but it must be remembered that analysis here is the handmaiden of design; we do analysis to better understand GAs and to make them work better. The experience of other fields involving complex systems—fluid mechanics for example—shows that it is easiest to do the necessary asymptotics on full equations once one knows where to look. Our approach here is designed to illuminate such nooks, perhaps permitting the now premature calls for rigor to be answered one day."

A similar approach is taken in this work. Rather than developing theoretically sound results that have little applicability, an experimental approach is adopted, with the aim to provide useful results and further the understanding of the effect of problem characteristics on the most suitable GA parameter values. The methodology proposed in this work draws on a number of results and observations from different areas of GA research. This thesis aims to use both modelling results and experimental results to investigate the relationship between problem characteristics and the optimal number of GA generations. The use of experiments ensures that the results are realistic, as all GA operators are included in the analyses, and the use of previous modelling results and a consideration of problem characteristics provides some generality to the results.

The GA calibration methodology proposed in this thesis is based on a number of observations regarding GA behaviour. The first observation is that for WDS optimisation, and most real-world optimisation problems, it is unlikely that the optimal solution will ever be found. For example, lower cost solutions have recently been found (Maier et al., 2003) to The New York Tunnels Water Supply optimisation problem (Schaake and Lai, 1969), a benchmark WDS optimisation problem which involves determining the lowest

cost network expansion to meet increasing demands on the system, even though the problem was first attempted almost 40 years ago. Therefore, as the optimal solution is most likely never found for realistic WDS optimisation problems, a GA can only aim to find the best solution possible in a specified time. Therefore, the proposed approach revolves around providing the best search behaviour possible, before the GA must provide a solution.

A second observation is that a larger population size generally locates better solutions than a smaller population size, provided there is time available for the GA to converge with the larger population size. A number of dimensional analysis studies, such as those by Thierens and Goldberg (1994) and Thierens et al. (1998), suggest there is a constant number of generations before a GA will converge on a given fitness function, and the number of generations is a function of the problem size, or the length of the solution string, l. If a GA will converge in a certain number of generations, g_{conv}, for a given fitness function, and the time before a solution is required, expressed as the number of available fitness function evaluations (FE), is known, then the most suitable population size, N, can be calculated using $N = FE/g_{\text{conv}}$. Therefore, the first hypothesis of this thesis is that the number of generations before a GA will converge for a given optimisation problem is constant for changes in the number of fitness function evaluations, and Chapter 3 is dedicated to testing this hypothesis.

Another observation is that the most suitable GA parameters must be found for each individual problem. Therefore, the best parameter values might be expected to be related to the characteristics of the fitness function. However, generally the characteristics of the fitness function are largely unknown, especially when the fitness function is based on a simulation model and the precise mathematical relationship between the decision variables is unknown. Chapter 4 investigates the second hypothesis of this thesis: that fitness function statistics can be used to provide useful information about the characteristics of fitness function.

The final hypothesis of this thesis is that the number of GA generations before convergence can be predicted using the information provided by the fitness function statistics. Chapter 5 is dedicated to quantifying the relationship between the information provided by the fitness function statistics and the number of generations before convergence is observed for a suite of fitness functions with varying characteristics. From this relationship and the time available before a solution is required, the most influential GA parameter, the population size, can be determined. Relationships are also developed to determine the most effective values for the remaining GA parameters, thereby producing a complete

GA calibration methodology.

While the observations stated above regarding GA behaviour are rather general, the proposed methodology has a sound basis in Quantitative Genetics. As GAs are derived from observations of natural selection, it is not surprising that results from Quantitative Genetics have been applied to the GA field. A number of studies have made use of the relationship that explains a biological population's response to natural selection to investigate GA convergence (Lobo, 2000; Mühlenbein and Schlierkamp-Voosen, 1993; Rogers and Pruegel-Bennett, 1999; Thierens and Goldberg, 1994; Thierens et al., 1998). The relationship states that the change in the average fitness value of the population after selection is equal to the selection pressure multiplied by the standard deviation of the fitness values. Also due to the influence of selection, the standard deviation of fitness values decreases by a constant value, k, each generation (Falconer, 1981), producing an exponential decay of the variance in the fitness function values in terms of the number of generations. Therefore, the time until the GA converges to a single solution is dependent on the rate of decay of the population variance.

A study of Quantitative Genetics also provides an insight into the components of the variance in the fitness values. Falconer (1981) cites a number of components that comprise the total fitness variance, namely:

- Breeding or Additive variance (σ_A^2);
- Dominance variance (σ_D^2);
- Interaction variance (σ_I^2); and
- Environmental variance (σ_E^2).

These components of variance that have been identified in Quantitative Genetics theory are analogous to characteristics of fitness functions that have been identified previously to affect the performance of GAs. σ_A^2 can be considered to be variance in the population, which will obviously affect the variance in the corresponding fitness function values. σ_D^2 is the variance provided by dominant, or highly salient, variables; where a highly salient gene has a greater influence on the fitness function value. The impact of this component on GA convergence has been highlighted in the work by Thierens and Goldberg (1994) looking at uniformly salient genes, and Thierens et al. (1998) considering exponentially scaled genes. σ_I^2 is the variance produced by interactions between genes, commonly known as epistasis, and a large body of work in this area indicates the significance of this component on GA performance. σ_E^2 is the variance produced by the environment, which can be interpreted as the shape of the fitness function, for example if it is relatively smooth, or if it is very rough and improved solutions are relatively harder

to find.

It is proposed that fitness function statistics that provide an insight into these characteristics of fitness functions can be used to provide information about these components of variance, and in turn can be used to predict the change in standard deviation of fitness values, k. Therefore, the number of generations before the variance in the fitness function values decreases to zero can be predicted, and from the time available to find a solution, the most suitable population size can be obtained.

1.3 Layout and Contents of Thesis

Chapter 2 provides the background relevant to the work undertaken in this thesis. Firstly, a review of WDS optimisation is presented, including different constraint handling methods, optimisation methods that have been applied to WDS problems, and the different applications that have previously been considered. This is followed by an in-depth review of GAs, including the different operators and representations that are available, the different theories that have been used to explain GA behaviour, and the approaches that have been applied to assist in the calibration of GA parameters. This chapter is concluded with the specific GA that has been adopted to be used in this work, and the assumptions that have been made to make this work possible.

The flow of work presented in this thesis, and the chapters in which the work is presented, is depicted in Figure 1.1. Chapter 3 is dedicated to experimentally observing the hypothesis that there exists a given number of generations before a GA population will converge to a single solution. A number of function characteristics are identified that when present validate this hypothesis, and based on this two classes of functions are identified; Optimal Generation Functions, which are best solved with a constant number of generations, and Maximal Generation Functions, which are best solved with as many generations as possible of a small population size. The second part of this chapter considers a number of controlled changes to the characteristics of a number of fitness functions, to determine that the number of generations before the population will converge is a function of the characteristics of the fitness function.

As the number of generations before convergence was found to be dependent on the characteristics of the fitness function in Chapter 3, Chapter 4 is dedicated to developing a number of fitness function statistics to provide information about the different characteristics of fitness functions identified. The first statistics developed is the spatial correlation measure, used to provide information about the structure and multimodality of a fitness function. Next, the separability measure is proposed to identify epistatic interactions be-

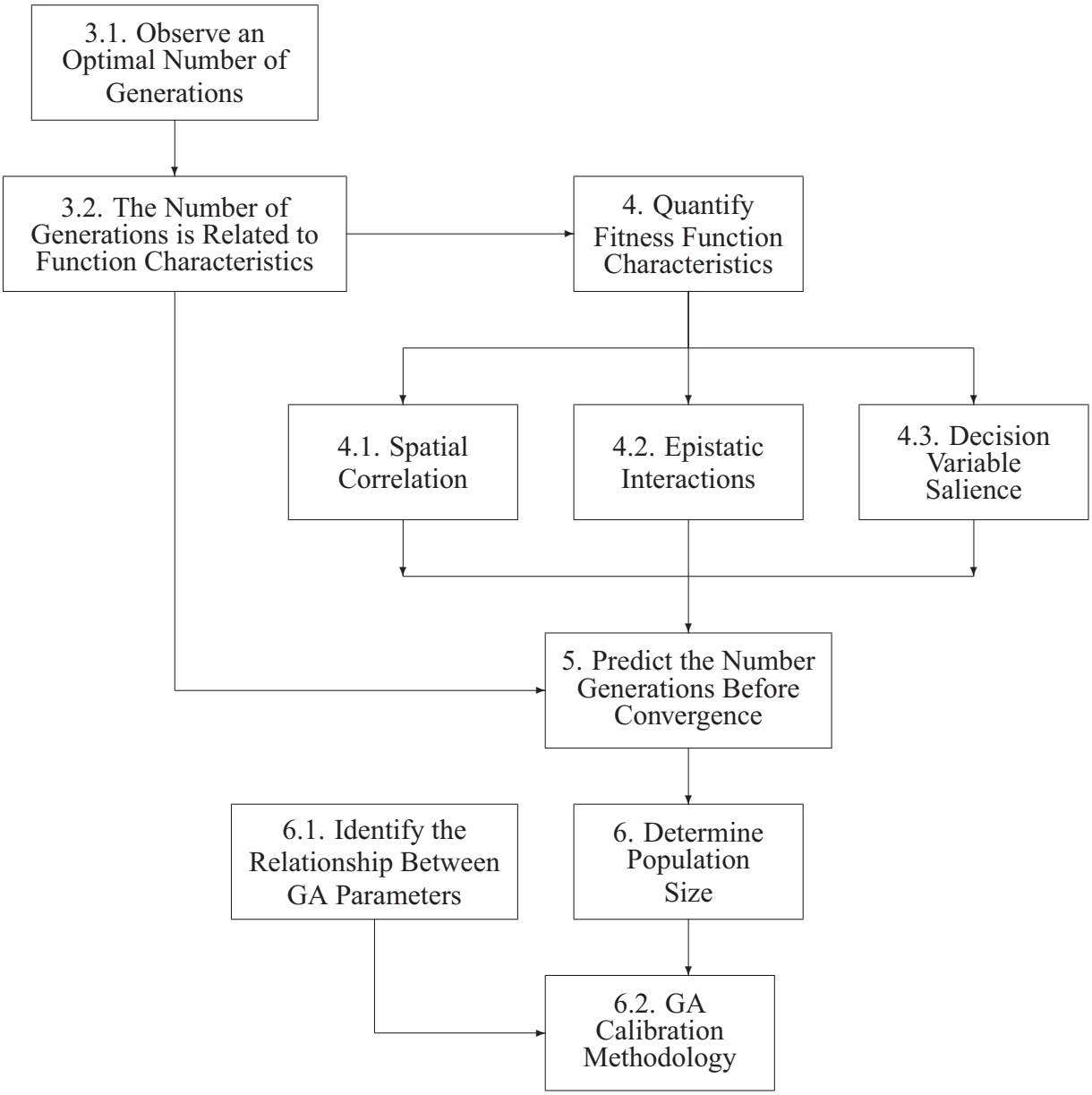


Figure 1.1 Flowchart depicting the relationship between the work presented in this thesis.

tween decision variables, both in terms of the number and the length of the interactions. The final statistic developed is the dominance measure, used to identify if any of the decision variables has a much larger influence on the fitness function value than the remaining decision variables. Each of the statistics was applied to fitness functions with known characteristics, and each statistic was confirmed to provide accurate information.

Chapter 5 is devoted to developing a relationship between the number of generations before the GA population converges and the information provided by the fitness function statistics developed in Chapter 4. A class of fitness functions with controllable characteristics is considered, and the change in population variance due to selection, and the values of the fitness function statistics, are computed for each variation of the fitness function considered. From these results an empirical relationship is developed to predict the change in population variance based on the values determined by the fitness function statistics. Based on the change in population variance due to selection, the number of generations before the population will converge can be determined. The final part of this chapter is dedicated to validating the predictions of both the change in population variance, and the number of generations before convergence, for the different functions considered in Chapter 3.

In Chapter 6 a full GA calibration method based on the characteristics of the fitness function is developed and tested. Based on the number of generations that can be predicted from the relationship developed in Chapter 5, the population size can be determined from the time available before a solution is required. However, this is only one of the GA parameter values, and the first part of this chapter is dedicated to developing relationship for determining suitable values for the remaining GA parameters. Due to the computation effort required to compute the value of the fitness function statistics, a GA calibration method based on the influence of genetic drift on the GA population has also been proposed. Using this method, a suitable GA population size can be determined from only the problem size and the time available before a solution is required. Both of the proposed calibration methods are then tested against other GA calibration methods that are available. A range of complex fitness functions are used as a basis for the comparison, and the results indicate that the GA calibration method developed produced the best performing GA parameter values.

Chapter 7 applies the GA calibration method proposed in Chapter 6, along with the other calibration methods that have been considered in Chapter 6, to the optimisation of WDSs. The first system considered is the Cherry Hill-Brushy Plain network, where the aim of the optimisation is to minimise the amount of the disinfectant injected into the

system over a 24-hour period. The second WDS considered is a 'real-world' case study taken from the Woronora WDS located in Sydney, Australia. The objective in this case is to minimise the costs involved in operating the pumps in the system, with consideration of the water quality produced by the operations.

Concluding remarks from this work are provided in Chapter 8. Firstly, the contributions to knowledge that have been made in this thesis are outlined. Secondly, conclusions based on the results presented in work are made, and finally suggestions for directions for future work are provided.

Chapter 2
Background

This chapter provides a review of the relevant background regarding the optimisation of WDSs, including a detailed review of GAs. The following section reviews the optimisation of WDSs, including the different applications that have been tackled, and the different optimisation methods that have been adopted. The remainder of this chapter is dedicated to background for GAs, firstly a review of the different GA operators, then the theory of GA behaviour, and finally a review of the different methods that have been proposed to assist in the calibration of the GA parameters. The chapter concludes with a summary of the literature covered, and how it is related to the calibration methodology proposed in Section 1.2.

2.1 WDS Optimisation

WDSs consist of a complex arrangement of reservoirs, tanks, pumps, pipes, valves and various other control structures. Due to this complexity, the design and management of these networks can be extremely complicated. In order to gain a better understanding of a WDS, engineers and operators make use of computer models to simulate the network. Quite often these models are used in a trial and error environment to test out different scenarios to determine the impact they will have on the system. While this approach can find suitable solutions, they will generally be of poor quality compared to the best possible solution, as only a limited number of alternatives can be evaluated, and the quality of the solutions implemented will be driven by the operator's experience. By applying an optimisation algorithm, this process of evaluating different potential solutions can be automated to some extent, allowing the algorithm to select which solutions to test based on the problem and previous solutions.

2.1.1 Constraint Handling

Before a WDS problem can be optimised, the problem must be presented as a suitable function to be used by the optimisation algorithm to compare different solutions. The main issue to be considered when developing this function is how to account for the constraints placed on the distribution network. Many of these constraints are not on the decision variables themselves, so it is not known if a solution is feasible before a simulation of the network takes place. For the example of a network expansion problem, it is not known if a solution will satisfy all pressure constraints just by looking at the different pipe diameters that have been proposed. Coello Coello (2002) provides an in-depth review of the different constraint handling methods that are avaliable, and the most common methods used for constraint handling are detailed below.

In cases where the constraints are the decision variables, or when it is clear that a constraint can be met by changing a certain decision variable, repair methods can be used. An example may be when optimising pumping trigger levels based on the water level in a tank, where the lower trigger level must be below the upper trigger level, and both levels are decisions to be made by the optimisation algorithm. Once the solution is repaired, it can be used for evaluation only, or to replace the infeasible solution in the optimisation algorithm. In terms of GAs, the approach of replacing infeasible solutions with repaired solutions in the population is called *Lamarckian Evolution*, which assumes that an individual improves during its lifetime and that the resulting improvements are coded back into the chromosome (Coello Coello, 2002) (see Section 2.2). Some studies have used repaired solutions for evaluation only (Liepins and Vose, 1990), or always replaced infeasible solutions (Nakano and Yamada, 1991), however, the best results have been reported using a combination of both approaches, where a probability is used to select one of the two approaches for each solution that is repaired. Orvosh and Davis (1993) suggest replacing 5% of infeasible solutions with their repaired values, whereas Michalewicz (1996) found that a 15% replacement rule was the best choice for numerical optimisation problems with nonlinear constraints.

If a solution cannot be easily repaired, penalty functions are the most common method of taking constraints into consideration when optimising WDSs (Murphy et al., 1993; Simpson et al., 1994; Dandy et al., 1996; Hewitson, 1999; Eusuff and Lansey, 2003; Maier et al., 2003). In this case, the cost of a solution used for the optimisation process consists of the real cost, such as pipe and electricity costs, and penalties on any violation of constraints, such as $\$20\,000$ for every metre the pressure at a demand node is below the required pressure. This allows a lower penalty on solutions closer to the acceptable range,

as a pressure violation of 2 mm will attract a much lower cost than a pressure violation of 2 m. This can be beneficial to the optimisation process, as the optimal solutions will generally exist on the boundary between the feasible and infeasible regions (Coello Coello, 2002), therefore information about solutions close to feasibility can be very useful to the optimisation algorithm.

Often the constraints placed on an optimisation problem will conflict with the objective. For example, the objective of an optimisation problem may be to minimise the cost of construction, however to meet pressure or reliability constraints, it may be necessary to increase the number or size of some pipes, thus increasing the overall cost of the design. Therefore, the balance of the penalty factors is very important in the optimisation process. If the penalty factors are too high, extremely high total costs can result for all infeasible solutions, even those solutions that are very close to the feasible region, which will therefore not be considered in future iterations of the optimisation process. Generally high penalty factors will produce a very rough fitness function (see Section 2.4.1), potentially making the problem much more difficult to solve. However, if the penalty factors are too small, the search process will spend most of the time searching the infeasible space, and the final solutions found may not be feasible.

A number of adaptive penalty methods have been proposed to address the problem of determining the most suitable penalty values. A number of studies (Joines and Houck, 1994; Michalewicz and Attia, 1994) suggest an increase in the penalty values with the number of generations. This allows the optimisation method to find low cost solutions first, and then focus on addressing the constraints as the penalty values are increased. However, it is just as difficult, if not more so, to determine the most suitable rate at which to change the penalty values, as it is to set static penalty values. Other methods utilise feedback from the search progress to adjust the penalty values (Coit et al., 1996; Rasheed, 1998), where the penalty values for constraints that are not being met are increased, while the penalties are decreased for constraints that are satisfied, in order to encourage exploration. The approach proposed by Rasheed (1998) tracks two key solutions, the solution that best satisfies the constraints on the problem, and the point that has the best value for the objective, irrespective of the constraints. The penalty values are then adjusted so that these two solutions have the same fitness function value, and a multiplicative limit on the change of the penalty coefficient is included at each increment to prevent abrupt changes in the penalty values (Rasheed, 1998). This approach has the advantage of automatically adjusting the penalty values to more appropriate values, increasing them if a certain constraint is not being met, or relaxing them if a constraint is satisfied, thereby encouraging

the optimisation algorithm to explore solutions closer to the constraint boundary.

A different approach to considering constraints on an optimisation problem is to use a multi-objective optimisation algorithm (Coello Coello, 2000). Rather than finding the best, or a number of the best solutions to a problem, this method produces trade off curves between the objectives and constraints on a problem, for example the degree of pressure satisfaction obtained for a network with a certain cost. The user must then determine which solution is the most suitable, as factors such as the increase in pressure obtained by installing a higher cost solution can be taken into consideration. However, it becomes much more difficult to determine the most suitable solution from the trade off curves once more than two constraints are considered at a time, as they can no longer be visualised in three dimensions. Multi-objective methods also become less advantageous when the constraints are not simply of the upper or lower bound, for example when the disinfectant residual must be kept within a certain bound. The main advantage of a multi-objective approach is that the explicit definition of penalty factors is not required, as the importance of one constraint violation over another is left to the user to decide.

2.1.2 Previous WDS Optimisation Methods

Once the function describing the problem is specified, including all constraints, the problem can be optimised. The simplest method that can be applied to solve the problem is complete enumeration, which involves evaluating every possible combination of the decision variables to find the best. This method has the advantage of not requiring derivatives of the objective function to be calculated, and guarantees that the best possible solution is found. However, this approach is very computationally intensive. Savic and Walters (1997) found that enumeration becomes infeasible when applied to the optimisation of any realistic-sized water distribution system. Pruning the sets of decision variables evaluated to those that seem practical, a technique known as partial enumeration, can significantly reduce computational requirements. However to reduce the problem to a practical size many solutions must be ignored, which can result in cropping the optimal solution as well (Dandy et al., 1993).

Virtually every mathematical programming technique developed has been applied to the WDS optimisation problem (Eusuff and Lansey, 2003). These procedures include linear and non-linear programming and gradient methods (Gupta et al., 1999). Generally, these methods required the derivatives of the search space to be calculated, which can be a computationally demanding process as numerical methods will almost certainly be required. The main disadvantage of these optimisation methods is the high likeli-

hood of premature convergence of the solution to a local optimum when solving complex non-linear optimisation problems (Calegari et al., 1999). Non-linear programming methods provide some improvements over simpler linear optimisation methods in this regard, however, they cannot guarantee that near-optimal solutions will be found due to the many peaks in the optimisation surface. Implementation of linear and non-linear programming methods may also involve altering the governing hydraulic equations (continuity and energy) to suit the optimisation technique. This approach leads to an approximation of the true problem, which may overlook the best area of the true search space.

Recently, developments in stochastic optimisation algorithms have been applied to WDS problems with great success. Evolutionary Algorithms (EAs) are a group of these optimisation methods that uses an objective function in a strategic manner to seek a solution to an optimisation problem (Eusuff and Lansey, 2003), and their application to the optimisation of WDSs is reviewed in the following section.

2.1.3 Evolutionary Optimisation of WDS

EAs are probabilistic optimisation methods that are based on naturally occurring phenomena, and how these natural systems manage to find very good solutions. By far the most commonly implemented EA is the GA, based on the theory of natural selection. An in-depth review of GAs is provided in Section 2.2.

The fundamental difference between EAs and traditional deterministic methods is that EAs work with a set of candidate solutions, generally called the population (Calegari et al., 1999). EAs improve the quality of the individuals by allowing the population to evolve. The evolution is usually achieved using information exchanges between individuals in order to create new, or to modify existing, individuals, through a succession of evolutionary steps (Calegari et al., 1999). As for traditional optimisation algorithms, an objective function (generally called the fitness function for EAs) is required to associate a value with each solution set in the population. As new, improved solutions are created from information from previous solutions, the process is independent of the slopes of the objective function being optimised. This provides EAs with the significant advantage of being less susceptible to becoming trapped in local minima, as is the case with deterministic mathematical optimisation algorithms. This can also be a significant advantage in the case when the objective function cannot be easily differentiated, which is typically the case for WDS optimisation problems. Generally, the optimisation process contains a random element to encourage exploration of the search space, and this reduces the chances of premature convergence to sub-optimal solutions.

The main disadvantage in applying EAs is the large number of simulations required to find a solution. WDS models that consider water quality must be run for an extended period of time, often requiring simulations of many days to ensure that water has the necessary time to travel to the end of the network. Even on the fastest computers, this simulation can take many seconds, even minutes to complete, and when 100 000s of these simulations are required, the time to find a solution can be in the order of weeks for some problems. This highlights the need for applying the most effective algorithm to the problem at hand, as it is necessary to explore the search space comprehensively to find the best solution, but any more than is required will have a substantial impact on the computation time taken to find the solution.

2.1.4 Applications of EAs to WDS Optimisation

GAs have been by far the most popular EA used for optimisation in the WDS field. Dandy et al. (1996), Savic and Walters (1997), Simpson et al. (1994) and various other authors have applied GAs to determine the optimum design of WDSs. The design has involved selecting pipe sizes for either a network expansion or a whole new system in order to identify the minimum cost design while meeting network constraints, which are typically minimum pressure requirements. Murphy et al. (1994) extended this optimisation procedure further by applying a GA to optimise both how the network was operated as well as the best pipe sizes required for design.

A number of studies have optimised the operation of WDSs using GAs, as distinct to determining the optimal design of the system. Generally, the main objectives are to minimise the electricity cost involved in operating a Water Treatment Plant (WTP), by scheduling the operation of the pumps in a way that takes advantage of the off-peak electricity tariffs (Mackle et al., 1995; Van Zyl et al., 2004), or determining the most cost effective disinfectant dosing regimes (Hewitson, 1999; Munavalli and Kumar, 2003; Ostfeld and Salomons, 2004b; Prasad et al., 2004), or both (Sakarya and Mays, 2000). Hewitson (1999) found that when water quality constraints were considered in the optimisation process, the optimum solution found was significantly different to that obtained when only hydraulic constraints were considered.

A great deal of other applications have been considered using GAs for water engineering applications. Reis et al. (1997) used a GA to optimise the locations of control valves to minimise leakage, Dandy and Engelhardt (2001) used a GA to determine the optimum replacement schedule for deteriorating pipes in a network, and similarly Halhal et al. (1997) used a messy multi-objective GA to optimise the benefits obtained from net-

work rehabilitation methods. The location of monitoring stations has also been optimised using GAs, either to measure indicators of water quality (Al-Zahrani and Moied, 2003), or to have the greatest chance of quickly identifying the injection of a foreign agent into a WDS (Ostfeld and Salomons, 2004a).

EAs other than the GA have also been applied to the optimisation of WDS. Maier et al. (2003) applied Ant Colony Optimisation Algorithms (ACOAs) to the optimal design of WDS. The ACOA was applied to two benchmark WDS optimisation problems, a simple 14-pipe network expansion problem and the New York City Water Supply Tunnels problem (Schaake and Lai, 1969). The results were compared with those obtained using GAs, and it was found that ACOAs are an attractive alternative to GAs for the optimal design of WDSs.

Eusuff and Lansey (2003) developed the Shuffled Frog Leaping Algorithm (SFLA), and similarly to Maier et al. (2003), applied their algorithm to a number of benchmark WDS optimisation problems. The SFLA was found to find the minimum known solution for the Alperovits and Shamir network (Alperovits and Shamir, 1977) in half as many evaluations as the previous best solution in the literature. However, it performed similarly to other algorithms for the Hanoi Network Problem (Fujiwara and Khang, 1990) and the New York City Water Supply Tunnels Problem (Schaake and Lai, 1969).

2.2 Genetic Algorithm Overview

GAs are probabilistic optimisation methods based on the theory of natural selection. These algorithms, along with a vast range of other EAs, are best suited to complex combinatorial problems where little is known about the fitness function, and are very difficult or impossible to convert into formulations that can be solved by traditional mathematical programming methods. A GA begins with a population of solutions with an associated fitness describing how well they solve the problem being solved. The best solutions in the population are selected for the next generation of the algorithm, analogous to survival of the fittest. Operators are then applied to recombine the current solutions to produce new ones, including most commonly crossover, which simulates mating in the evolutionary process, and mutation. This process is repeated for a number of generations until some stopping criteria is met, such as a given number of generations or if the population has converged to one solution.

The analogy with evolution and survival of the fittest has produced a great deal of different terminology that can be used to describe the different components of the algorithm. The values describing how well a solution solves a problem has been called the objective

function value, fitness function value, or the phenotype. The genetic analogy to the solution itself is a chromosome, and the values in the chromosome, genes. The individual values that make up a solution also have a number of names depending on the encoding used (see Section 2.2.1), bits for a binary coding, alleles for higher cardinality codings, or if a real coding is used, decision variables. In this work the terminology fitness function and decision variables is used, except in cases where there is a more appropriate term, for example if a binary coding is being described (Section 2.2.1), or if genetic theory is being reviewed (Section 2.3.3).

Assuming that the fitness function to be optimised is known, the two main decisions to be made in applying a GA are to select an encoding scheme for the decision variables and the operators that will alter the encoded values to generate new solutions. The parameters associated with each of the GA operators must then be calibrated to the fitness function, and some stopping criteria for the algorithm must be selected. This section begins by outlining the possible encoding schemes to select from, followed by the GA operators that exist, broken up into the three main categories of selection, crossover and mutation. An example of advanced operators that have been developed for GAs are also described.

2.2.1 Encoding Scheme

2.2.1.1 Binary Representation

The traditional representation for GA decision variables is a binary string (Goldberg, 1989b), where each variable is represented by a series of 0s and 1s. The binary bits are decoded into their actual values and then passed to the fitness function to evaluate each solution. The main advantage of this simple formulation is that it lends itself to theoretical studies of how the GA finds better solutions. Holland (1975) developed *Schema Theory*, which states that many small, above average, combinations of 0s and 1s in the solution string are identified in parallel and reinforced in the population through selection. As the generations continue, these small combinations are built upon, leading the algorithm to find better and better solutions. Goldberg (1991) suggested that by having an alphabet size of only two, the number of schemata per solution in the population will be maximised, hence a maximum level of efficiency in making use of good solutions, or parts of good solutions, that have been identified is achieved. The phenomenon of processing many partial solutions present in the population at the same time is known as *implicit parallelism*. Schema Theory is described in more detail in Section 2.3.1.

While there is a significant theoretical base for a binary representation, there are a number concerns when applying this encoding scheme to real valued decision variables.

The number of bits for each decision variable must be chosen based on the required accuracy of the solution, where greater accuracy requires more bits for each decision variable, increasing the dimensionality of the problem presented to the GA. Also, if the required range is not a power of two, redundant codes are introduced, thereby increasing the search space size unnecessarily.

The GA will consider all bits in the string equally important, however, the more dominant bits for a decision variable will overshadow any arbitrary values that the algorithm has converged on for less significant bits. For example, if a decision variable is represented by four bits, the first bit will be more significant than any of the others, and if a value of 1000 is changed to 0000, only one bit flip occurs, but there is a huge change in the value for the decision variable, greater than the contribution provided by the combination of the three following bits.

This property of binary encoding can also lead to the algorithm becoming trapped and being unable to move to better solutions that are nearby in the solution space. In this case, a number of bit flips are required to increase the value of the decision variable by the resolution provided by the encoding, known as *hamming cliffs*. Using the same four-bit example, to increase the decision variable from 0111 to 1000, increasing the value by the resolution provided by the encoding, three bit flips are required, which will be highly unlikely. A number of authors have suggested the use of Gray codes (Caruana and Schaffer, 1988) to overcome hamming cliff problems, where consecutive decision variables values are always one bit flip away from each other in the encoding scheme. However, Goldberg (1989a) suggests that doing so introduces higher order non-linearities with respect to recombination, which causes the degree of implicit parallelism to be reduced.

2.2.1.2 Integer Representation

To address some of the concerns with the binary representation above, an integer representation has been suggested, where each decision variable is encoded to be represented by an integer value. This will reduce the dimensionality of the problem presented to the GA compared to a binary representation, and therefore the opportunity for deception (Goldberg, 1989b) to take place, which occurs when fit lower order schemata do not combine to produce fit higher order schemata. Also, with one bit per decision variable, recombination operators cannot occur in-between a decision variable, breaking up good solutions that have been identified. This disruption that can occur in a binary representation may provide a mechanism to explore the whole range of each decision variable, and may need to be accounted for in an integer representation through the recombination operators with

increased exploration, for example, a higher mutation rate or mutation range.

As outlined above, the lower cardinal alphabets (binary representation) will produce more schemata, but more schemata, and more higher order schemata, are required to find good solutions, due to the higher dimensionality of the representation. Also, it is not clear how useful many of the inter-decision variable schemata will be, as many of them may be overshadowed by the more important bits for each decision variable. For an integer representation, smaller order schemata are between different decision variables, not intra decision variable which will occur with a binary representation. Consequently, this encoding scheme may pick up relationships between decision variables much more effectively, as there are fewer lower order schemata to confuse.

The main disadvantage with an integer representation for real valued decision variables is that the range of the variable must be discretised, with an associated resolution for each integer value. Similar to the number of bits for a binary representation, this introduces the problem of how many integers are required for each decision variable, with more values providing greater accuracy in the final answer, but also producing an increase in the size of the search space.

2.2.1.3 Real Value Representation

If the decision variables are real valued, is seems logical to apply the GA to operate directly on real values. This allows the algorithm to operate on the decision variables themselves, and minimises additional difficulties that can be introduced into the problem through an encoding scheme (such as hamming cliffs). This may allow the algorithm operators to perform more efficiently and exploit the *graduality* of the function (Herrera et al., 1998). For example, a mutation that changes a decision variable by a small step can operate as an effective hill climbing mechanism, and will not be prone to the discontinuities introduced by a binary representation.

Goldberg (1991) defined virtual characters and a virtual alphabet for Real Coded Genetic Algorithms (RCGAs). He suggested that a GA will converge quickly to solutions in basins of above average local optima, each one being a virtual character in the encoding scheme. The virtual alphabet is then a collection of virtual characters along a given dimension. With these definitions, Goldberg (1991) found the RCGA is consistent with the Schema Theorem developed for binary GAs, also determined by Wright (1991). Goldberg (1991) suggested that if a selectionist method is not presented with a low-cardinality alphabet one will be chosen implicitly, and therefore avoid the precision and aliasing problems that may occur with low-cardinality representations.

However, Goldberg (1991) also found RCGAs can be subject to blocking, where the algorithm will converge to a number of local optima in one dimension based on average values of the decision variables, but will then become trapped from moving to the better optimum in later stages as the other decision variables improve, as these better values are no longer in the population. To reduce the likelihood of the occurance of blocking within a RCGA, reproduction operators that are capable of producing values of the whole range of each decision variable may be desirable.

2.2.2 Selection

The selection operator simulates survival of the fittest in the evolutionary analogy, where the better solutions in the population are reinforced, to be recombined and identify other improved solutions, and poorer solutions are discarded from the search. While there are many variations of selection operators available, and the two most commonly used operators are proportional and tournament selection.

As part of proportional selection (Holland, 1975), solutions for the next generation are chosen based on a probability calculated from each solution's quality. However, the probability of selection for each solution is computed from the relative difference between the fitness function values and their absolute value, so in the later stages of the optimisation when all function values are similar, proportional selection provides only a very slight preference for the better solutions, as all solutions will have a very similar probability of selection. A number of methods have been developed to address this problem to a certain extent, such as introducing a selection pressure parameter to increase the difference between solutions, or ranking selection, where the probability for each solution is based on its rank in the population, rather than its actual fitness function value.

The other common approach to selecting the best solutions in the population is tournament selection, where a number of solutions are compared, and the best in the group, or tournament, is selected for the next generation. Hence, the solutions only need to be able to be compared against each other, and the actual fitness function values are not important. Typically, a tournament size of two is used, and in this case two copies of the best solution will appear in the next generation, no copies of the worst solution, and the number of solutions for the rest of the population will depend on the tournament pairings. Larger tournament sizes produce a higher selection pressure, and therefore more copies of the best solutions are present in the population after selection. A larger tournament size, and therefore higher selection pressure, can lead to a more efficient algorithm, as only the best solutions survive the selection process, however, it can also lead to premature

convergence of the GA, as the diversity in the population is decreased much quicker.

2.2.3 Crossover

Crossover is used to simulate mating in the evolutionary process, where the fittest individuals that have been selected produce offspring with their characteristics for the next generation. Crossover exploits solutions that have been identified previously by attempting to combine good parts of a number of solutions to produce new ones, and is therefore generally regarded the premier search operator for GAs.

The most common crossover operator is simple crossover, where parts of two solutions are interchanged at a crossover point. If there are two parent solutions, $P_1 = \{c_1^1, \ldots, c_n^1\}$ and $P_2 = \{c_1^2, \ldots, c_n^2\}$, then the two children for the next generation at random crossover point i are $C_1 = \left\{c_1^1, \ldots, c_i^1, c_{i+1}^2, \ldots, c_n^2\right\}$ and $C_2 = \left\{c_1^2, \ldots, c_i^2, c_{i+1}^1, \ldots, c_n^1\right\}$. Variations of the simple crossover operator include multi-point crossover, where there is more than one crossover point, and uniform crossover, where the value for the child solutions are randomly allocated from one of the parents' values. Multi-point crossover has a greater potential to identify and retain good decision variable values that are not adjacent in the solution string, where uniform crossover will ensure good mixing of the values found, but is likely to break up any higher order schemata that have been identified.

These crossover operators were developed for a binary GA, where each bit in the solution string can only be one value or the another (0 or 1), and therefore can be very limited when applied to decision variables that have a larger range (as is the case for RCGAs). Herrera et al. (2003) provide a review of crossover operators for RCGAs, and proposed four classes of crossover operators:

- Discrete Crossover Operators (DCOs), which do not alter the parent values, only recombine them;
- Aggregation-Based Crossover Operators (ABCOs), where the child values are obtained from a combination of the parent values;
- Neighbourhood-Based Crossover Operators (NBCOs), which generate the child values from a probability distribution defined around the parent values; and
- Hybrid Crossover Operators (HCOs), which can selectively choose from a number of operators belonging to different crossover classes.

In their empirical study of 18 crossover operators on a suite of 13 test functions, Herrera et al. (2003) found that discrete crossover operators performed very poorly when used with a RCGA. The ABCOs provided the best performance on average, while the NBCOs presented a favourable behaviour as well. However, many ABCOs use a weighted average

and produce solutions between the two parent decision variables, and while this is termed 'exploitation' by Herrera et al. (2003), the parents do not provide any information about this area when they are far apart in the search space. Hence, these operators may not be making good use of the values that have been found, and the good performance identified by Herrera et al. (2003) may be due to many of the test functions considered having their optimum in the centre of the search space. The authors identified the Dynamic Heuristic Crossover as the most promising operator. However, this is a relatively complex operator which makes use of the fitness function value of the parent solutions, and Beyer and Deb (2001) argue that the purpose of the recombination operator is to explore the search space, and use of the function value may lead to premature convergence.

Beyer and Deb (2001) performed a theoretical analysis on the effect of different crossover operators for RCGAs on the mean and variance of the population values, and compared their results against experimental simulations. With the task of exploration and exploitation in mind, the authors suggest that the mean value of each decision variable in the population should not be changed by the variance operators (both crossover and mutation). In terms of population variance, Beyer and Deb (2001) suggest that a tendency toward increasing the population variance is beneficial for the variance operators, depending on the fitness function and crossover operator used.

2.2.4 Mutation

The role of mutation in a GA is to restore lost or identify unexplored solutions into the population to prevent premature convergence of the GA to suboptimal solutions (Herrera et al., 1998). Mutation ensures that the probability of reaching any point in the search space is never zero.

For binary coded GAs, mutation involves a bit flip randomly in the solution string with a low probability. Creeping mutation can be used to fine tune solutions that have been identified, where a decision variable's value is changed up or down, rather than randomly over the whole range, which is generally the case with a standard mutation operator.

Higher cardinality representations, especially real codings, lend themselves to more advanced mutation operators. While many variations exist, these mutation operators generally consist of selecting a new value within a certain range around the original value, selected from a given probability distribution (uniform or normal for example). In many cases, this mutation range is decreased with the number of generations, hence initially the mutation is exploring over the whole range of each decision variable, whereas later in the optimisation process it is focused on tuning good solutions that have been identified.

Mutation is generally considered an exploration operator, as a new, random solution is generated in the search space. However, when combined with selection, mutation can behave similarly to a hill-climbing local search, where one decision variable is altered, and if it is an improvement it will be retained through selection, or rejected if it reduces the fitness of the solution (Goldberg, 1991).

A mutation operator that produces a new value over the whole range of each decision variable will overcome the blocking problem observed by Goldberg (1991) for RCGAs, however this can be very disruptive and should generally be used with a low probability. Also, the more advanced mutation operators usually introduce additional parameters, such as a mutation step size, which must be calibrated for each optimisation problem.

2.2.5 Advanced Operators

In the proceeding sections, the three main GA operators have been outlined, namely selection, crossover and mutation. A number of further advancements have been developed in an attempt to improve the effectiveness of GAs. Some of the more common advanced GA operators proposed are outlined below including: elitism, local search, niching operators and linkage learning.

2.2.5.1 Elitism

The most commonly used operator not already outlined in this chapter is elitism (De Jong, 1975). This involves reinforcing good solutions that have been found in the population to ensure they are not lost, and therefore continue to contribute to the search. In some cases, this reinforcement can lead the algorithm to areas of the search space that are not optimal. However, it is possible to prove that a GA will converge to the optimal solution with an elitism operator, but not without (Suzuki, 1995).

2.2.5.2 Local Search Operators

To be confident that a GA is at least converging to local optima in the search space, deterministic or heuristic hill climbing operators have been included in the GA methodology. These algorithms have been called *Memetic Algorithms* (Moscato, 1999), the analogy being that the solutions evolve culturally during their lifetime before they evolve genetically, where a *meme* is a unit of cultural evolution (Eusuff and Lansey, 2003). Incorporating a hill-climbing operator will ensure that all solutions in the population are local optima, however this is achieved at the cost of significantly more function evaluations. This may be the case early on in the search especially, when the population is well spread and many

of the local optima identified will not be useful to the search. Also, as mentioned previously, the mutation—selection combination may provide an effective local search method. The SFLA (Eusuff and Lansey, 2003) applied to a number of WDS problems contained elements of Memetic Algorithms.

2.2.5.3 Niching Operators

Niching operators have been developed to retain diversity in the population and identify a number of suitable solutions to an optimisation problem over the whole search space. The three main classes of niching methods are sharing, crowding (Mahfoud, 1995) and clearing (Petrowski, 1996). Sharing methods decrease the fitness of solutions in a similar area of the search space, or niche, so solutions that are in a highly populated area of the search space are not seen as fit by the algorithm are moved to other areas of the search space to increase diversity. A sharing method has been applied to the WDS rehabilitation problem, however it did not provide any improvement in the best solutions found (Halhal et al., 1997). Crowding methods organise the selection operator to choose between solutions that are closer together in the search space, in an attempt to maintain the diversity in the population. Clearing niching methods have a defined niche size, and if more than a specified number of solutions exist within a niche, the least fit solutions have their fitness value altered to ensure they are removed through selection. While niching methods attempt to provide a mechanism for the algorithm to continue searching the search space and identify a number of different solutions, most niching operators introduce a number of parameters that must be calibrated, such as the niche size, adding to the difficulty in correctly applying GAs.

2.2.5.4 Linkage Learning

Two crucial factors of GA success – a proper growth and mixing of good building blocks – are often not achieved (Thierens, 1995). If two interacting decision variables are not located near each other in the solution string, the building blocks containing good combinations of values for the two variables may be continually broken up by a one-point or multi-point crossover operator. A number of GA operators have been developed in order to identify these building blocks and retain them in the population so better solutions can be found. The first operator developed to improve the processing of building blocks was inversion and other reordering operators (Goldberg, 1989b), where the order of the bits in the solution string are changed, to bring variables that interact with each other together to allow them to be mixed more efficiently. However, generally the time required for an in-

version operator to mix the best partial solutions in the population is much longer than the time available before the population will converge due to selection (Goldberg and Bridges, 1990). More recent methods have tried to identify the linkage and exploit it, rather than relying on a random reordering of the solutions, including evolving the representation of the string along with the solution to the fitness function (Goldberg et al., 1989) or using a probabilistic representation of the population, and using the joint distributions to estimate the linkage between genes (Harik, 1999; Pelikan et al., 1999). While the combination of good gene values is crucial to finding a good solution, identifying and exploiting linkages has proven to be a difficult undertaking (Harik, 1999).

2.3 GA THEORY

A great deal of theoretical work has been undertaken to understand the inner workings of GAs. Eiben and Rudolph (1999) provide an overview of the different approaches that have been adopted for all types of EAs, and suggest that the key topics in evolutionary computation theory are: the limit behaviour, or whether an algorithm can guarantee to converge to an optimal solution or prematurely converge to a local optimum; dynamic behaviour of the population, or the change in the distribution of function values as the GA run progresses; and predicting the run time of the algorithm. A number of tools have been used in an attempt to investigate different aspects of these topics, and are outlined in the remainder of this section.

2.3.1 Schema Theory

Schema Theory represents the first attempt to explain the behaviour of a GAs (Holland, 1975). Schema Theory relies on the idea that a good solution can be constructed by combining good pieces, or building blocks, from different solutions in the population (Lobo, 2000). The Schema Theorem was initially defined for binary coded GAs, however it has since been extended to RCGAs (Goldberg, 1991; Wright, 1991). For the binary coded case, the schemata are constructed from the values $0, 1$, as well as a wild symbol, $*$. For example $H' = 1**0**$ represents all strings with a 1 for the first variable, and a 0 for the fourth variable. Therefore, strings 110011 and 100001 both belong to schema H', however 010011 and 100111 do not. Two important definitions arise from this definition of schemata: the *order* of the schema, or the number of fixed positions; and the *defining length* of the schema, or the distance between the two outermost fixed position. Therefore, the example schema H' has order 2 and defining length 3.

Based on these definitions of schemata, Holland (1975) proposed the Schema Theorem to quantify the change in the number of schemata for each generation:

$$m(H, t+1) \geq m(H, t)\frac{\overline{f}(H, t)}{\overline{f}(t)}\left(1 - p_c\frac{\delta(H)}{l-1} - p_m O(H)\right) \tag{2.1}$$

where $m(H, t)$ is the number of schema H at time t, $\overline{f}(H, t)$ is the average function value of schema H at time t, $\overline{f}(t)$ is the average function value of the population at time t, $\delta(H)$ is the defining length of schema H, $O(H)$ is the order of schema H, p_m is the probability or mutation, p_c is the probability or crossover, and l is the problem size. The Schema Theorem in Equation 2.1 is specific to GAs with proportionate selection and one point crossover, however it can be generalised to other variants of the algorithm.

The general implication of the Schema Theorem is that selection emphasises the fittest schemata, and the variation operators destroy some of these schemata. The main result from the theorem is that the highly fit schemata that are not too disrupted by the variation operators, those that are of a short order and defining length, tend to grow from generation to generation (Lobo, 2000). These short, highly fit schemata are combined with other short, highly fit schemata to produce even better solutions as the GA search progresses. This has long been believed to be the advantage of GAs, as every generation many of these small sub-solutions are evaluated and combined, many more than the N solutions in the population that have actually been computed. This effect is called *implicit parallelism* (Holland, 1975).

The Schema Theory was considered fundamental to the understanding of GAs until the early 1990s (Eiben and Rudolph, 1999). Eiben and Rudolph (1999) propose a number of reasons for this change of opinion: the schema theory cannot explain the dynamic or limit behaviour of GAs; it is implicitly assumed that the problem is separable to some extent; and the application of Markov chain theory to the field of evolutionary computation provided a more detailed analysis.

2.3.2 Markov Chain Theory

Since the population of an GA only depends on the state of the previous population in a probabilistic manner, Markov chains may be used to model and analyse GAs (Eiben and Rudolph, 1999). Lozano et al. (1999) and Suzuki (1995) used Markov chain analysis to prove that GAs incorporating an elitist operator are guaranteed to asymptotically converge to the global optimum of a fitness function. Interestingly, this cannot be done for simple GAs that do not include elitism (Lozano et al., 1999). Suzuki (1995) and He and Kang (1999) used Markov Chain Analysis to determine bounds on the convergence rate for

simple GAs. He and Kang (1999) derived that the convergence rate for a GA with time-invariant genetic operators is bounded by:

$$||u_t - \pi|| \leq (1 - \delta)^{g-1}, \tag{2.2}$$

where u_t is the probability distribution of the population at generation g, π is an invariant probability distribution and $0 < \delta \leq 1$.

Although the entire information about the evolutionary process is contained in the Markov chain transition matrices, the degree of aggregation is too high to allow a simple derivation of detailed answers to particular questions, for example the expected time of visiting the optimum solution for the first time (Eiben and Rudolph, 1999). Consequently, only simple versions of GAs and general results have been successfully produced by this approach.

2.3.3 Quantitative Genetics

As GAs are derived from observations about natural selection, it is not surprising that results from Quantitative Genetics have been applied to the GA field. One of the most common relationships used in GA theories is the response of a population to selection. Provided that the phenotypic values (fitness function values) are normally distributed, and the selection is by truncation, the response to selection can be predicted by (Fisher, 1930):

$$S = i\sigma_{\mathrm{P}}, \tag{2.3}$$

where S is the selection differential, or the difference in the mean phenotypic value between the offspring and the parents ($\overline{f_{g+1}} - \overline{f_g}$), i is the selection intensity, and σ_{P} is the standard deviation of the fitness function values for the population at time g. Equation 2.3 has been used in a number of studies (Lobo, 2000; Mühlenbein and Schlierkamp-Voosen, 1993; Rogers and Pruegel-Bennett, 1999; Thierens and Goldberg, 1994; Thierens et al., 1998) to predict the convergence of GAs. Thierens and Goldberg (1994) used this relationship with other types of selection, such as proportionate and tournament selection, and Thierens et al. (1998) considered fitness functions with more complex distributions (exponentially distributed), both experimentally validating their approaches. Thierens and Goldberg (1994) and Thierens et al. (1998) used Equation 2.3 to suggest that the number of generations before convergence for a particular optimisation problem was a function of the string length, where the order of the relationship depends on the salience of the variables. Lobo et al. (2000) extended this analysis to include the number and size of building blocks into the relationship. A number of these studies resulted in rules for GA

calibration, and are reviewed in more detail in Section 2.4.3.

Further results from Quantitative Genetics may be useful to explain the dynamic behaviour of GAs. Equation 2.3 suggests that convergence occurs when $\sigma_{\mathrm{P}} = 0$, (i.e., when there is no more diversity in the phenotype values). Due to the influence of selection, σ_{P} changes from generation to generation in accordance with (Falconer, 1981):

$$\sigma_{\mathrm{P}}^{'} = \sigma_{\mathrm{P}} k, \tag{2.4}$$

where k is a constant with $0 < k < 1$. Therefore, at any generation, g:

$$\sigma_{\mathrm{P,g}} = \sigma_{\mathrm{P,0}} k^{g}. \tag{2.5}$$

Falconer (1981) identified a number of components that comprise the total phenotypic variance. The total variance is the phenotypic variance, as used in Equation 2.3, and is the sum of the separate components of variance:

- Genotypic variance, (σ_G^2);
- Breeding or additive variance (σ_A^2);
- Dominance variance (σ_D^2);
- Interaction variance (σ_I^2); and
- Environmental variance (σ_E^2).

The total variance in the population is the sum of these components, in the form (Falconer, 1981):

$$\begin{aligned} \sigma_P^2 &= \sigma_G^2 + \sigma_E^2 \\ &= \sigma_A^2 + \sigma_D^2 + \sigma_I^2 + \sigma_E^2. \end{aligned} \tag{2.6}$$

These results concerning the change in genetic variance are the basis for the GA calibration methodology proposed in this work, as outlined in Section 1.2.

2.3.4 Dimensional Analysis

Dimensional analysis has been used to identify the important dimensions, or key features, of GA behaviour, to establish a functional relationship between them (Eiben and Rudolph, 1999). Features of GAs that have been considered include iterated selection, crossover, and mutation operators, which are put into a functional relationship and generally validated by simulations. A number of studies using a dimensional analysis approach have produced rules and relationships for GA calibration, which are considered in more detail in Section 2.4.3.

2.4 GA Calibration Methods

As outlined in Chapter 1, before a GA can be applied, the user must specify a number of parameter values, such as population size, probability of crossover, and probability of mutation. Several studies have shown that a GA can tolerate some variation in its parameter values without dramatically affecting overall performance (Lobo, 2000). However, even though the GA is quite robust with respect to some parameters settings, it does not imply that the GA will work well regardless of the settings. If a poor choice of values is made, the GA will not perform well. The No Free Lunch Theorem (Wolpert and Macready, 1997), stipulates that a global set of GA parameters that will be effective for every optimisation problem does not exist.

Typically, GA parameter values are found using a trial-and-error approach. This involves manually tuning the parameter values, where all parameters are initially given typical values, and then one parameter at a time is changed until a suitable set of values is identified. However, the GA operators will interact with each other, and therefore the behaviour produced by the value of one parameter will be dependent on all the others, and therefore cannot be considered separately. Due to computational requirements, only a limited number of parameter combinations can be trialled; therefore it is unlikely that sensitivity analyses will identify parameter values close to those that will produce the best performance from the selected algorithm. As the GA parameter values are generally different for each problem, it might be expected that the best values would be related to characteristics of the problem.

A number of problem characteristics have been identified that affect the difficulty of a problem in the context of GAs. These include (Naudts and Kallel, 2000): *isolation* or *deception*, where the global optimum is not located near the best of the set of local optima; *multimodality*, or the number of local optima present in the fitness function to potentially trap the GA at sub-optimal solutions; and *epistasis*, or the degree of interaction between decision variables, requiring accurate processing of combinations of values in the population. A highly epistatic problem has many interactions between a number of the decision variables. If something was known about these characteristics, this knowledge could be used to determine appropriate GA parameters. However, often these characteristics are unknown, and GA parameters must be found using other methods.

A number of alternate approaches have been developed in an attempt to provide information on the best GA parameters to use for a given problem. The different methods typically fall into three categories: i) empirical experiments; where results from GA experiments are use to develop empirical relationships, ii) dimensional analysis; where sim-

ple models are used to represent how certain aspects of a GA will behave, or iii) parameter control; where the GA parameter values are changed as the GA is solving the problem. A number of comprehensive GA calibration methodologies have been proposed based on results from some of these approaches.

Firstly in this section, methods that have been developed to measure characteristics of the fitness function are outlined, followed by an outline of the different approaches that have been developed to calibrate GAs to optimisation problems. The GA calibration methods that arise from a number of these studies are outlined in Section 2.4.6.

2.4.1 Measuring Optimisation Problem Difficulty

A number of statistics have been developed to measure different characteristics of fitness functions. Examples include the fitness distance correlation (Jones and Forrest, 1995), correlation length (Weinberger, 1990), epistasis variance (Davidor, 1991), operator correlation (Manderick et al., 1991), schema variance (Radcliffe and Surry, 1995) and hyperplane ranking (Mathias et al., 1995). However, for each example that indicates that any of these measures provide information about the difficulty of an optimisation problem, there appear to be as many counter examples displaying that statistic's unreliability.

One of the main reasons that these measures have been so unreliable is that no rigorous definition of the concept of difficulty is available in the framework of GAs (Kallel, 1998). Early attempts to characterize difficulty in the context of GAs propose criteria based on isolation, deception, and multimodality. It is clear that isolation and deception contribute to problem difficulty, regardless of the algorithm chosen (Naudts and Kallel, 2000). However, there exist examples that show that epistasis and multimodality are neither necessary nor sufficient to make a problem difficult (Naudts and Kallel, 2000).

The main reason for the ambiguity of the term 'problem difficulty' is that the performance of an algorithm on a problem is dependent on the algorithm itself, and a number of papers (Culberson and Lichtner, 1996; De Jong et al., 1995; Guo and Hsu, 2003) stress the futility of speaking of problem difficulty without considering the algorithm, and all the parameters of the algorithm (Kallel, 1998). For example, on a given problem a GA coupled with a local search may find the optimum every time regardless of the initial population, while the same GA without the local search operator may find the problem very difficult. Such differences are unmeasurable by any problem difficulty measure that does not directly take into account how the selected algorithm finds improved solutions.

Studies that attempt to demonstrate the unreliability of fitness function statistics (including Kinnear, 1994; Kallel et al., 1999; Manela and Campbell, 1992; Naudts, 1998;

Naudts and Kallel, 2000; Quick et al., 1998; Reeves and Wright, 1995a; Rochet et al., 1998) have drawn conclusions about the convergence of the algorithm, without considered the calibration of the GA parameters. It might be expected that optimisation problems with different characteristics would be solved more effectively by GAs that behave differently, as controlled by the GA parameters. Therefore, it is likely that the best performing GA parameter values are related to the characteristics of the problem, as opposed to the performance of a GA with arbitrary parameter values.

The characteristics of a certain *fitness function* are contained in the *fitness landscape*. The fitness landscape, first introduced by Wright (1932), describes the search space of an optimisation problem as a multidimensional landscape defined by the possible solutions, through which the optimisation moves, mapped to the corresponding fitness value (Smith et al., 2002). The fitness landscape can be visualised as all the possible solutions to a problem mapped into the x-y plane, with solutions that are close together in this plane being those that are most likely to be created from one another through the application of the genetic operators employed (Kinnear, 1994). The fitness of each of these solutions is plotted on the z-axis, creating a surface where the peaks are the locations of solutions with poor fitness, and the basins show the locations of the better solutions (for a minimisation problem). Discovering the best solution to the problem then becomes equivalent to searching over this landscape for the deepest basin (Kinnear, 1994). This is opposed to the *fitness function*, which is a more traditional way to consider the optimisation problem, where variables of similar values are mapped next to each other, as opposed to values that are most likely to be created from the GA operators. However, for many GA operators, it might be expected that the fitness function and fitness landscape are very similar.

The parameters used by the algorithm will affect how solutions are mapped next to each other in the fitness landscape, and therefore will have a large impact on the characteristics of the landscape. An effective parameter set may produce a relatively smooth landscape, while a comparatively poor parameter set will produce a very rough landscape, making it very difficult to locate the lowest basin and producing a high likelihood that the algorithm will become trapped in a local optimum. Properties of the fitness landscape such as these are directly related to the difficulty of the optimisation problem, which many researchers have attempted to define and characterise.

While the characteristics of fitness landscapes are a function of the GA implemented and its corresponding parameter values, there will be characteristics of the fitness function that will have an impact on what is required by any algorithm. Unless a particularly problem specific algorithm is applied, characteristics such as the roughness of the fitness

function and the number and size of local minima present in the fitness function could be expected to have an impact on the search progress, irrespective of the algorithm used.

A number of statistics have been proposed to provide information about the characteristics of a fitness function that have been identified to affect GA performance, such as isolation, deception, multimodality and epistatic interactions. It is unlikely that any statistics will be useful in determining isolation or deception in a problem, as there is no useful information about this characteristic in the function. In fact, an optimisation algorithm will only find the best solutions that are isolated from other good areas in the search space by chance. The different statistics that have been developed can be grouped into what they provide information about, either the structure or multimodality of the landscape, or the degree of epistasis between the variables.

2.4.1.1 Fitness Function Structure

Eremeev and Reeves (2002) set out to characterize the roughness of a function by estimating the total number of local optima from a sample of local optima found in the search space. From the estimated total number of local optima in the search space, the average basin size of the local optima can be derived, to provide an indication of the roughness of the search space. For Memetic Algorithms, Merz (2004) proposed a local search escape analysis by the use of random walks, to estimate the effectiveness of recombination relative to that of mutation, based on the structure in the fitness function.

The most common statistics to provide information about the structure of the fitness function are correlation measures. Correlation measures are known to be closely related to the 'ruggedness' of a fitness function (Manderick et al., 1991; Stadler and Happel, 1999; Weinberger, 1990). A number of correlation measures have been proposed, most notably the autocorrelation function (Weinberger, 1990) and the Fitness Distance Correlation (FDC) (Jones and Forrest, 1995). The FDC computes the correlation between the fitness function values of local optima that are a similar distance from the global optimum. In computing this statistic, the location of the global optimum must be known, and therefore it is not practical use it used prior to the optimisation of an unknown function. Also, the location of a number of local optima in the fitness function must be known, resulting in a high computational demand to compute the statistic.

The autocorrelation function (Weinberger, 1990) can be easily applied to any function, does not require any prior knowledge of the fitness function (such as location of the global optimum as for the FDC), and does not require a large number of function evaluations, which would be the case for identifying local optima. The autocorrelation of a function

is calculated from a set of fitness function values, correlated in relation to their arrangement in the search space. The traditional approach to express this arrangement is by a time series, where function values are sampled along a random path through the fitness landscape, producing a 'time series' of function values. This approach to computing the autocorrelation of a function has been termed the temporal autocorrelation function in this work, to provide a distinction to the spatial methods proposed in Chapter 4. The series of function values can then be correlated using the standard autocorrelation function:

$$\rho(i) = \frac{\sum_{j=1}^{n-i} (x_j - \overline{x})(x_{j+i} - \overline{x})}{\sum_{j=1}^{n} (x_j - \overline{x})^2}, \tag{2.7}$$

where n is the number of samples, x_j is the j^{th} value in the series of function value samples, and $\overline{x}$ is the average of the samples. An important assumption made in this analysis is that the fitness landscape is statistically isotropic (Hordijk, 1997), so regardless of the direction taken by the random walk, the autocorrelation calculated will be the same.

The correlation length is the average distance between points in the search space where a positive correlation is still only just detectable. This statistic provides an indication of the average roughness of the landscape, as a long correlation length indicates a smoother landscape on average, and *vice versa*. The correlation length may also provide an indication of the average size of the basins of attraction of the local optima, which will be inversely related to the number of local optima. For example, a short correlation length may indicate that there are many local optima, each with a small basin of attraction.

This approach to computing the autocorrelation of a fitness function has a number of shortcomings, including: (i) only a small portion of the search space may be covered; and (ii) the potential exists for values to be a large number of steps apart in the generated time series, but actually be very close to each other in the search space. A fundamental shortcoming of the time series approach is that spatial data have to be converted into a representative sequence of temporal data. Originally, Weinberger (1990) defined the correlation of a function in terms of the distance between points, and in practice this distance was computed using random walks, and the relationship between the correlation values produced using the two different approaches was defined for AR(1) functions. However, most optimisation problems do not fall into this class of functions. In order to address some of the concerns raised when using random walks to determine the distance between points, the use of spatial autocorrelation statistics, based on the true distance between points in the search space, is introduced in Chapter 4.

2.4.1.2 Epistatic Interactions

The other problem characteristic that fitness function statistics have attempted to quantify is the degree of epistatic interactions between variables. An epistatic function is more complex than a simple summation of the relationship between each decision variable and the fitness function value, and is therefore derived from combination of the decision variable values. Davidor (1991) proposed epistasis variance to measure the epistasis of a problem, which in simple terms is a measure of the distance between a separable approximation to the fitness function and the fitness function itself. Reeves and Wright (1995b) improved upon the epistasis measure proposed by Davidor (1991), by using Walsh Transforms to perform the linear decomposition of the fitness function. These methods have only been applied to fitness functions composed of binary strings, however Mason (1991) used Partition Coefficients to extend the Walsh Transformation approach to non-binary alphabets. There are a number of concerns regarding the application of epistasis variance to quantify the epistasis of a fitness function, namely: i) the statistic can only provide information about the presence of epistasis, not a measure of the degree of epistasis; and ii) the epistasis measure is sensitive to nonlinear scaling of the fitness function, something a GA with tournament or ranking selection is entirely insensitive to (Naudts and Kallel, 2000).

Seo et al. (2003) proposed a method to measure interactions between decision variables based on entropy (Shannon, 1948). The measure is called gene epistasis, and it provides an indication of the combined effect of two decision variables on the fitness function value.

Entropy provides a measure of how much uncertainty, or randomness, there is in a signal or random event. The Mutual Information (MI) between two variables can be computed by considering the entropy of each random variable and their joint entropy, to determine the reduction in uncertainty (Cover and Thomas, 1991). The MI of two random variables provides a measure of the dependence between the two variables, and is calculated using:

$$I(X,Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \tag{2.8}$$

where X is the alphabet of x, $p(x)$ is the probability density function (pdf) of x, Y is the alphabet of y, and $p(x,y)$ is the joint pdf of x and y. If X and Y are independent, then X contains no information about Y and *vice versa*, so $I(X;Y) = 0$. If X and Y are identical, then all information conveyed by X is shared with Y, therefore the mutual information is the same as the information conveyed by X (or Y) alone, hence $I(X;Y) =$

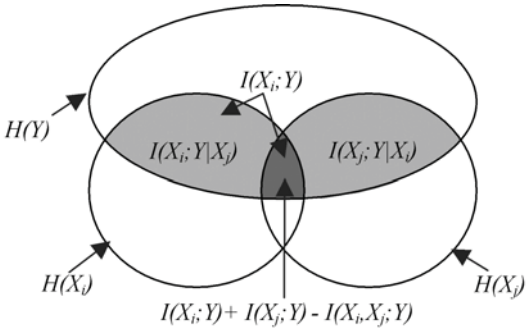


Figure 2.1 Representation the interaction between decision variables in terms of Mutual Information.

$H(X)$, where $H(X)$ is the entropy of X. This relationship is depicted in Figure 2.1, where the interaction between decision variables is represented by the dark shaded region, $I(X_i;Y) + I(X_j;Y) - I(X_i, X_j;Y)$.

In terms of an optimisation problem, if the relationship between one decision variable, X_i, and the fitness function value, Y, is independent of another decision variable, X_j, then $I(X_i;Y)+I(X_j;Y) = I(X_i, X_j;Y)$. Otherwise, there is an epistatic interaction between the two decision variables X_i and X_j, and their mutual information regions overlap, as shown in Figure 2.1. For this case, $I(X_i;Y) + I(X_j;Y) > I(X_i, X_j;Y)$, and the degree of interaction is given by the difference between the sum of the individual mutual information values, $I(X_i;Y)+I(X_j;Y)$, and the joint mutual information, $I(X_i, X_j;Y)$.

Based on the MI terms, the gene epistasis can be computed using (Seo et al., 2003):

$$\epsilon_{ij} = \begin{cases} 1 - \frac{I(X_i;Y)+I(X_j;Y)}{I(X_i,X_j;Y)} & \text{if } I(X_i, X_j;Y) \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

It can be seen that the gene epistasis only considers pairs of decision variables, and therefore larger building blocks are not identified. However, if an interaction is detected between all combinations of three variables, x_i, x_j, x_k, then it would be expected that there is a building block of order 3 between these variables. The analysis can be continued to identify building blocks of higher orders. Seo et al. (2003) defined the problem epistasis, η, to be the mean absolute value of the gene epistases of all gene pairs:

$$\eta = \frac{1}{l(l-1)} \sum_{i=1}^{l} \sum_{j=1}^{i-1} |\epsilon_{ij}|. \tag{2.10}$$

2.4.2 Empirical Studies of GA Parameters

While it is generally accepted that the characteristics of the fitness function affect the performance of GAs, the link between function characteristics and GA theory has not yet been well developed (Merz, 2004). The simplest method for determining useful relationships considering all aspects of a GA is to perform empirical analyses on a range of test functions. De Jong (1975) performed a systematic parametric test on five fitness functions with different characteristics, such as unimodal and multimodal, separable and epistatic, concave and convex, and deterministic and noisy. The parameters included in the study were the population size, probability of crossover, probability of mutation and generation gap, which allowed for a certain percentage of the population to remain intact over each generation. Based on the results of the study, De Jong (1975) recommended a population size of $50 - 100$, probability of crossover of 0.6, probability of mutation of 0.001, and generation gap of 1.0 (the whole population should be evolved every generation). These results were subsequently applied to a large number of applications, and lead to the 'standard settings' for GAs (Lobo, 2000).

Schaffer et al. (1989) performed a similar analysis on the five functions used by De Jong (1975), as well as five more functions. Somewhat similar recommendations were made, where a population size of $20 - 30$, probability of crossover of $0.75 - 0.95$, and a probability of mutation of $0.005 - 0.01$ produced the best results. Schaffer et al. (1989) also observed a relationship between the population size and the probability of mutation, where a small population size performed the best with a large probability of mutation, and *vice versa*.

Empirical relationships provide realistic results for the situations they have been applied to, however, the applicability of the results is limited to the situations that have been considered, and may not extend to the general case. Whitley et al. (1995) provide an example of this, where test functions that have previously been used in empirical experiments are *separable functions*, where there are no epistatic interactions between decision variables. Without the epistatic interactions between the decision variables, each decision variable can be optimised independently of the value of the other decision variables. Separable functions are often readily solved by local search methods (Whitley et al., 1995), and are most likely unrepresentative of more realistic problems that would be tackled in GA applications. At best, empirical analyses can provide some useful results about the classes of functions that have been considered in the analyses.

2.4.3 Dimensional Analysis

A number of theoretical modelling studies have been performed to provide an insight into the complex mechanisms at work in a GA run, to assist with the calibration of certain parameters. To enable any meaningful results to be drawn, generally the models can consider the effect of only one or two aspects of a GA. Some of the more important studies consider convergence due to genetic drift (Mühlenbein and Asoh, 1994; Rogers and Pruegel-Bennett, 1999), convergence due to selection (Thierens and Goldberg, 1994; Thierens et al., 1998), the interaction between selection and crossover (Goldberg et al., 1993; Lobo, 2000), population sizing (Goldberg et al., 1992; Harik et al., 1999), and the effect of mutation alone (Bäck and Schwefel, 1993; Mühlenbein, 1992).

Due to the repetitive application of a selection operator in GAs, the population will eventually converge to a single solution. This process is termed *genetic drift*. Mühlenbein and Asoh (1994) determined that the number of generations before the population will converge to an arbitrary solution due to genetic drift was $g_{drift} \approx 1.4N$, where N is the population size. However, the genetic drift model used does not include a mutation operator to introduce new values into the population, and the constant value in the relationship $g_{\text{drift}} \approx 1.4N$ is the expected value with a large variance (Thierens et al., 1998). Rogers and Pruegel-Bennett (1999) present a method of calculating the rate of genetic drift in terms of the change in population fitness variance, $\sigma_P^{'} = k\sigma_P$, similar to that outlined in Section 2.3. k is found in terms of the population size, N, and for a simple GA with tournament selection $k = 1 - \frac{1}{N}$.

Goldberg et al. (1993) developed two simple models in terms of the selection pressure, s, and probability of crossover, p_c: one to represent the time required by crossover to mix the population and locate the optimal solution, the other to represent the time until the population will converge due to selection. Therefore, if the selection time is greater than the mixing time, it is expected that the GA will have time to form good solutions, otherwise the GA will converge prematurely. Based on these models, Goldberg et al. (1993) developed *control maps* for values of s and p_c, where combinations of values for s and p_c inside the boundaries on the map lead to good GA performance. Lobo (2000) observed that the growth of building blocks is given by $s(1 - p_c)$. Therefore, in order to satisfy the Schema Theorem (Section 2.3.1), any combination of values that provides a growth greater than one will ensure that the number building blocks in the population will grow at each successive generation.

The relationship governing the convergence of a population during selection, given in Equation 2.3, has been used in a number of studies (Lobo, 2000; Rogers and Pruegel-

Bennett, 1999; Mühlenbein and Schlierkamp-Voosen, 1993; Thierens and Goldberg, 1994; Thierens et al., 1998) to predict the convergence of GAs. Thierens and Goldberg (1994) and Thierens et al. (1998) used simple models of GA convergence to relate the number of generations before convergence occurs, g_{conv}, to the length of the solution string, l. Thierens and Goldberg (1994) considered normally distributed fitness functions to derive that the number of generations before convergence, g_{conv}, is of the order $O(\sqrt{l})$, where l is the length of the solution string. As a specific case, the OneMax problem is considered, where expressions can be derived for $\overline{f(g)}$ and σ_g as a proportion of the genes that have converged to the optimal value, 1. The relationship $g_{\text{conv}} = \frac{\pi}{2}\sqrt{\pi l}$ is derived for this problem with a tournament size of two, and validated experimentally with tournament selection and uniform crossover.

Thierens et al. (1998) provided a similar analysis for arbitrarily distributed fitness functions, where some building blocks have a higher marginal fitness contribution than others. As an extreme case, exponentially scaled fitness functions were considered. For these functions, there exists a convergence window in the solution string of decision variables that are converging at a given time during a GA run, where decision variables before the convergence window have fully converged to a single value, and decision variables after the convergence window do not contribute enough to the fitness function value to experience selection pressure, and have not yet started to converge (Thierens et al., 1998). Therefore, the decision variables converge in order of their position along the solution string, a phenomenon known as *domino convergence*. For exponentially scaled fitness functions, it was determined by Thierens et al. (1998) that the number of generations before convergence, g_{conv}, is of the order $O(l)$. The BinInt problem is considered as a specific case, as relationships can be derived for $\overline{f(g)}$ and σ_g, and the relationship $g_{\text{conv}} = 1.76l$ is derived for this problem. The model is experimentally validated, however, an upper limit to g_{conv} must be taken into consideration, which occurs when the population prematurely converges due to genetic drift.

Goldberg et al. (1992) modelled the supply of building blocks in the initial population, along with making the correct decision between building blocks during a GA run, to determine a relationship for the optimal population size. The relationship proposed by Goldberg et al. (1992) was later refined by Harik et al. (1999). The relationship was derived by drawing an analogy between a random walk in one dimension and the initial supply and correct selection of building blocks in a GA population. The analogy uses the gamblers ruin model in the analysis, and the following relationship was obtained (Harik

et al., 1999):

$$N = -2^{k_{\text{BB}}-1} \ln \alpha \frac{\sigma_{\text{BB}} \sqrt{\pi m_{\text{BB}}}}{d}, \tag{2.11}$$

where N is the population size, k_{BB} is the building block size, α is the probability that the GA selects a sub-optimal building block, σ_{BB} is the building blocks' fitness variance, m_{BB} is the number of building blocks, and d is the building blocks' fitness signal. The relationship was validated experimentally for relatively simple test functions where the required parameter values could be determined. However, generally this is not the case, and little is known about the parameters in Equation 2.11 for most optimisation problems. Also, the derivation assumed that all building blocks must be present in the initial population, and that mutation and crossover do not create new building blocks but only align them. However, the relationship does identify a number of fitness function characteristics that affect the most suitable population size.

Mühlenbein (1992) and Bäck and Schwefel (1993) performed separate theoretical investigations on the effects of the mutation operator. Both studies concluded that for a fixed mutation rate throughout the run, the optimal mutation rate in the case of a unimodal problem is $1/l$, where l is the length of the solution string.

While theoretical dimensional analyses can provide useful insights into certain aspects of GA behaviour, the approach is limited to relatively simple algorithms and problems (Merz, 2004). Conclusions must be drawn based on simplified models, while ignoring beneficial GA attributes. Also, the vast majority of GA modelling has been conducted using binary coding, as it is much simpler to perform the analyses when each bit in the solution string can only be in one of two states. For problems where the decision variables are real values, each decision variable can take any value over a specified range, making the modelling of these algorithms much more complex. Similarly, the inclusion of all GA operators, for example both crossover and mutation, as well as other operators that have shown potential, such as elitism, further complicates the modelling, and it very quickly becomes extremely difficult, if not impossible, to develop realistic models.

2.4.4 Parameter Control

Parameter control methods have been developed to address some of the shortcomings in attempting to preset parameter values, where the GA parameters are adjusted during the optimisation process in an attempt to provide a more efficient search (Beyer and Deb, 2001). As the parameter values can change, the reliance on finding the 'optimal' set of parameters before applying a GA is reduced, and it allows different GA behaviour at different stages of the optimisation. Also, self adaptive GAs have been shown to adapt to

changing fitness landscapes, displaying the ability to quickly escape the current optimum found and proceed towards the new global optimum when the fitness landscape changes (Beyer and Deb, 2001). This property may be useful in the case of a constrained problem with dynamic penalty functions, where the values of the penalty constraints are changing during the evolutionary process, and therefore so is the fitness function.

A review of parameter adaptation methods is provided in Eiben et al. (1999). The authors divide the parameter control methods into three classes: *deterministic parameter control*, where the values change in accordance with a predetermined rule; *adaptive parameter control*, where the values are changed based on the performance of the algorithm; and *self-adaptive parameter control*, where the parameters are built into the optimisation problem for the GA to determine along with solving the optimisation problem itself. The major advantage of parameter control is that it allows the parameter values to change as the GA is solving a problem. It might be expected that different GA behaviour will be suitable at different times in the optimisation run, for example a large amount of mutation may be required at the beginning of a GA run to explore the search space, however toward the end this may be disruptive to the search progress, and smaller amounts may be more beneficial. Each of the classes of parameter control are described in more detail in the remainder of this section.

2.4.4.1 Deterministic Parameter Control

The most common method for adjusting parameter values online is to use a deterministic function, usually dependent on the number of generations. While many alternatives exist, Bäck and Schütz (1996) proposed:

$$p_m(g) = \left(2 + \frac{l-2}{G} \cdot g\right)^{-1}, \tag{2.12}$$

where the probability of mutation is decreased from $p_m(0) = 0.5$ initially to $p_m(G) = 1/l$, producing the 'optimal' static value found by Mühlenbein (1992), where g is the generation number, G is the maximum number of generations, and l is the string length.

The main advantage of dynamic function parameter control is that the changes to the parameters are predetermined and predictable. However, this can also be a disadvantage, as changes are caused by a deterministic rule triggered by the number of generations, without taking any consideration of the actual progress in solving the problem (Eiben et al., 1999). Also, the form of the parameter control rule must be defined, and a suitable selection of the functional form may be just as difficult as selecting a static value in a parameter tuning case. However, Eiben et al. (1999) suggest that sub-optimal selection

of a deterministic parameter control rule often leads to better results than a sub-optimal selection of a static parameter value.

2.4.4.2 Adaptive Parameter Control

Adaptive parameter control methods take the actual search process into account by incorporating feedback from recent population changes into the parameter adaptation method. One of the most common adaptive methods is Rechenberg's $1/5$ success rule (Rechenberg, 1971), which was developed for use with Evolution Strategies and has been adapted to RCGAs in some cases. The rule states that the ratio of successful mutations should be one in five, and if the ratio is greater than this the mutation step size should be increased, or decreased if there are not enough successful mutations.

Recently, a number of feedback based adaptive methods have been developed to alter the population size. Eiben et al. (2004) proposed a method to adjust the population size based on how frequently improvements in the best solution were found. They concluded that their method was the most effective population adjustment method available, however, it introduces four more parameters that must be calibrated, the increase factor, decrease factor, minimum and maximum population sizes.

Feedback based adaptive parameter control methods are designed to supervise the optimisation process, generally by monitoring improvements in the fitness of the solutions found. It is also possible to consider other measures, such as the diversity of the population and whether it is converging or diverging (Beyer and Deb, 2001). While this approach would be expected to provide a more robust algorithm that can recalibrate itself if it is under performing, rules must be developed to determine which parameters to change, how much to change them by and how often the changes should occur. This can be a difficult task, as the impact of each GA parameter is not well understood and suitable values will be different for different optimisation problems.

2.4.4.3 Self-adaptive Parameter Control

Setting GA parameter values can be considered a poorly structured, poorly defined, complex problem, and GAs are generally considered to perform better than other methods on these types of problems (Eiben et al., 1999), therefore there may be an advantage in allowing a GA to calibrate itself. This can be achieved by incorporating the GA parameters into the solution to the problem, so each individual in the population contains values for the set of decision variables for the optimisation problem, as well as a set of parameter values for the GA. As the optimisation progresses, the good solutions are retained in the

population, as well as the good parameters that produced these solutions, enabling the algorithm to find better solutions with the most effective parameters. The most common method involves including the probability of mutation or mutation step size in the solution (Bäck, 1992; Hinterding, 1995; Srinivas and Patnaik, 1994), which is then applied to the solution it is attached to, and evolved along with the decision variable values.

Eiben et al. (1999) suggested that it is likely that using self-adaptation methods such as this is the most promising method of control, as the evolution itself will determine the beneficial interactions among the GA operators, while also finding a near-optimal solution to the problem. However, this is also the major drawback to the self-adaptive calibration approach, entrusting the algorithm to make the best choices for itself. It is not clear how effective a GA will be at addressing both the calibration and optimisation problems in parallel. The effect of constantly changing the parameter values over a large range is also unclear. Incorporating the parameter values into the solution will increase the number of decision variables to be optimised, increasing the size of the search space, making it more difficult to find the optimal solution to problems which are most likely already very difficult. Also, even though a self-adaptive methodology provides the mechanism for different parameters values at different times during a GA run, it is unlikely to produce different parameter values to perform better later in a GA run, after the algorithm has converged to one set of values initially.

2.4.5 Supervisory Algorithms

In a similar approach to self-adaptive parameter control, optimisation algorithms have been used to optimise the parameters for a GA that is applied to an optimisation problem. A meta-GA approach has been proposed (Grefenstette, 1986), where a higher-level GA is used to identify a good set of parameters for a lower level GA. The higher level GA runs on a population consisting of parameter values, while the lower level GA is a regular GA that uses the settings found by the higher-level GA (Lobo, 2000).

However, this approach introduces the problem of what parameters are used for the meta-GA? It would be expected that setting GA parameters would be a more straight forward optimisation problem than optimising the fitness function itself, hence Boeringer et al. (2005) implemented a supervisory gradient type algorithm to adapt the number of crossover points, mutation rate and mutation range to maximise the fitness improvement each generation. The authors found their method required fewer function evaluations compared to a calibrated static GA for the fitness functions considered.

2.4.6 GA Calibration Methodologies

Based on a number of the results outlined in Section 2.4.3, Harik and Lobo (1999) proposed the 'Parameterless GA'. The relationship between selection pressure and the probability of crossover to ensure the growth of building blocks, outlined in Section 2.4.3, is used to set $s = 4$, and $p_c = 0.5$, producing a net growth of building blocks of 2. A mutation operator is not included in the GA implemented. The population sizing algorithm proposed by Harik et al. (1999) was not implemented in this calibration method, as the parameters in Equation 2.11 are not able to be estimated for arbitrary problems. To remove the population sizing problem from the user, the initial population size is set to a very small value, such as $N = 4$, and after the population converges, the population size is doubled and reinitialised, with a copy of the best solution found in the previous GA run inserted into the population. The process is then repeated until no improvement in solution quality is observed from one GA run to the next, or a user specified criterion is met.

However, the process of doubling the population size can suffer from genetic drift, and if one population size is slow to converge, this can disadvantage the search. To overcome this problem, Harik and Lobo (1999) proposed a race against concurrent GAs with different population sizes, and if a bigger population size finds a better solution than a smaller population size, the GA run with the smaller population size is stopped, and reinitialised with an even bigger population size. More function evaluations are given to the GAs with smaller population sizes, until they converge or are overtaken by the GAs with larger population sizes. The authors found the Parameterless GA to be an effective self-calibrating method, however significantly more computational effort than for an optimally calibrated static GA was required.

A similar GA calibration approach is proposed by Minsker (2005). The major difference is how the initial population size is determined. Instead of an arbitrary small initial population size ($N = 4$), the initial GA population size is determined using a combination of the relationships $g_{\text{conv}} = 1.76l$ (Thierens et al., 1998), in the form of $g_{\text{conv}} = 2l$, and the number of generations before the population will arbitrarily converge due to genetic drift, $g_{drift} \approx 1.4N$ (Mühlenbein and Asoh, 1994). By setting $g_{\text{conv}} = g_{drift}$, the initial population size is then given by $N = 1.4l$. A relationship between s and p_c that is similar to that suggested by Harik and Lobo (1999) is used, however, tournament selection with $s = 2$ is used, and the maximum allowable value of p_c to ensure building block growth is used (i.e., $p_c = 0.5$). After the initial GA has converged, the population size is doubled, however the race against concurrent GA runs with different population sizes to avoid excessive fitness

function evaluations is not implemented. Minsker (2005) included a mutation operator in the approach, with the probability of mutation taking on the maximum value out of $1/l$, as proposed by Mühlenbein (1992) and Bäck and Schwefel (1993), and $1/N$.

The calibration methodologies outlined above are based on theory developed for binary coded GAs. Reed and Yamaguchi (2004) proposed a similar calibration methodology for real valued decision variables, specifically Differential Evolution (Storn and Price, 1997), however, the authors suggest that the approach can be easily adopted for any RCGA. A similar methodology to those proposed by Harik and Lobo (1999) and Minsker (2005) is used to set the population size, where a small population size is used initially ($N = 10$ is recommended), and the population size is doubled after a minimum number of generations have been computed, given by $g = Nl$, and a 1% increase in the best fitness function value found is no longer obtained. Values for the mating and mutation operators are adopted from Storn and Price (1997), and are shown to have little affect on the algorithm's performance for the test function considered.

The GA calibration methodologies reviewed in this section are the most practical, comprehensive, and theoretically sound calibration methods available. However, the most critical GA parameter, the population size, is found from an iterative approach, it is not tailored to each individual problem. A number of studies (Cantú-Paz and Goldberg (2003), for example) suggest that one GA run with a large population is more likely to find better solutions than a number of GA runs with smaller population sizes, which is essentially the approach taken in the case of sequentially doubling the population size. While the population size does increase as the GA run progresses, the other algorithm parameters are fixed, no matter what the characteristics of the fitness function are (with the exception of the mutation rate proposed by Minsker (2005), which is taken as the maximum value out of $1/l$ and $1/N$). Hence, the characteristics of fitness functions that affect GA performance, such as multimodality and epistatic interactions, are ignored.

2.5 Summary and Proposed Methodology

From the review of the literature presented in this chapter, it can be seen that the application of GAs to WDSs is not a new concept, and has been achieved successfully in many cases. Despite this, a strong background in GAs is still required to apply the methods, and a great deal of time and effort is spent to calibrate the algorithm to each problem.

This problem is not specific to the WDS field; it is not clear how to calibrate a GA to any fitness function, not just WDS optimisation. As indicated by the sheer volume of research invested in GA calibration methods, it is one of the major concerns in applying

GAs to an optimisation problem. While parameter tuning is by far the most common approach, it requires a large computational effort to determining parameter values, and even this approach is unlikely to produce the best possible results. Adaptive and self-adaptive parameter control methods allow feedback from the GA run to alter the parameter values, however, a suitable method to adapt the parameters is not available, and it is not clear how effective a GA is at calibrating itself in parallel to solving an optimisation problem. A number of the results from dimensional analysis studies have been used in the GA calibration methodologies proposed by Harik and Lobo (1999), Minsker (2005) and Reed and Yamaguchi (2004). However, these are not based on the characteristics of the optimisation problem, and a number of sequential GA runs may not be the most effective use of the fitness function evaluations that are available before a solution is required.

From the review of the literature regarding GA calibration methods presented in this chapter, it is clear that a complete GA calibration methodology based on the characteristics of the fitness function does not exist. This is a concern, as practical experience tells us that each problem is best solved with different GA parameters, or at least that there is not a global set of parameters that are suitable for all problems. Therefore, it is the aim of this thesis to provide a practical GA calibration methodology, based on the characteristics of the function to be optimised. The remainder of this section contains details of the GA selected for the work undertaken in this thesis, based on the review provided in Section 2.2, followed by an overview of how the review of the literature presented in this chapter is related to the calibration methodology proposed in Section 1.2, including some important assumptions that have been made.

2.5.1 GA Adopted for This Research

Based on the outline of GAs presented in Section 2.2, the following operators have been selected for the GA used in this research:

- The decision variables are represented by real coding, as all the optimisation problems considered in this thesis have real valued decision variables. This approach has avoided any further complexities being added to the problem by the encoding scheme, such as hamming cliffs, or discretisation issues.
- Tournament selection, as only the absolute difference in objective function values is considered , thereby producing a constant selection pressure, rather than the relative difference, which is the case with proportional selection. A tournament size of two has been used throughout this work.
- A one-point distributed crossover operator has been used, as neighbourhood-based

crossover operators such as this have been found to exploit the numerical nature of RCGAs (Herrera et al., 2003). The crossover operator used for this work is similar to the Simulated Binary Crossover operator Deb and Agrawal (1995) and the Fuzzy Recombination Crossover operator Voigt et al. (1995). A one-point crossover operator has been selected, as a statistic to determine tightly linked decision variables is developed in Chapter 4, and this operator facilitates the processing of epistatic interaction between adjacent variables, more so than uniform or multipoint crossover operators. If crossover occurs, a random crossover point is generated, and a new value of each decision variable is generated from a normal distribution centred around the first parent's solution values, p_1, before the crossover point, or centred around the second parent's solution values, p_2, after the crossover point. The spread of the distribution is determined from a parameter related to the distance between the two parent solutions. The distribution used for a decision variable can be seen in Figure 2.2.

- A uniformly distributed mutation operator, producing any value over the range of each each decision variable. A Gaussian mutation operator is commonly used for RCGAs, however, this would provide a very similar search mechanism to the crossover operator adopted for this work. This operator has been selected as it has the potential to generate values all over the search space, at any time in the search process and reduce the potential for blocking. Also, a uniform distribution is used as it does not introduce additional parameters to be calibrated. A range of mutation probabilities has been considered, thus if uniformly distributed mutation is proving to be disruptive to the search, a small or zero probability of mutation will be the most effective.
- An elitist strategy to preserve good solutions, as Markov chain modelling of GA behaviour suggests that an elitist operator is necessary to ensure convergence to the global optimum. While convergence to the global optimum may not occur in practice, the results indicate that the operator is beneficial to the search process.

Many more operators could be considered, such as those outlined in Section 2.2.5, and may provide a greater search potential. However, the main focus of this research is to reduce the effort and knowledge required for the calibration of GAs, and more advanced operators will make the application of these methods even more difficult for the non-expert, and introduce even more parameters to be calibrated for each optimisation problem. Therefore, for the operators that have been selected, the following parameters are produced:

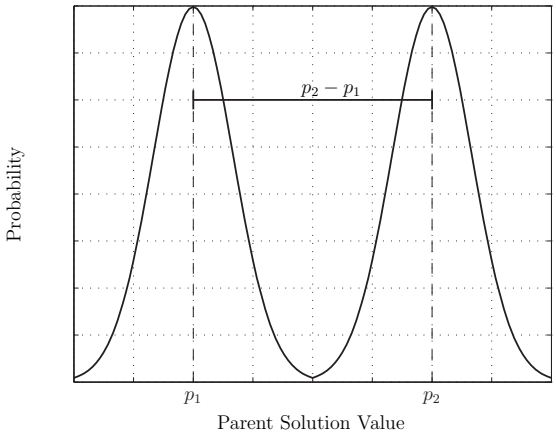


Figure 2.2 Probability distribution used for crossover

- Population size, N;
- Probability of crossover, p_c;
- Number of elite solutions, e;
- The probability of mutation, p_m, used as the probability that mutation is applied to a solution string. If mutation is applied, the decision variable to be mutated in that string is randomly chosen. Therefore, $p_m = 1$ for each string corresponds to the empirical mutation rule for a binary-coded GA under bitwise mutation, $p_m = 1/l$ (Mühlenbein, 1992);
- The standard deviation of the distribution to be used for the crossover operator, σ, taken as a fraction of the distance between the two parent values, c. A smaller fraction of c will produce a tighter distribution around the parent values, and therefore greater exploitation of current solutions. A crossover distribution using the value $c = 6$ is presented in Figure 2.2, and therefore the Gaussian standard deviation used for the crossover operator was $\sigma = |p_2 - p_1|/6$.

2.5.2 Relevance of the Literature

The main theories that the proposed methodology is based on are reviewed in Section 2.3 and Section 2.4.3. Most importantly, Equation 2.5 is used, which states that the variance in the fitness function values will decrease by a constant value each time the selection operator is applied. This relationship is derived from Quantitative Genetics, however it has also arisen from theoretical studies of GAs, including in Markov chain analysis studies (He and Kang, 1999), as well as the results of Rogers and Pruegel-Bennett (1999). Therefore, there is a strong theoretical basis for making use of this relationship. The outcome

from this relationship is that a GA will converge in a certain number of generations for a given fitness function; that there is an optimal number of generations to solve a given problem. Chapter 3 is dedicated to experimentally validating this observation.

The other result that the proposed methodology draws upon is that the best GA parameter values are related to the characteristics of the fitness function. This can be observed in a number of the relationships that have been reviewed in Section 2.4.3. The results of Thierens and Goldberg (1994) and Thierens et al. (1998) suggest that the number of generations is related to the size of the problem, and that the order of the relationship is related to the salience of the variables in the fitness function. Therefore, a novel dominance measure, based on the mutual information between the decision variable values and the fitness function value, is proposed in Chapter 4.

Equation 2.11, proposed by Harik et al. (1999), suggests that the population size is a function of the size and number of the interactions between the variables, and the signal to noise ratio in the fitness function. The signal to noise ratio is expected to be quantified by the proposed spatial correlation measure, and to provide information regarding the interactions between the variables, a novel separability measure is proposed.

Equation 2.11 suggests that there is an optimal population size for a fitness function with certain characteristics, as opposed to a constant number of generations, as proposed in this work. However, this difference can be explained by the α parameter in Equation 2.11, the probability that the GA will make a mistake when selecting building blocks. If the number of function evaluations is increased, and a constant number of generations is used, the population size will increase proportionally. If N is increased in Equation 2.11 there is a corresponding increase in $-\ln \alpha$, and therefore a decrease in α, suggesting that the GA will have a lower probability of selecting the incorrect building blocks, and therefore finding better solutions. This result agrees with one of the observations that the proposed methodology is based on; that a larger population size will locate better solutions provided there is time for the population to converge.

It is reassuring that the characteristics of fitness functions that have been identified to influence the most suitable GA parameter values also correspond to the components that influence the genetic variance (Falconer, 1981), outlined in Equation 2.6. The development of the statistics necessary to quantify these components is given in Chapter 4. The relationship between these statistics and the optimal number of generations is considered in Chapter 5, where the functional form of the relationship is developed based on the relevant dimensional analysis results (Harik et al., 1999; Thierens and Goldberg, 1994; Thierens et al., 1998).

In order to undertake the necessary investigations, a number of important assumptions must be made. The first assumption is that the fitness function is a reasonable representation of the fitness landscape, where the difference between the two is discussed in Section 2.4.1. It is believed that this is a reasonable assumption, as it is more likely that solutions close together in decision variable space will be generated by the genetic operators that have been adopted.

It is assumed that the fitness function is relatively regular, that is that a statistic computed as an average from samples taken over the whole search space provides a reasonable representation of the fitness function. The statistics are computed using the same information that the GA has to solve the problem, namely a random sample of solutions in the search space, consequently it is assumed that the statistics provide useful information about the fitness function the GA is applied to.

Another important assumption is that selection is the only operator that will influence the change in variance of the population. This assumption is based on the fact that the crossover operator used in this work is normally distributed around the original values, and therefore on average the variance will not be changed by this operator. For the other operators that may alter the population variance (mutation increasing the variance and elitism decreasing the variance) the operators are applied with a low probability, and therefore will not have a significant impact on the population variance.

The proposed methodology is focused on the population size, in terms of the number of generations before the population will converge. While rules are used to determine the values of the remaining GA parameters, it is assumed that the population size has the greatest influence on the performance of the GA. This assumption is based on a number of previous results (Lobo, 2000; Sadegheih, 2006), and is verified to have the greatest effect on the quality of the solutions found using the results from the empirical studies undertaken in Chapter 6.

This chapter has presented the background relevant to the work proposed in this thesis. Based on this review, the need for a GA calibration methodology based on the characteristics of the fitness function is evident, and the theory behind the proposed methodology has been outlined. Chapter 3 is dedicated to the experimental validation of the first hypothesis of the proposed methodology: that there exists a given number of generations to solve a certain fitness function.

Chapter 3
The Number of Generations Until Convergence

For WDS optimisation, and most real-world optimisation problems, it is unlikely that the optimal solution is ever found due to an extremely large, complex, search space. Therefore, a GA can only aim to find the best possible solution in the time available to find a solution. Therefore, the calibration methodology proposed in this thesis revolves around providing the best search behaviour possible, before the GA must converge.

It is generally accepted that for GA optimisation, a larger population size will locate a better solution than a smaller population size, provided there is time available for the GA to converge. Therefore, as the number of available function evaluations to solve the problem, FE, increases so does the best population size, N. As the number of generations g, is give by $g = FE/N$, if FE and N increase at the same rate, this observation suggests that the most efficient number of generations of the GA population to solve the problem may perhaps be constant. The relationship $g = FE/N$ assumes that every solution is evaluated every generation, irrespective of whether the solution has been subject to crossover or mutation or not.

A number of dimensional analysis studies, such as those by Thierens and Goldberg (1994) and Thierens et al. (1998), support this observation, suggesting that $g_{\text{conv}} \propto \sqrt{l}$ and $g_{\text{conv}} \propto l$, respectively. The result of their work was that there is a constant number of generations before a GA will converge on a given fitness function, and the number of generations is a function of the problem size, l. A number of other theoretical results, reviewed in Chapter 2, also support this observation (He and Kang, 1999; Mühlenbein and Schlierkamp-Voosen, 1993; Rogers and Pruegel-Bennett, 1999).

If a GA will converge in a certain number of generations, g_{conv}, for a given fitness function, and the time available to find a solution, expressed as the number of available

fitness function evaluations (FE) is known, then the most suitable population size can be calculated using $N = g_{\text{conv}}/FE$. The ability to predict the number of generations before convergence, g_{conv}, would therefore be extremely beneficial to assist GA calibration, as the most appropriate population size could then be determined using the number of function evaluations that are available. The population size is widely regarded as the most influential GA parameter, so obtaining an accurate estimate for the most efficient population size is the basis of the GA calibration methodology proposed in Chapter 6.

The first hypothesis of this thesis is that the number of GA generations before a GA will converge for a given optimisation problem is constant for changes in the number of fitness function evaluations. The first section of this chapter is dedicated to testing this hypothesis on a number of mathematical optimisation problems. The second part of this chapter investigates if the number of generations before convergence changes with controlled changes to the characteristics of a number of fitness functions.

3.1 Observing an Optimal Number of Generations

The aim of this section is to empirically test the hypothesis that there exists a constant number of generations, g_{conv}, to most efficiently solve a given optimization problem, irrespective of the convergence criteria. In this work, to 'most efficiently solve' an optimisation problem is defined as locating the best values observed for a given value of FE. All the problems considered in this work are minimisation problems, therefore the best value observed is the lowest value found.

A large parametric study has been conducted to allow the best performing GA parameters to be identified. By considering the results for a number of different function evaluations, the best population size, and therefore the optimal number of generations, can be identified for different convergence criteria. If the observed number of generations is similar for different convergence criteria, then the hypothesis of a constant number of generations is validated for the cases considered.

As this thesis has focused on the calibration of GAs to the optimisation of WDSs, it would be desirable to perform this analysis on WDS optimisation problems. However, the simulation of WDSs for extended periods of time is extremely computationally intensive for systems of any realistic size. Therefore, the computer run times involved in the proposed analysis renders the direct application to WDS optimisation prohibitive, even if run simultaneously on many of the fastest processors available. Therefore, in order to allow the proposed hypothesis to be tested, mathematical optimisation functions have been used. By considering problems with different characteristics, for example different de-

grees of epistatic interactions and multimodality, it is proposed that the results generalise to a wide range of optimisation problems, including WDS optimisation problems. Different problem sizes are also considered to investigate if there is a relationship between g_{conv} and the problem size, l. The following section outlines the methodology used to test the hypothesis, including the test functions and GA parameter values used.

3.1.1 Methodology

An empirical approach has been adopted to test if there exists a constant number of generations before convergence, g_{conv}, for selected test functions with different characteristics. This approach allows for realistic results to be obtained, and, by considering the characteristics of the test functions, the results obtained are likely to extend to other problems with similar characteristics. To determine g_{conv}, the GA used throughout this research, outlined in Section 2.5.1, has been calibrated by means of a large-scale parametric study. The number of generations before convergence has been determined by identifying the best performing GA parameter values every $1\,000$ function evaluations, as it is assumed that the best solutions are identified by the GA converging to them, as opposed to randomly locating a good solution by chance. To support this assumption, each set of GA parameter values that has been tested was run with 30 different sequences of random numbers, with the average fitness function value found from these runs taken as the solution found.

3.1.1.1 Test Functions

Table 3.1 lists the test problems that have been adopted for this research. In an attempt to cover a range of problem types, the analyses have been applied to test functions with different characteristics. F1 is the sphere function (De Jong, 1975), F2 is the Rastrigin Function (Mühlenbein and Schlierkamp-Voosen, 1993), F4 is a variation of Griwank's Function (Whitley et al., 1995), F5 and F6 are functions proposed by Rochet et al. (1998), and F7 is the sine-envelope sine-wave function (Schaffer et al., 1989). These functions have been selected as they provide a diverse mix of problem characteristics, including: structure toward the optimal solution, such as 'big bowl' problems (e.g., F1 and F2), compared with relatively flat search spaces (F7); differently scaled contributions from each decision variable (F5 and F6); and separable (e.g., F1 and F2) and non-separable functions (F4 and F7). All functions are simply scaled into higher dimensions, with the exception of F7, which has been extended into higher dimensions by summing sub-functions of the original 2-dimensional problem.

Along with these common benchmark optimisation problems, F3 has been constructed

Table 3.1 The test functions used in the parametric study, and the interval used for the decision variables.

	Function to be Minimised	Interval
F1	$\sum_{i=1}^{l} x_i^2$	$[-5.12, 5.11]$
F2	$10l + \sum_{i=1}^{l} \left(x_i^2 - 10\cos\left(2\pi x_i\right)\right)$	$[-5.12, 5.11]$
F3	$\sum_{i=1}^{l-1} \left(\left(\frac{x_i + x_{i+1}}{2} \right)^2 - 10\cos\left(2\pi \frac{x_i + x_{i+1}}{2} \right) \right) + x_l^2 - 10\cos\left(2\pi x_l\right) + 10l$	$[-5.12, 5.11]$
F4	$1 + \frac{1}{4000} \sum_{i=1}^{l} x_i^2 - \prod_{i=1}^{l} \cos\left(\frac{x_i}{\sqrt{i}} \right)$	$[-512, 511]$
F5	$\sum_{i=1}^{l} i \left\lvert x_i \right\rvert$	$[-100, 100]$
F6	$\sum_{i=1}^{l-1} i x_i + x_l^2$	$[0, 100]$
F7	$\frac{l}{4} + \sum_{i=1}^{l/2} \left(\frac{\sin^2\left(\sqrt{x_{2i-1}^2 + x_{2i}^2}\right) - 0.5}{1 + 0.001\left(x_{2i-1}^2 + x_{2i}^2\right)} \right)$	$[-100, 100]$

using the guidelines proposed by Whitley et al. (1995). The guidelines allow a non-linear, non-separable, scalable function to be developed, which provides a realistic challenge to any optimisation algorithm. The Rastrigin Function (F2) has been used as a basis, and to produce the non-separable component of the function, which introduces interactions between the decision variables, the x_i terms have been replaced with $(x_i + x_{i+1})/2$. The final function developed can be seen in Table 3.1 as F3, where the sub-function in variable l has been included to ensure that there is a unique solution to the problem. This function allows for a direct comparison with F2 to investigate the effect of epistatic interactions on the optimal GA parameters.

The interval considered in the optimisation for each function can be seen in Table 3.1, which has been taken from the relevant studies outlined above for each function. Four different problem sizes have been considered for each of the seven functions in Table 3.1, namely $l = 5$, 10, 20, and 30 dimensions.

Table 3.2 The GA parameter values used for the parametric study.

Parameter	Values
Population Size, n	10, 50, 100, 200, 400, 800
Probability of Mutation, p_m	0, 0.2, 0.4, 0.6, 0.8 , 1.0
Probability of Crossover, p_c	0.7, 0.85, 1.0
Distance for the Standard Deviation of Crossover, c	6, 18
Elite Solutions per Generation, e	0, 1

3.1.1.2 Parametric Study

To determine the most efficient number of generations for the GA for each problem considered, a large-scale parametric study has been undertaken. While the values for the GA parameters other than population size do not influence the number of generations required to solve the problem, they must be included in the calibration of the GA to ensure that they are not biasing the results toward a certain population size. The parameter values tested are given in Table 3.2. Each combination of parameter values was tested 30 times with different sequences of random numbers to evaluate the performance. This resulted in a total of $12\,960$ GA runs for each problem size considered for each function. Four problem sizes have been considered for each of the seven test functions tested, resulting in a total of $362\,880$ separate GA runs required to produce the results presented in Section 3.1.2.

Each individual GA was run until a solution within 10^{-6} of the actual solution ($F(\mathbf{x}) = 0$) was found, or a maximum of $500\,000$ function evaluations. The analysis for a set of GA parameter values concluded once the average from the 30 different runs was within 10^{-4} of the optimum solution, as this slightly higher convergence criterion (compared to 10^{-6} used for individual GA runs) allowed for some random effects, such as one initial population for a combination of GA parameters taking especially long to converge. There were two functions that did not converge to the optimal solution in the permitted function evaluations, bounded by the GA parameter values used in the parametric study. The first was F3, where, as the number of function evaluations was increased, the optimal population size increased until the maximum tested, $N = 800$, performed the best. In this case, the analyses were stopped once the optimal population size had reached this maximum population size, as it would be expected that beyond this, larger population sizes would outperform those considered. The second function was F7, which did not converge to

the optimum solution for the problem sizes considered. Consequently, the convergence criterion was the $500\,000$ function evaluations that were available to solve the problem.

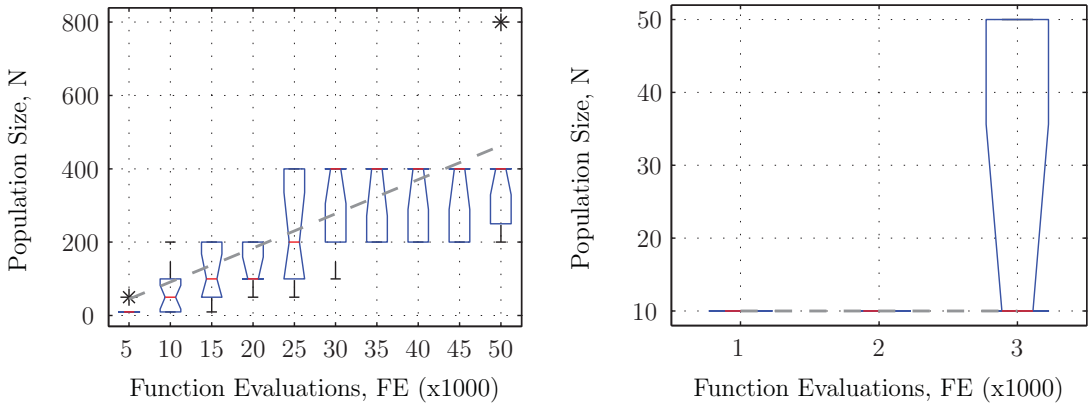
The average and standard deviation of the fitness function values of the best solution found over the 30 different GA runs for each GA parameter set were recorded every $1\,000$ function evaluations. A 2-tailed Student's t-test with a 95% confidence interval was used to compare parameter sets and identify those that were statistically the best as the GA solved each function. Of the best performing population sizes found by the Student's t-test, the median was used as the optimal population size. From the optimal population sizes identified, the optimal number of GA generations can be computed from the number of function evaluations that have been made. The hypothesis of this section is that as FE is increased, N will increase proportionally, producing a constant g_{conv}. The following section presents results from the parametric study for the four problem sizes considered for each of the test functions.

3.1.2 Parametric Study Results

A graphical illustration of the method used to determine the optimal number of generations for F3 with $l = 5$ is shown in Figure 3.1(a). All of the parameter combinations that find the current best solution (not necessarily the optimal solution), as determined from the Student's t-tests, are plotted as box plots against the number of function evaluations made. The median of each box plot has been taken as the optimal population size. The variations in population size are produced by different combinations of the other GA parameters (probability of crossover or probability of mutation, for example) producing statistically similar results. The inverse of the slope of the straight line fitted between the optimal population sizes (the median of the box plots) and the number of function evaluations produces the optimal number of generations to solve the problem.

Two different relationships were observed from the parametric study for the test functions considered, the results of which are given in Table 3.3 and Table 3.4. For the functions that are presented in Table 3.3, the hypothesis that there exists a constant g_{conv} is confirmed, as there is an approximately constant number of generations that most efficiently solves the problem for different convergence criteria. For these functions, there was a clear relationship between N and FE: as FE is increased, N also increased. This linear relationship produces an optimal number of generations, g_{conv}, to solve the problem, as $g_{\text{conv}} = FE/N$.

The optimal number of generations, g_{conv}, for each problem size, l, of the functions considered that follow this relationship, namely F3, F4, and F6, can be seen in Table 3.3.



(a) Optimal population sizes for F3 ($l = 5$)

(b) Optimal population sizes for F1 ($l = 5$)

Figure 3.1 The different results obtained for the Optimal Generation Functions and the Maximal Generation Functions.

Table 3.3 Optimal Generation Function results.

Function	l	FE	g_{conv}	g_{conv}/l	$g_{\text{conv}}/\sqrt{l}$
F3	5	–	75	15.0	33.5
	10	–	107	10.7	33.8
	20	–	219	11.0	49.0
	30	–	325	10.8	59.3
F4	5	61 000	121	24.2	54.1
	10	41 000	109	10.9	34.5
	20	63 000	121	6.1	27.1
	30	89 900	236	7.9	43.1
F6	5	13 000	91	24.6	49.2
	10	36 500	240	23.6	52.8
	20	76 000	408	19.3	60.9
	30	117 000	660	17.9	80.3

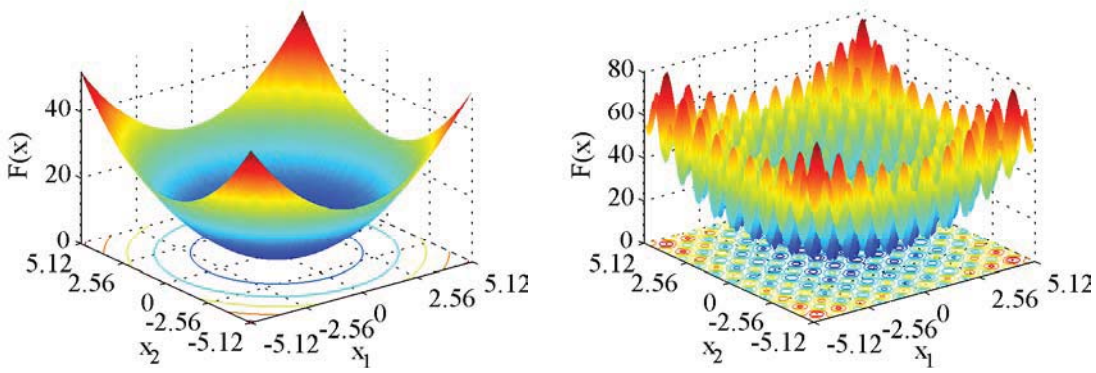
Table 3.4 Maximal Generation Function results.

l	$F1$	$F2$	$F5$	$F7$
5	4 000	11 000	4 000	–
10	8 000	19 000	5 000	–
20	18 000	48 000	5 000	–
30	26 000	74 000	7 000	–

The number of function evaluations, *FE*, taken to find the optimal solution and the proportionality constants for the $g_{\text{conv}} \propto l$ and $g_{\text{conv}} \propto \sqrt{l}$ relationships, as proposed by Thierens and Goldberg (1994) and Thierens et al. (1998) are also given in Table 3.3. This set of functions has been termed *Optimal Generation Functions*, as the most efficient way to solve the functions is to use a constant number of generations, and adjust the population size based on the number of function evaluations that are available.

For the remaining test functions considered, F1, F2, F5, and F7, a different relationship between N and FE was observed. The best performance for these functions was obtained with the smaller population sizes, irrespective of the number of function evaluations. The number of function evaluations, FE, required to solve each function for the different problem sizes considered is shown in Table 3.4. The optimal N for F1 with $l = 5$ can be seen in Figure 3.1(b), where the best performing population sizes every $1\,000$ function evaluations is given, until the optimal solution is found after $FE = 3\,000$. For $FE \leq 2\,000$ it can be seen that only $N = 10$ performs the best, and for $FE = 3\,000$, $N = 50$ also performs well. The horizontal line at $N = 10$ is fitted through the optimal population sizes, as determined from the median of the box plots, indicating that $N = 10$ is the most suitable population size for this function. Similar optimal population size plots were also obtained for the other test functions shown in Table 3.4 which display this relationship. This set of functions has been termed *Maximal Generation Functions*, as the most efficient way to solve the functions is to use the smallest possible population size, maximizing the number of generations available to update the population.

Section 3.1.2.1 discusses the characteristics of the Maximal Generation Functions, before a similar discussion on the Optimal Generation Functions in Section 3.1.2.2. The effect of increasing the size of an optimisation problem on the optimal number of generations is considered in Section 3.1.2.3, before Section 3.1.2.4 discusses the effect of epistatic interactions on the optimal number of generations. These results lead to the



(a) 2–dimensional plot of F1 (b) 2–dimensional plot of F2

Figure 3.2 Functional form of F1 and F2.

identification of four classes of fitness function characteristics, that each have an optimal number of generations to solve the problem. These calibration classes are defined in Section 3.1.3, along with a discussion of the wider implications of these results.

3.1.2.1 Maximal Generation Functions

The functions that were most efficiently solved with the smallest population sizes considered in the parametric study were F1, F2, F5, and F7 (Table 3.4). The problem characteristics that lead to this result are discussed in the remainder of this section.

F1 A two–dimensional plot of F1 can be seen in Figure 3.2(a). This function has an equal contribution from each decision variable to the fitness function value, therefore it might be expected that the $g_{\text{conv}} \propto l$ relationship would be observed, as proposed by Thierens and Goldberg (1994). However, from Figure 3.3, it can be seen that the smallest population size always produced the best results, irrespective of the number of function evaluations made for all problems sizes considered. As F1 does not have any interactions between decision variables, each decision variable can be optimized independently of the others. Therefore, it is not necessary to store combinations of the decision variable values, and function evaluations are wasted on poor quality solutions stored in a large population. Also, as each decision variable has the same contribution to the fitness function, a change in any of the decision variables will affect the fitness function value, and therefore there is no need to retain a large degree of diversity, potentially contained in a large population. Hence, it is more efficient to have a smaller population size, which is updated frequently through more generations, for this completely separable function.

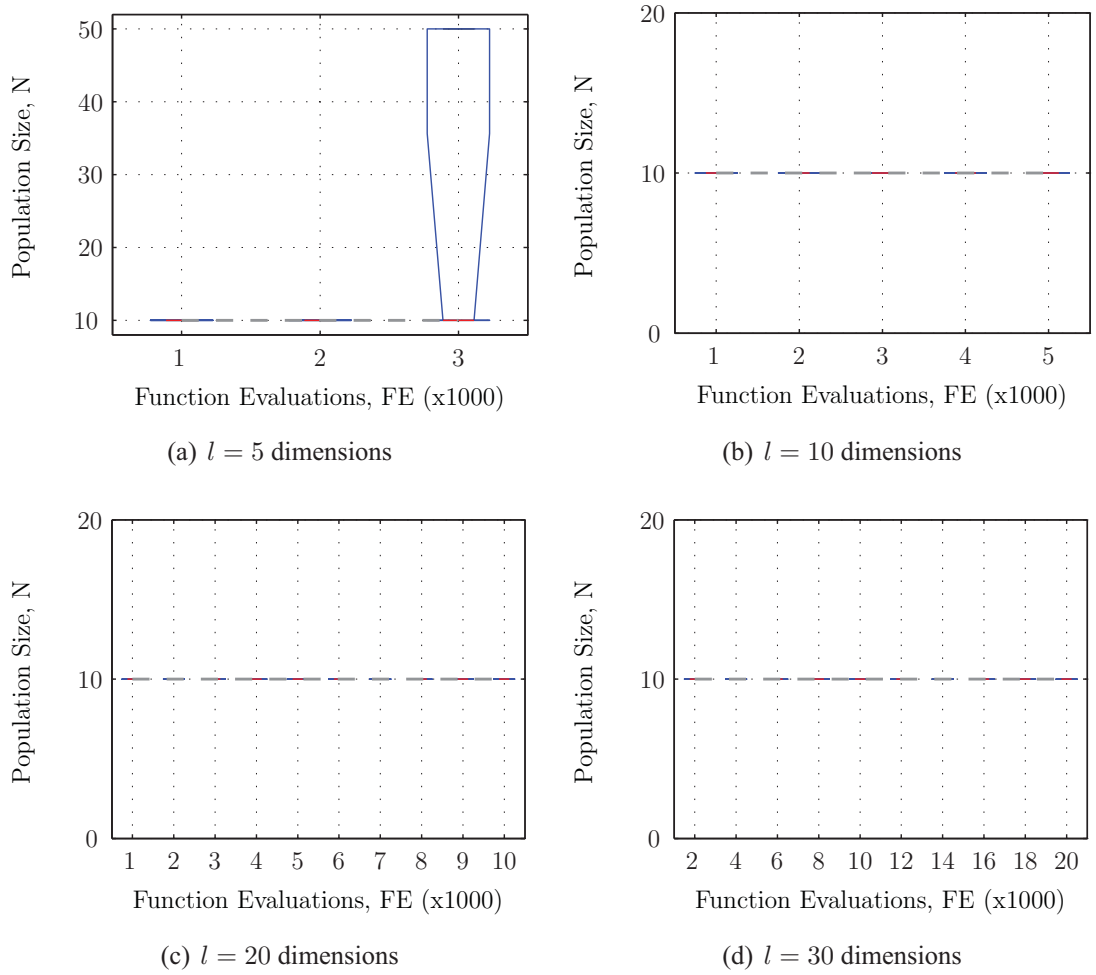


Figure 3.3 Optimal populations sizes for F1 for different function evaluations.

F2 The fitness function for F2 is similar to F1, the difference being the local optima produced by the sum of the cosine terms, seen in the plot of F2 in Figure 3.2(b). The results presented in Figure 3.4 indicate that the presence of these local optima did not affect the best population size to solve the problem, which was again always the smallest of the sizes tested. However, the presence of local optima did increase the number of function evaluations required to locate the optimal solution, requiring approximately 2.6 times the number of function evaluations taken to solve the same size problem with F1 (as seen in Table 3.4).

F5 Unlike F1 and F2, F5 does not have an equal contribution from each decision variable, as there is an increasing contribution from each decision variable to the fitness function value. This can be seen in Table 3.1, where the value for each decision variable is

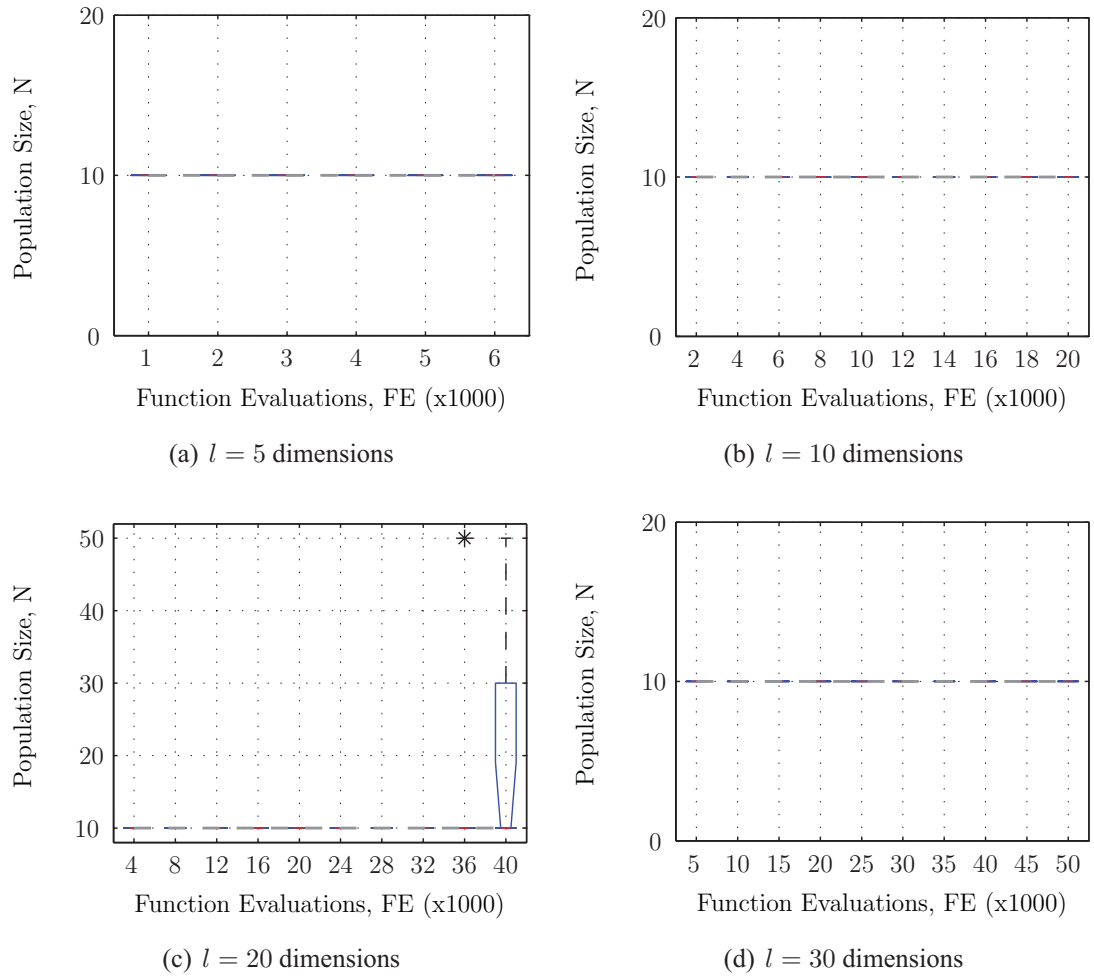


Figure 3.4 Optimal populations sizes for F2 for different function evaluations.

multiplied by its position in the solution string. However, the different contributions from the decision variables were not enough to alter their convergence rates, and again, the smallest population size produced the best results, as seen in Figure 3.5.

F7 The optimal population sizes found for the four problem sizes considered for F7 can be seen in Figure 3.6, and again the line fitted between the optimal population size for different number of fitness evaluations corresponds to the smallest population size considered. This result is for a different reason than for the other Maximal Generation Functions described. The fitness function for F7 suggests that there are strong interactions between adjacent pairs of decision variables, and therefore it might be expected that a larger population size would be more efficient at evaluating these combinations of values. However, this is not the case, because of the characteristics of the fitness function. A 2–

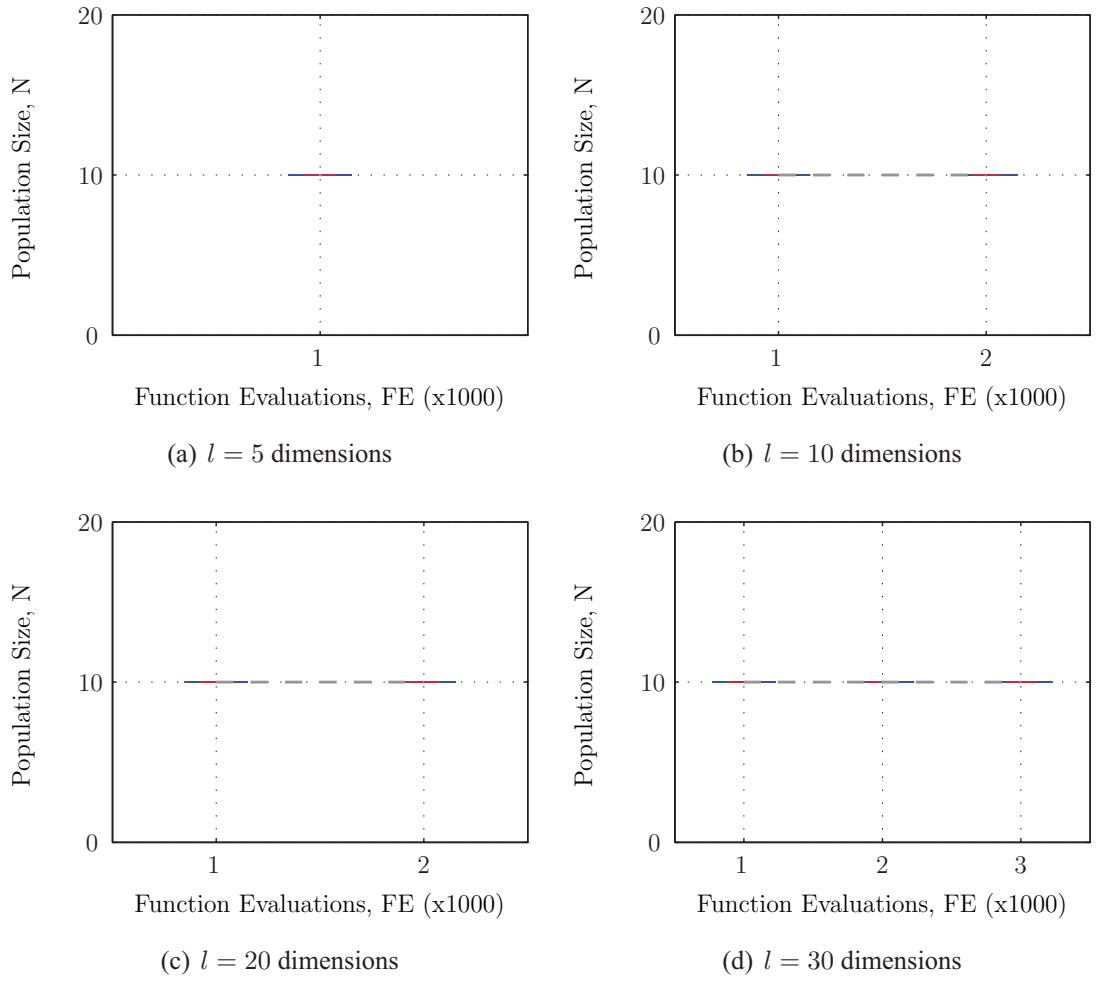
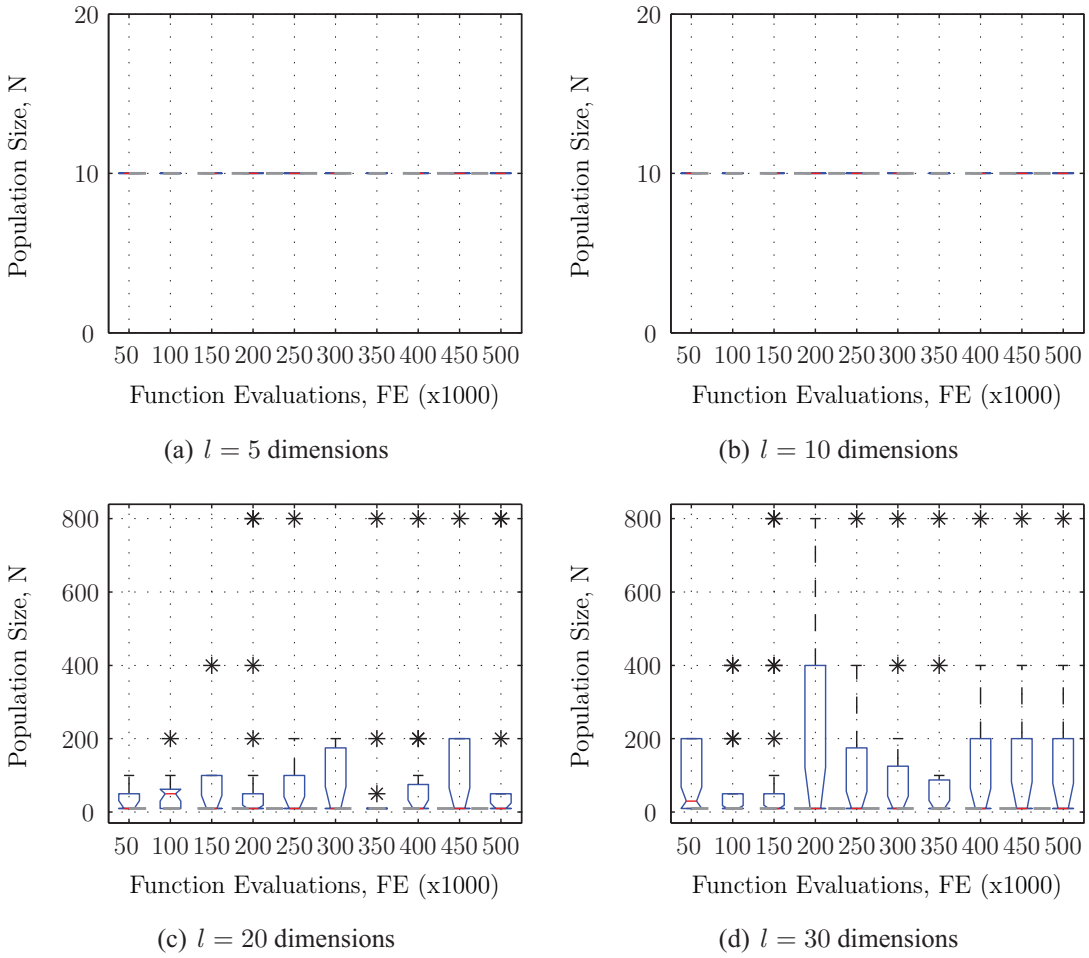


Figure 3.5 Optimal populations sizes for F5 for different function evaluations.

dimensional version of F7 is shown in Figure 3.7(b). Essentially, the search space is flat, with large changes in the fitness function value around the optimum solution of F7($\mathbf{x}$) $= 0$ at $\mathbf{x}= 0$. There is no structure in the fitness function to guide the GA, and therefore the algorithm is randomly searching the solution space. Hence, the parameters have very little impact on GA performance. This can be seen in Figure 3.6(c) and Figure 3.6(d), where all the population sizes considered in the parametric study find the best solution, as indicated by the bars and outliers on the box plots spanning the whole range of population sizes. However, there are more occurances of the smaller population sizes, and therefore $N = 10$ is computed as the optimal population size. For the smaller problem sizes of $l = 5$ and 10 from Figure 3.6(a) and Figure 3.6(b) it can be seen that only $N = 10$ located the best solutions, due to more updates of the population through more generations for the smaller population sizes, with elitism in place to continually evaluate and improve the



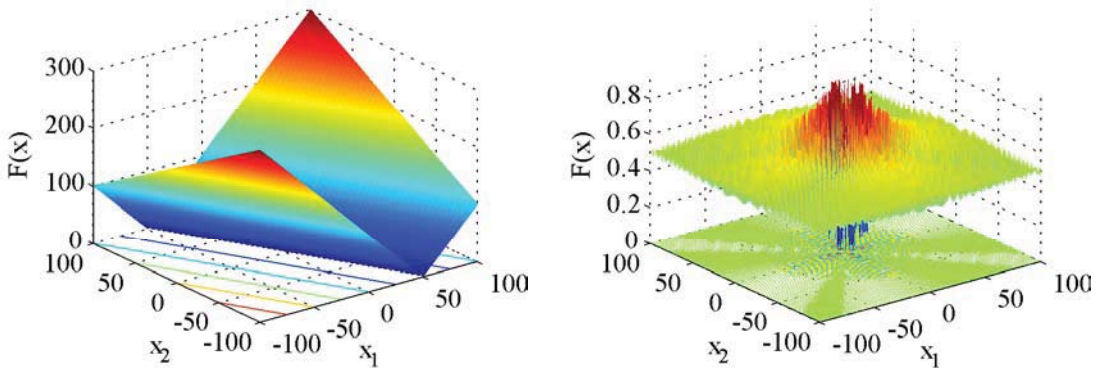
(a) $l = 5$ dimensions

(b) $l = 10$ dimensions

(c) $l = 20$ dimensions

(d) $l = 30$ dimensions

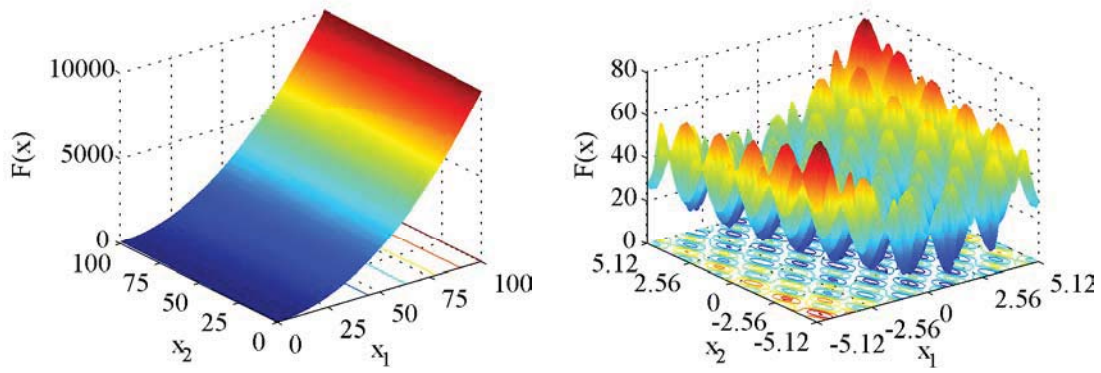
Figure 3.6 Optimal populations sizes for F7 for different function evaluations.



(a) 2–dimensional plot of F5

(b) 2–dimensional plot of F7

Figure 3.7 Functional form of F5 and F7.



(a) 2–dimensional plot of F6 (b) 2–dimensional plot of F3

Figure 3.8 Functional form of F6 and F3.

best solution found so far.

3.1.2.2 Optimal Generation Functions

As seen in Table 3.3, the functions that are best solved with an optimal number of generations are F3, F4, and F6. Similarly to the Maximal Generation Functions, this result can be explained by the characteristics of these problems.

F6 A 2–dimensional plot of F6 can be seen in Figure 3.8(a). Initially, the x_l^2 term in the fitness function dominates the fitness function value, and therefore this decision variable must converge close to the optimal value before any of the remaining decision variables can significantly contribute to the fitness function value. The fitness function for F6 suggests that it is a completely separable function, and therefore each decision variable can be optimized independently, so it might be expected that a smaller population size would perform the best, similar to that observed for the Maximal Generation Functions described previously. However, an increase in population size as the number of function evaluations increased was the most efficient method to solve F6, as seen in Figure 3.9. This result is due to the phenomenon of genetic drift. Larger population sizes will preserve more diversity for all decision variables, so once the l^{th} decision variable has converged enough for the remaining decision variables to contribute to the fitness function value, there is still some diversity in the population for the others to also converge to their optimal solution. The results for F6 make for an interesting comparison with those obtained for F5. The consequence of the inclusion of the x_l^2 term into the fitness function is that an optimal number of generations is required to solve the problem (F6). However, without

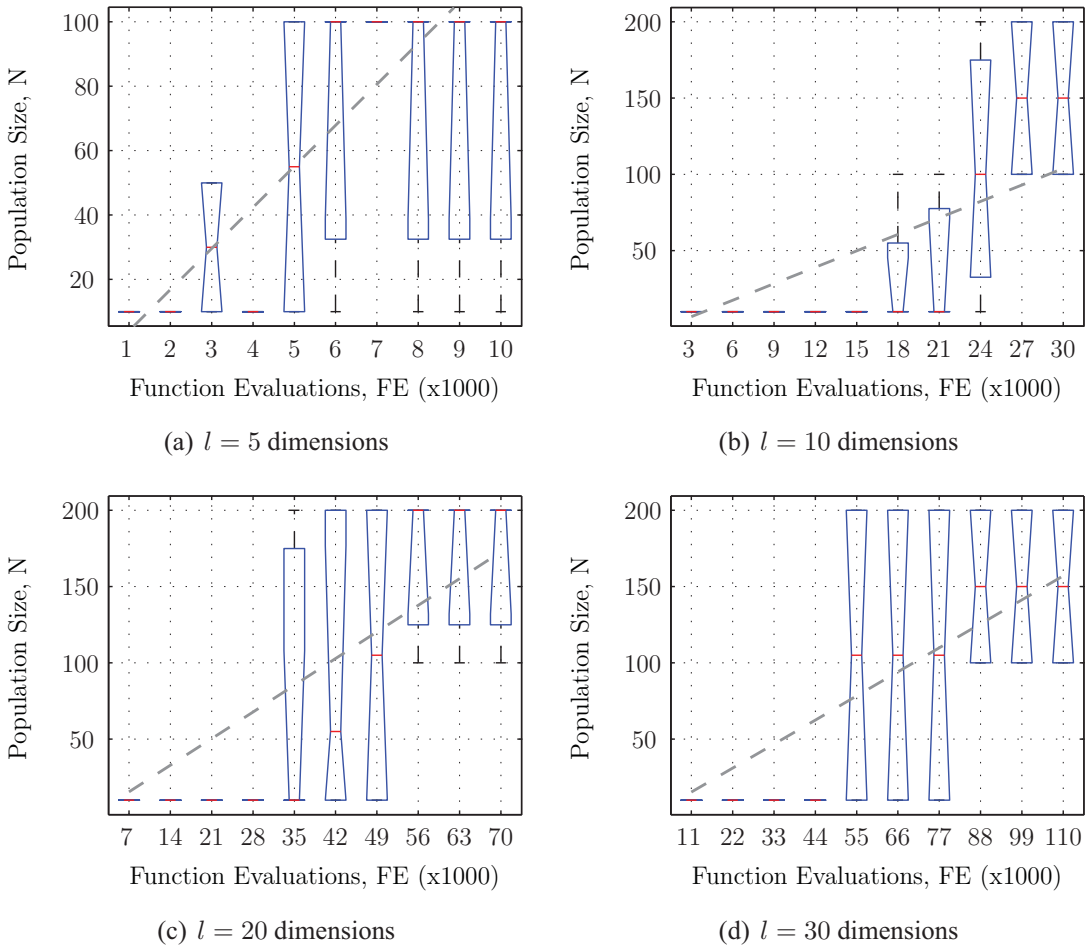
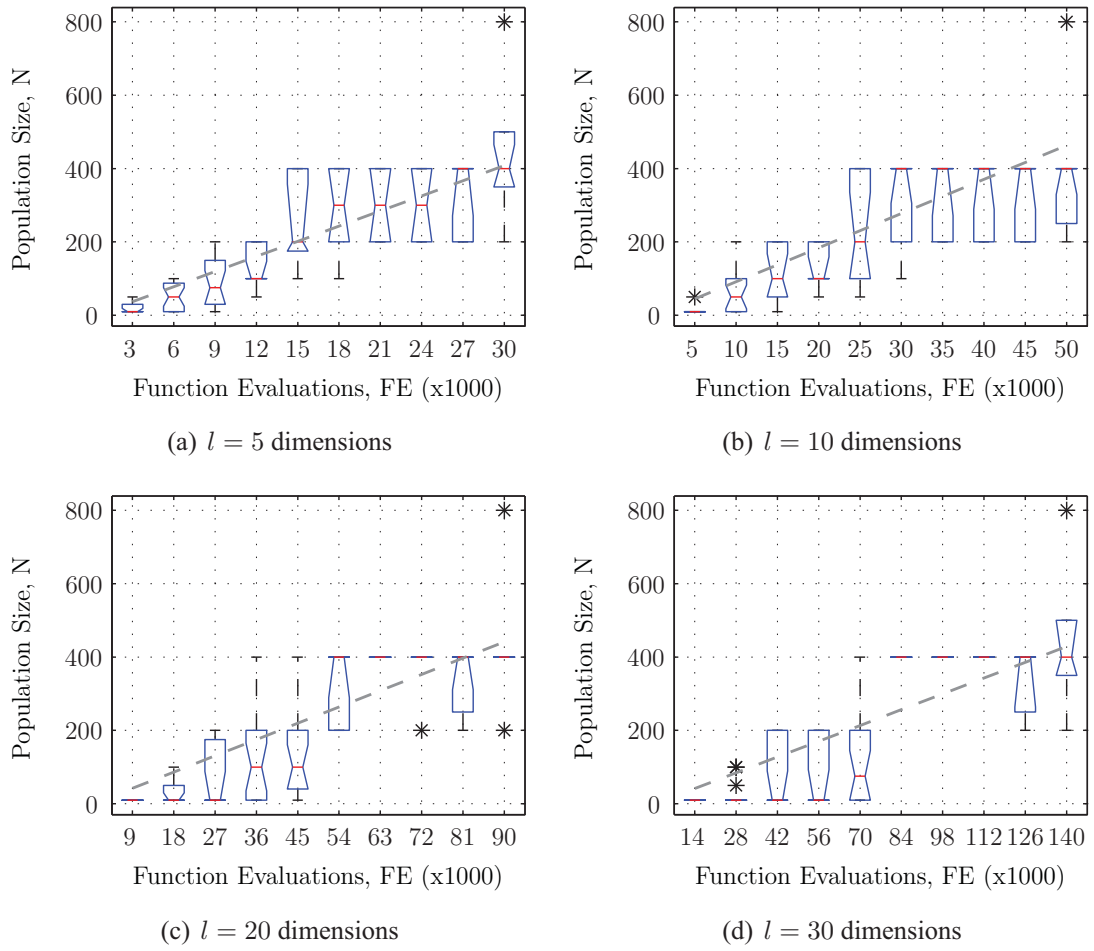


Figure 3.9 Optimal populations sizes for F6 for different function evaluations.

the x_l^2 term, and even with the inclusion of the modulus of the decision variable values, the smallest population size is the most suitable (F5).

F3 A very strong relationship between the optimal N and FE was observed for F3, as seen in Figure 3.10. F3 has an equal contribution from each decision variable to the fitness function value (although there is a slightly higher contribution from the l^{th} decision variable). A 2–dimensional plot of the function can be seen in Figure 3.8(b). Consequently, it might be expected that the $g_{\text{conv}} \propto \sqrt{l}$ relationship found by Thierens and Goldberg (1994) would be observed. This is the case for the two smaller problem sizes ($l = 5, 10$), where the $g_{\text{conv}} \propto \sqrt{l}$ relationship occurs with a proportionality constant of approximately 34, as seen in Table 3.3. However, for the problem sizes of F3 with $l \geq 10$, the relationship between g_{conv} and l changes to the domino convergence relationship pro-



(a) $l = 5$ dimensions

(b) $l = 10$ dimensions

(c) $l = 20$ dimensions

(d) $l = 30$ dimensions

Figure 3.10 Optimal populations sizes for F3 for different function evaluations.

posed by Thierens et al. (1998), $g_{\text{conv}} \propto l$. This relationship can be seen in Table 3.3, with a proportionality constant of approximately 11. For the larger problem sizes of F3, there are more decision variables, and due to the random nature of the GA, it is more likely that the decision variables will converge at different rates. When this is the case, the decision variables, or small combinations of decision variable values, will have different contributions to the fitness function value. Hence, the decision variables that have values that are most distant from the optimum, and therefore have the biggest contribution to the fitness function value, must be improved first before the contribution from other decision variables is significant again. Hence, for the larger problem sizes, it is more likely that there will be variations in the rate of convergence of the different decision variables, and therefore domino convergence occurs, and the $g_{\text{conv}} \propto l$ relationship is observed.

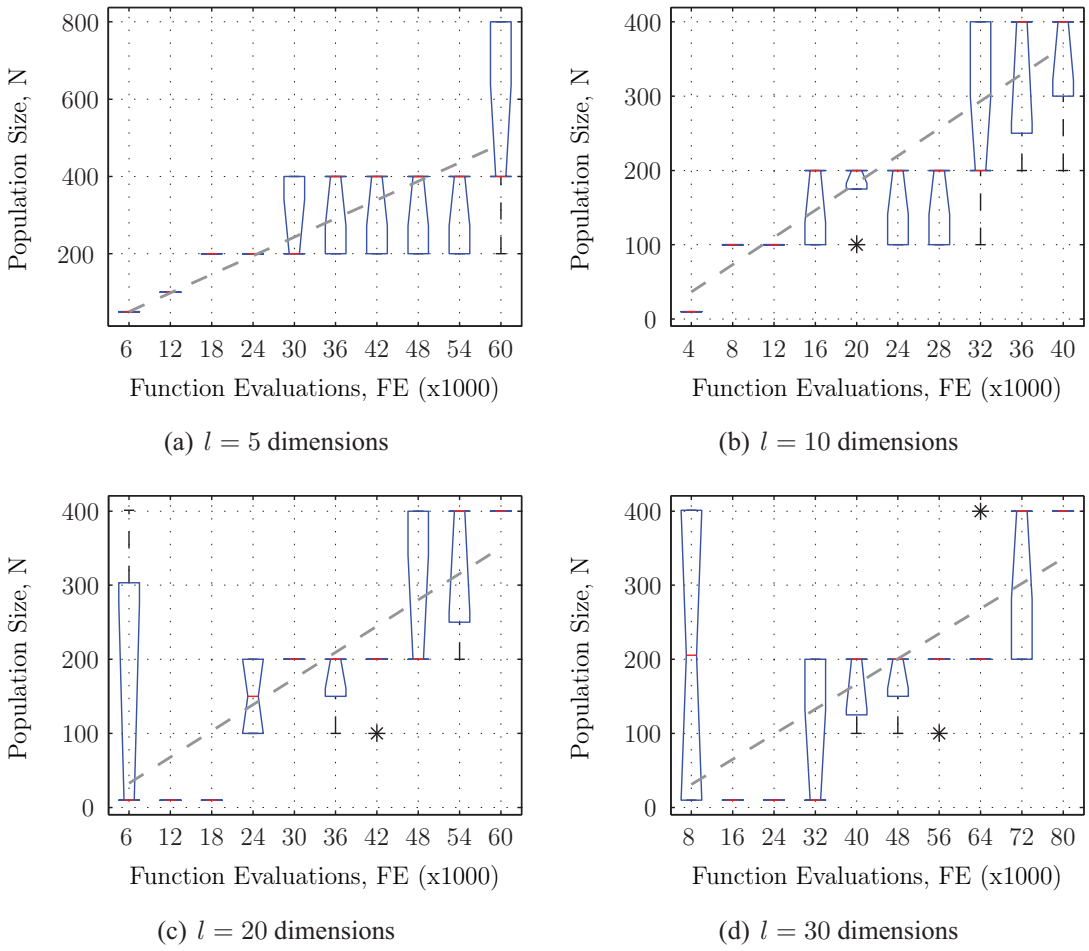
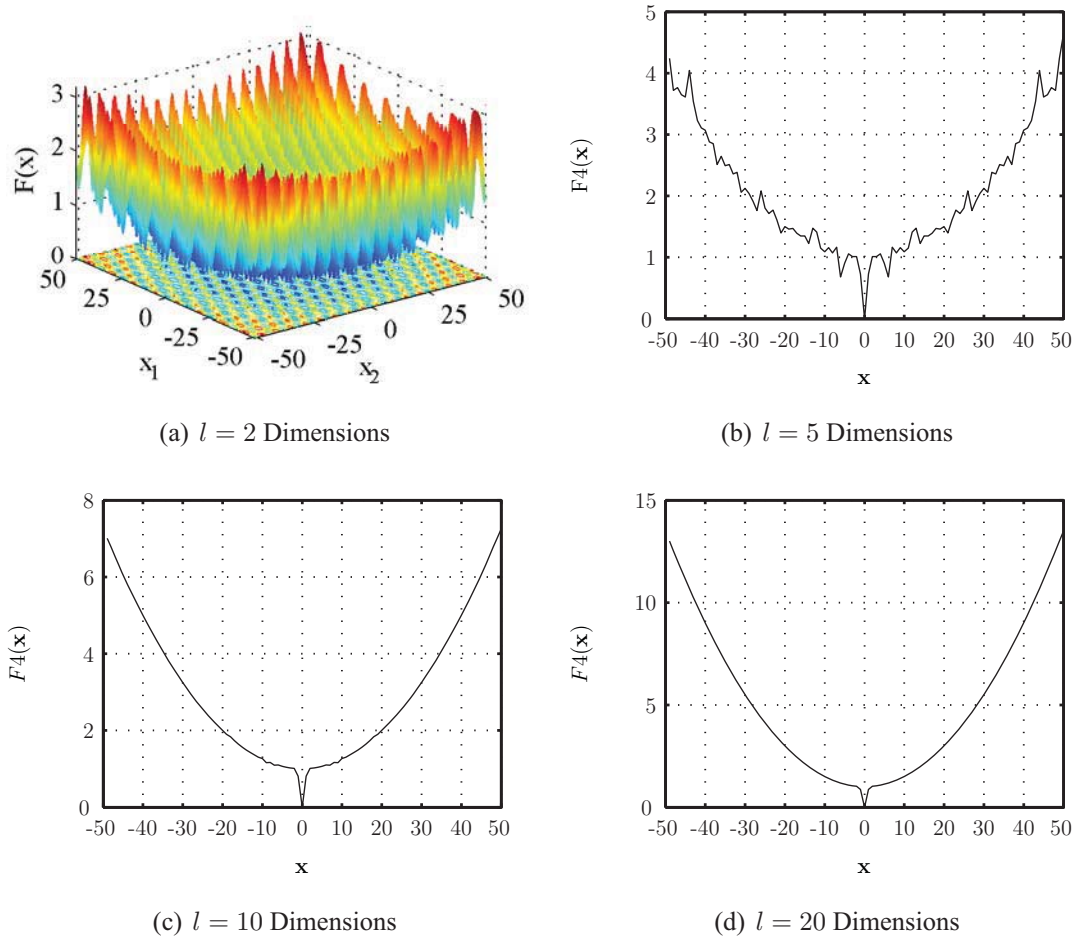


Figure 3.11 Optimal populations sizes for F4 for different function evaluations.

F4 F4 has a similar contribution from each decision variable to the fitness function, the only difference being that the $x_i/\sqrt{i}$ terms alter the frequency of the local optima produced by the cosine term in each dimension. From Figure 3.11 it can be seen that again a strong relationship between N and FE was observed for F4, producing a constant g_{conv} value for this function. On first inspection of Table 3.3, it appears as though neither of the two relationships between g_{conv} and l occurred, and, unexpectedly, that the number of function evaluations required to locate the optimum solution does not increase with the size of the problem. This result is due to changes in the form of the fitness function as the problem is scaled into higher dimensions. In Figure 3.12, F4 is plotted in $l = 2$ dimensions, as well as l-dimensional cross sections of the function taken along the diagonal of the hypercube of the search space for the function in $l = 5, 10$, and 20 dimensions (Whitley et al., 1995). From the plots it can be clearly seen that as the dimension



(a) $l = 2$ Dimensions

(b) $l = 5$ Dimensions

(c) $l = 10$ Dimensions

(d) $l = 20$ Dimensions

Figure 3.12 Plot of F4 in $l = 2, 5, 10$, and 20 dimensions. The higher dimensional plots are 1–dimensional cross-sections of the function taken along the diagonal of the hypercube.

of the problem increases, the size of local optima in the fitness function, produced by the product of the cosine term, decreases compared to the 'big bowl' structure of the fitness function produced by the x^2 term.

This change in the characteristics of the fitness function can be seen to influence the results for F4 in $l = 5$ and 20 dimensions, which are solved in a similar FE of 61 000 and 63 000, respectively. As the size of the problem increases, it would be expected that the FE required to solve the problem would also increase. Also, a decrease in the influence of the local optima on the fitness function value will produce a decrease in FE required to solve the problem. This was observed between F1 (Sphere) and F2 (Rastrigin), where F2 had a number of local optima introduced into the fitness function, and therefore required approximately 2.6 more FE than to solve F1. Thus, the result of a combination of increasing the problem size and decreasing the size of local optima is that for F4,

both cases are solved in a similar number of FE. For the case of F4 in $l = 5$ and 10 dimensions, the effect of reducing the size of the local optima outweighs the effect of the increase in problem size, and reduces the FE required to locate the optimal solution for the larger problem size. Therefore, as the problem size is increased, the characteristics of the fitness function change, and a different number of GA generations performs best. From Figure 3.12, it can be seen that the cosine term has a negligible impact on the fitness function for F4 in $l = 20$ dimensions, thus the change in fitness function as the problem size is increased is much less prominent. Therefore, for the case of F4 in $l = 20$ and 30 dimensions, a relationship of $g_{\text{conv}} \approx 7l$ is produced, as seen in Table 3.3.

3.1.2.3 Problem Size Effects

For the separable functions with similar contributions to the fitness function from each decision variable, F1, F2, and F5, the size of the problem does not affect the optimal population size, as seen in Figure 3.3, Figure 3.4 and Figure 3.5, respectively. For these functions the optimal population size was always the smallest population size that was considered. This result indicates that it is more efficient to have a smaller population size updated more frequently through more generations for the completely separable function, and the most efficient way to locate better solutions is through more generations for the small population size. Therefore, for these cases, and possibly for all separable functions with similar decision variable contribution to the fitness function, better results are obtained by increasing the number of generations, as opposed to increasing the population size.

For two of the Optimal Generation Functions considered, F3 and F6, the problem size had an effect on the optimal number of GA generations. As noted above, the optimal g_{conv} increases in proportion to $\sqrt{l}$ for the smaller problem sizes of F3, increasing to $g_{\text{conv}} \propto l$ for the larger problem sizes of F3, as well as for all problem sizes of F6. For F4, the optimal number of generations is very consistent for the first three problem sizes considered ($l = 5, 10, 20$), which on first inspection may suggest that there is an optimal number of generations to solve this function, irrespective of its size. However, this is due to the combination of the local optima in the fitness function becoming smaller, and therefore decreasing the optimal g_{conv} (as outlined in Section 3.1.2.2), and at the same time the increase in problem size increasing the optimal g_{conv}. This is reinforced by the results for F4 in $l = 20$ and 30 dimensions, as shown in Table 3.3 and Figure 3.12, where the optimal number of generations increases from $g_{\text{conv}} = 121$ to 236. From Figure 3.12, it can be seen that the local optima for F4 are already very small for $l = 20$. Therefore,

the increase to $l = 30$ increases the optimal number of generations required to solve the problem, without the corresponding decrease in the roughness of the fitness function that occurred for the smaller problem sizes. Hence, the optimal number of generations to solve the problem increases with problem size, as would be expected for this function, but for this case the effect is being offset by changes in the fitness function as it is scaled into higher dimensions.

The results indicate that the effect of problem size on the most suitable number of GA generations is dependent on the problem characteristics. For the functions considered, if the problem was a Maximal Generation Function then the best results were always obtained with the smallest population size and maximizing the number of generations, irrespective of problem size. However, from Table 3.3 it can be seen that if the fitness function is an Optimal Generation Function, then an increase in the problem size, l, results in an increase in the optimal number of generations, g_{conv}. The relationship between g_{conv} and l ranged from $O(\sqrt{l})$ for the smallest problems sizes considered, up to $O(l)$ for the larger problem sizes.

3.1.2.4 Epistasis Effects

The effect of the epistatic interactions on the optimal g_{conv} to solve the problem can be directly investigated from the results for F2 and F3, as the only difference between the two functions is the introduction of interactions between the decision variables in F3 by the $(x_i + x_{i+1})/2$ terms. Even though both fitness functions have a similar form, the introduction of the interactions produces vastly different optimal GA parameter results. The difference in results can be explained by the efficient use of function evaluations. For F3, there are interactions between decision variables, and combinations of decision variable values must be stored and evaluated to solve the problem. For a GA, this is done using a large N, and better solutions are found using a larger population size, provided there is time for the GA to process the information in the population and converge to a solution. Hence, as the number of function evaluations increases, so does the optimal population size. However, these larger population sizes do not find the best results for fewer function evaluations, as there is insufficient time to process all the combinations of decision variable values in the population and thus to converge on a near optimal solution.

As outlined in Section 3.1.2.3, for F2, which does not have any interactions between decision variables, it is more efficient to have a smaller N with more generational updates, as opposed to an optimal g_{conv}, which was the case for F3. Functions F1 and F5 also fall into this class, where each decision variable has a similar contribution to the fit-

ness function and can be optimised independently of the other decision variable values. F6 is also a separable function without any epistatic effects between decision variables, however, as outlined in Section 3.1.2.2, the l^{th} decision variable has a much higher contribution to the fitness function value and therefore a larger N is required to ensure that there is enough diversity in the population for the remaining decision variables once the l^{th} decision variable has converged close to the optimal value.

As was case for F3, epistatic effects between decision variables are produced in the fitness function for F4, this time by the product of the cosine terms. Therefore, each decision variable cannot be optimised separately, and a large N must be used. A similar relationship between g_{conv} and l to that found for F3 is not observed for F4, due to the changes in the fitness function characteristics as the problem size increases. F7 also has epistatic interactions between pairs of adjacent decision variables, so it might be expected that there would also be a strong relationship between the optimal g_{conv} and l. However, for F7, there is no useful information in the fitness function to guide the GA toward better solutions, and the GA parameters had very little effect on the solution found. While there was little difference between the GA results obtained using different GA parameter values, the best results were achieved using a small N, and improving the best solution found through as many generational updates as possible.

3.1.3 Discussion of Parametric Study Results

The results presented in this section suggest that there were four classes of fitness function characteristics, each of which required different GA behaviour in order to obtain the best solutions most efficiently. The different calibration classes identified are depicted in the flow chart shown in Figure 3.13. The first class of problem characteristics is where there is no structure in the fitness function to lead the GA to better solutions. F7 belongs to this class, and for this class of problems the results suggest that the GA parameters have little effect on the solution found. However, the best results were obtained with a small population size with as many generational updates of the population as possible. The second class of problem characteristics is where there is structure in the fitness function to lead the GA toward the optimal solution, and there are epistatic interactions between decision variables. Functions F3 and F4 belong to this class, and the results suggest that there is an optimal g_{conv} to most efficiently solve these problems. There is also an optimal g_{conv} to solve problems in the third class, where there is structure in the fitness function and significantly different contributions to the fitness function value from at least one decision variable. F6 belong to this class, due to the x^2 term dominating the fitness function value.

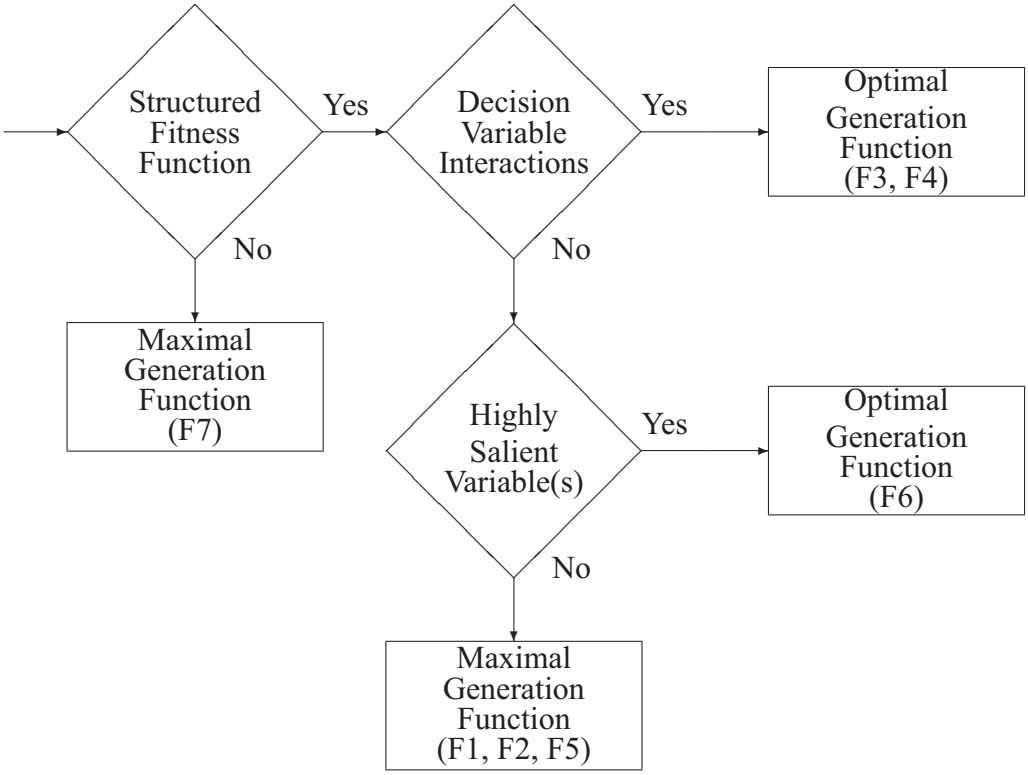


Figure 3.13 Flow chart depicting the fitness function calibration classes.

The final class of problem characteristics seen in Figure 3.13 is where there is structure in the fitness function, there are no epistatic interactions between decision variables, and each decision variable has a similar contribution to the fitness function value. Functions F1, F2, and F5 belong to this class, and in this case, the problem is most efficiently solved with a small population size, updated through as many generations as possible.

Figure 3.13 does not consider the possibility of a function belonging to both classes 2 and 3, where a function has both epistatic interactions between decision variables and at least one decision variable with a significantly higher contribution to the fitness function value than the others. For this case, it is expected that the function would be an Optimal Generation Function, however, a function with these characteristics was not included in the set of test functions considered.

The calibration classes that have been identified are useful in the case where the equation of the fitness function is known, and therefore some information can be inferred about the problem characteristics. However, often this is not the case, for example, if the fitness function is constructed from the results of a simulation model, which is the case for WDS optimisation. This highlights the potential usefulness of fitness function statistics.

For example, a gene significance measure (Seo et al., 2003) could detect a difference in the contribution from each decision variable, an epistasis measure (Davidor, 1991) could detect interactions between decision variables, and a correlation measure (Weinberger, 1990) could detect the size of local optima relative to the global structure of the fitness function.

A correlation measure such as this would provide useful information about the changes in F4 as it is scaled into higher dimensions. In 2 dimensions (Figure 3.12(a)), a relatively low degree of correlation would be expected, as the local optima produced by the cosine term are of a similar magnitude as the structure in the fitness function, leading to the global optimum produced by the x^2 term. However, for F4 in 20 dimensions (Figure 3.12(d)), a much higher correlation measure would be expected, as the size of the local optima are negligible compared to the strong global structure leading to the optimal solution.

Statistics to determine these characteristics of fitness functions are proposed in Chapter 4. However, before these statistics are introduced the remainder of this chapter considers controlled changes to the characteristics of the Optimal Generations Functions, to investigates if these changes have an influence on the observed value of g_{conv}.

3.2 The Effect of Function Characteristics on the Number of Generations

The hypothesis to be empirically tested in this section is that if a function is an Optimal Generation Function, a change in the characteristics of the function will produce a change in the value observed for g_{conv}. The methodology used to test this hypothesis is outlined in the following section.

3.2.1 Methodology

A similar methodology to that used in the previous section is also adopted here. Again, an empirical approach has been used, in this case to test if the value of g_{conv} observed in Section 3.1 can be related to the characteristics of the fitness function. In order to achieve this, the three Optimal Generation Functions identified in Section 3.1 have been tested with different characteristics, to investigate the impact of the change in characteristics on the value of g_{conv}.

The same conditions have been adopted for this analysis as Section 3.1: every combination of the GA parameter values outlined in Table 3.2 have been used; each combination has been run for 30 different sequences of random numbers; the GA has been applied to

Table 3.5 The test functions with controllable characteristics used in the parametric study.

	Function to be Minimised
F3A	$\sum_{i=1}^{l}\sum_{j=i}^{l}\begin{cases}\left(\frac{x_i+x_j}{2}\right)^2 - A\cos\left(f\pi\frac{x_i+x_j}{2}\right)+A & \text{if } i,j \text{ interact,}\\ 0 & \text{otherwise.}\end{cases}$
F3B	$\sum_{i=1}^{l-1}\left(\left(\frac{x_i+x_{i+1}}{2}\right)^2 - A\cos\left(f\pi\frac{x_i+x_{i+1}}{2}\right)\right) + x_l^2 - A\cos\left(f\pi x_l\right) + Al$
F3C	$\sum_{i=1}^{l-1}\left(\left(\frac{x_i+x_{i+1}}{2}\right)^2 - 10\cos\left(2\pi\frac{x_i+x_{i+1}}{2}\right)\right)^{p_i} + x_l^2 - 10\cos\left(2\pi x_l\right) + 10l$
F4A	$\frac{1}{4000}\sum_{i=1}^{l} i^p x_i^2 - A\left(\prod_{i=1}^{l}\cos\left(\frac{x_i}{\sqrt{i}}\right) - 1\right)$
F4B	$\frac{l^p x_l^2}{4000} + \frac{1}{4000}\sum_{i=1}^{l-1} i^p\left(\frac{x_i+x_j}{2}\right)^2 - A\left(\cos\left(\frac{x_l}{\sqrt{l}}\right)\prod_{i=1}^{l-1}\cos\left(\frac{\left(\frac{x_i+x_j}{2}\right)}{\sqrt{i}}\right) - 1\right)$
F6A	$\sum_{i=1}^{l-1} i^A x_i + x_l^p$
F6B	$\sum_{i=1}^{l-2} i^A\left(\frac{x_i+x_{i+1}}{2}\right) + \left(\frac{x_{l-1}+x_l}{2}\right)^p$

each variation of each test function for problem sizes of $l = 5$, 10, 20, and 30; and the best performing GA parameters were stored every 1 000 function evaluations as the algorithm solved each function.

3.2.1.1 Test Functions

The Optimal Generation Functions identified in Section 3.1 have been used to test the effect of different fitness function characteristics on the observed value of g_{conv}. These functions were F3, F4, and F6, and the changes to their characteristics that have been adopted are given in Table 3.5.

For F3, different degrees of interaction, roughness in terms of the size and frequency of the local optima, and the scaling of the variables have all been considered. F3A allowed for changes in the degree of epistasis of the problem to be altered. All the functions considered have pair-wise interactions, however if one decision variable is included in more than one interaction, larger building blocks are produced. The different cases that have been considered are presented in Table 3.6. Not all combinations of the two parameters are possible, for example one interaction, $m_{\text{BB}} = 1$, between adjacent variables, $\delta_{\text{BB}} = 1$,

Table 3.6 The function values used for F3A.

	δ_{BB}		
m_{BB}	1	$l/2$	$l-1$
1	x	x	x
$l/2$	x	x	
$l-1$	x		

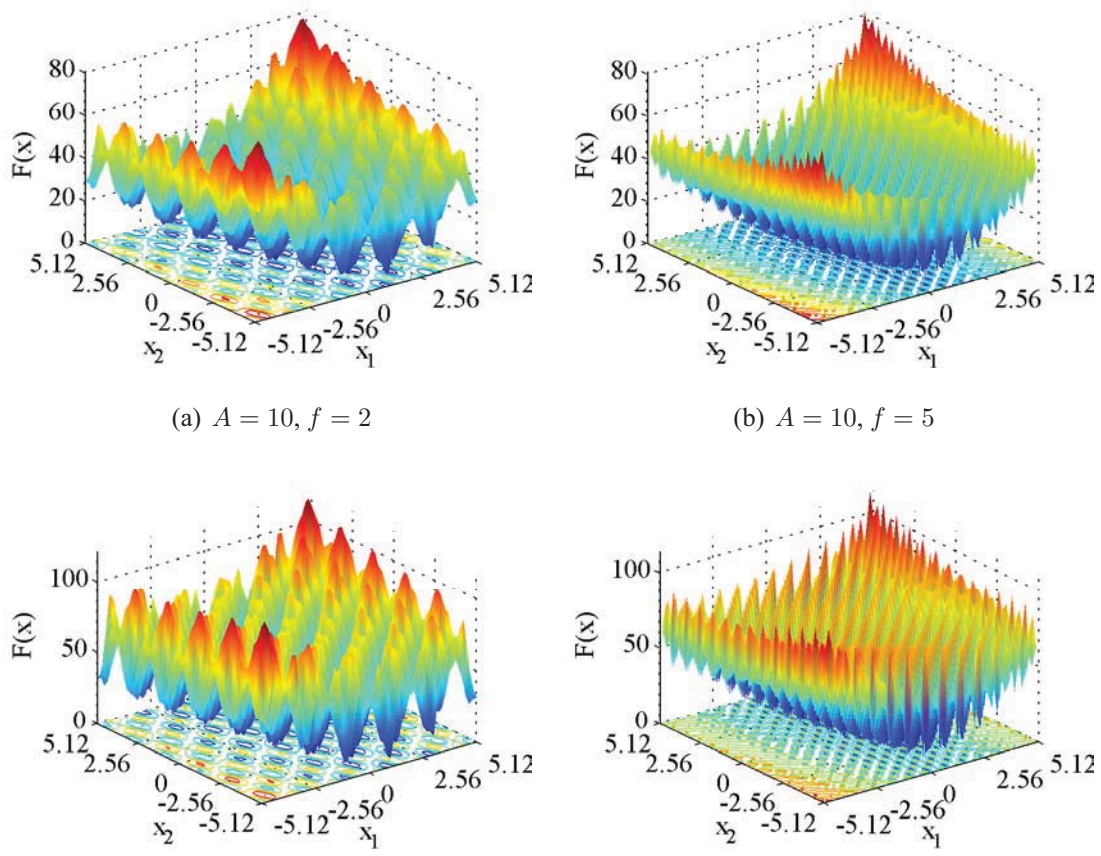
Table 3.7 The function values used for F3B.

Parameter	Values
A	10, 20
f	2, 5

between variables half the solution string apart, $\delta_{\text{BB}} = l/2$, and the complete solutions string apart, $\delta_{\text{BB}} = l - 1$ have been tested, however, it is not possible to have all of the decision variables interacting with the distance between all the interactions the whole solution string apart, which would produce values of $m_{\text{BB}} = l$ and $\delta_{\text{BB}} = l - 1$. The interaction cases that have been tested are marked with an 'x' in Table 3.6.

Variations in the multimodality and roughness of the Rastrigin Function have been considered in function F3B. The values considered for the parametric study can be seen in Table 3.7, where all combinations of A and f have been tested. The effect of changes in these values on the characteristics of the function can be seen in Figure 3.14. An increase in the value of A, and therefore the roughness of the function, can be seen in the difference between the two–dimensional plots of F3B in Figure 3.14(a) and Figure 3.14(c), as well as Figure 3.14(b) and Figure 3.14(d). Similarly, the effect of an increase in the value of f, and subsequently an increase in the multimodality of the function, can be seen between Figure 3.14(a) and Figure 3.14(b), as well as Figure 3.14(c) and Figure 3.14(d).

Variations in the salience of the decision variables for the Rastrigin Function, or the contribution of each variable on the fitness function values, have been considered in the different cases considered for F3C. The different cases tested in the parametric study for F3C are presented in Table 3.8. Case 1 is the original version of F3, with each variable having the same contribution to the fitness function value. Case 2 of function F3C has



(a) $A = 10$, $f = 2$

(b) $A = 10$, $f = 5$

(c) $A = 20$, $f = 2$

(d) $A = 20$, $f = 5$

Figure 3.14 Plots of the effect of parameters A and f on the characteristics of F3B

the sub-function of the Rastrigin Function in terms of $(x_1 + x_2)/2$ squared, with $p_1 = 2$, and a linear contribution from the remaining pairs of variables, which all have the value $p_i = 1$. Case 3 has the contribution of the first two terms squared, as opposed to only the first term for Case 2. To provide a comparison with Cases 2 and 3, Case 4 has the contribution of all variables squared.

The effect of this change in the salience on the fitness function characteristics can be seen in Figure 3.15, where x_2 has a much larger effect on the values of $F(\mathbf{x})$ than x_1, and hence values for x_2 close to the optimal values must be found before x_1 contributes enough to the fitness function values to also be optimised. Note that Figure 3.15 does not include the interactions between the variables for F3C seen in Table 3.5, as only two dimensions of the function can be plotted, and is used for demonstration only.

Two parameters have been introduced into F4 to allow the effect of changes to the degree of roughness and the degree of salience to be tested. These parameters can be

Table 3.8 The function values used for F3C.

Case	Parameter Values
1	$p_i = 1 \; \forall i \in 1 \ldots l$
2	$p_1 = 2, p_i = 1 \; \forall i \in 2 \ldots l$
3	$p_1 = p_2 = 2, p_i = 1 \; \forall i \in 3 \ldots l$
4	$p_i = 2 \; \forall i \in 1 \ldots l$

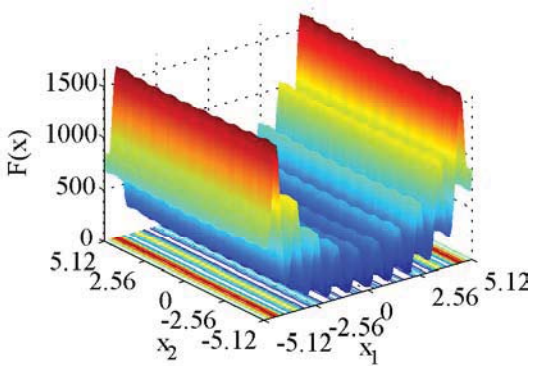


Figure 3.15 The effect of increasing the salience of x_2 relative to x_1 for F3C

seen in Table 3.5, where the A parameter can be used to change the roughness of the function, and the p parameter can be used to alter the salience of the variables. The values used for this function in the parametric study can be seen in Table 3.9. The effect of the change in these variables on the fitness function characteristics can be seen in Figure 3.16, where an increase in A, and therefore increases in the roughness of the function, can be seen between Figure 3.16(a) and Figure 3.16(b), Figure 3.16(b) and Figure 3.16(d), and Figure 3.16(e) and Figure 3.16(f). Changes in p are also evident in the changes in the fitness function characteristics, where both x_1 and x_2 have the same contribution to the fitness function in Figure 3.16(a) and Figure 3.16(b), with $p = 0$. In contrast, it is clear from Figure 3.16(e) and Figure 3.16(f) with $p = 2$ that x_2 has a much larger effect on the fitness function value than x_1.

The effect of interactions on the values of g_{conv} can also be investigated by the difference in values obtained for F4A and F4B. From Table 3.5, it can be seen that additional pair-wise interactions have been introduced into the function between adjacent decision variables for F4B compared to F4A. The same range of values for parameters A and p are

Table 3.9 The function values used for F4A and F4B.

Parameter	Values
A	1, l
p	0,1,2

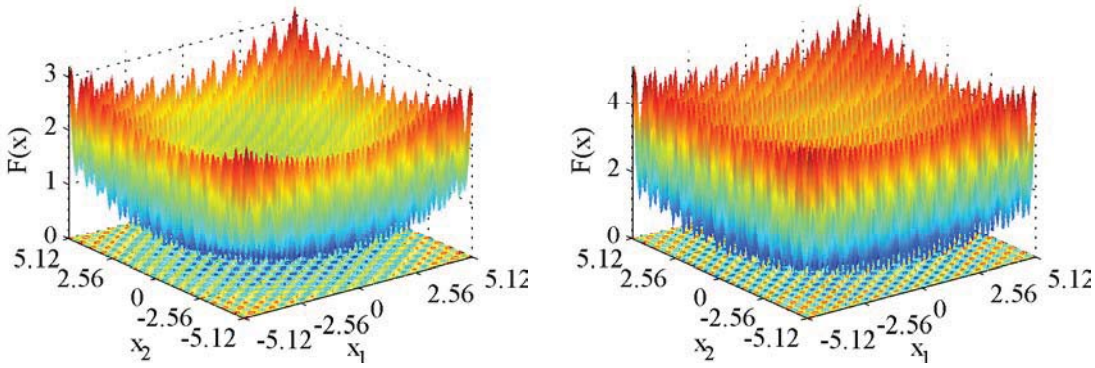
Table 3.10 The function values used for F6A and F6B.

Parameter	Values
A	2, 3
p	2, 3

also tested for F4B as those presented in Figure 3.16 for F4A.

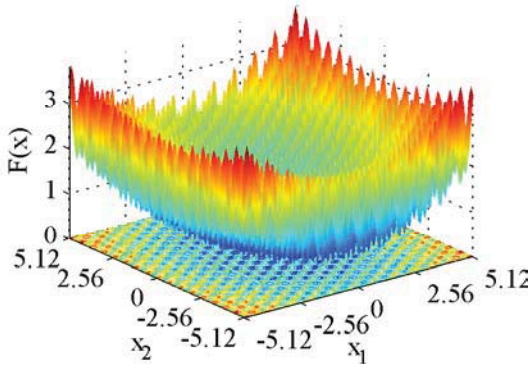
For F6, changes to the salience of the variables and degree of interaction have been considered. From Table 3.5 it can be seen that the parameter A can be used to change the size of the incremental dominance of each adjacent variable. Parameter p can be used to control the salience of the last decision variable in the solution string, x_l, by altering the magnitude of the order of its relationship with the fitness function value. The effect of parameter p on the characteristics of the fitness function can be seen in Figure 3.17, where in Figure 3.17(a) $p = 2$ and the linear increase in x_1 can be seen against the quadratic increase in x_2. However, in Figure 3.17(a) with $p = 3$, the effect of x_1 on the fitness function can hardly be noticed compared to that of x_2. It is not until the value for x_2 has converged close to the optimal value that x_1 will have a significant contribution to the fitness function value and be able to be optimised. As was the case for F4 above, the effect of interactions on the values of g_{conv} can also be investigated by the difference in values obtained for F6A and F6B, where pair-wise interactions have been introduced into the function between adjacent decision variables for F6B compared to F6A.

Problems sizes of $l = 5$, 10, 20, and 30 have been considered for every variation of each function. Every combination of each of the variables in Table 3.7, Table 3.9, and Table 3.10 has been tested. Therefore, a total of 136 functions have been tested in this section. The same GA parameters and convergence criteria used in Section 3.1.1.2 have been used, resulting in a total of 1 762 560 GA runs to produce the results presented in Section 3.2.2.

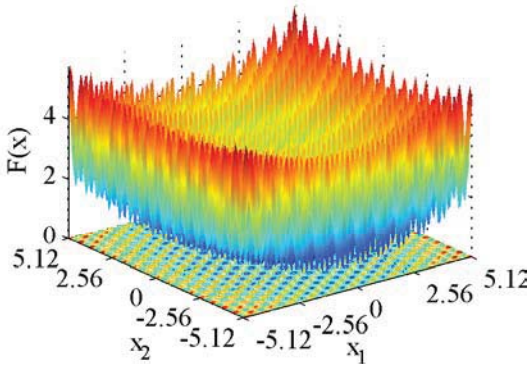


(a) $A = 1, p = 0$

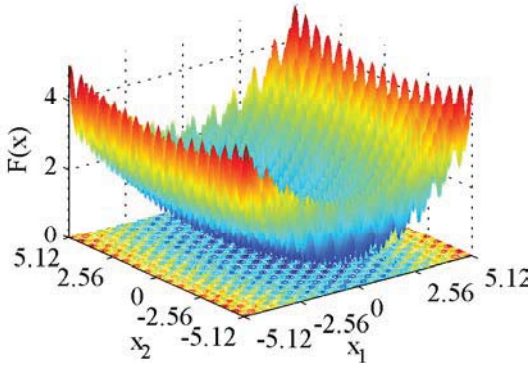
(b) $A = l, p = 0$



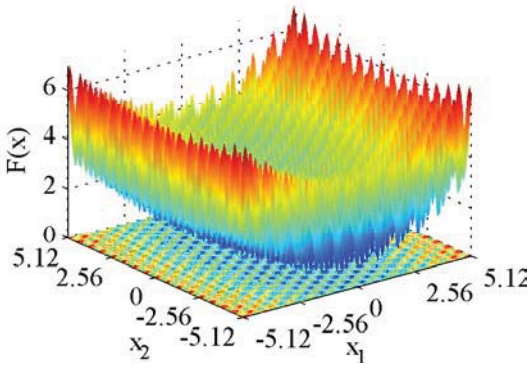
(c) $A = 1, p = 1$



(d) $A = l, p = 1$

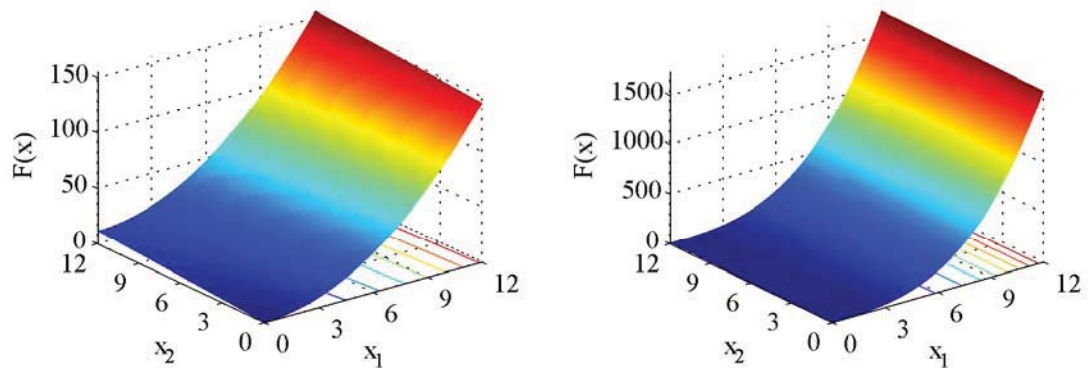


(e) $A = 1, p = 2$



(f) $A = l, p = 2$

Figure 3.16 Plots of the effect of parameters A and p on the characteristics of F4A



(a) $A = 1, p = 2$ (b) $A = 1, p = 3$

Figure 3.17 Plots of the effect of parameter p on the characteristics of F6.

3.2.2 Function Characteristics Results

In this section, the results from changing the characteristics of the Optimal Generation Functions on the value of g_{conv} are presented. This is followed by an explanation of the results and how they relate to the changes in the function characteristics, before a summary of the investigation into the relationship between function characteristics and g_{conv} is presented.

3.2.2.1 F3

F3A Function F3A considered variations in the degree of interaction between the decision variables for the Rastrigin Function. The values for g_{conv} obtained from the parametric study for each function considered are presented in Table 3.11. From Table 3.11 it can be seen that an increase in the number of interactions between pairs of variables, m_{BB}, produced a slight increase in g_{conv} for all cases considered. The only exception occurs for F3A in 30 dimensions with $\delta_{\text{BB}} = 1$ and $m_{\text{BB}} = 1$, with $g_{\text{conv}} = 385$. An increase in δ_{BB}, or an increase in the distance between the interacting decision variables, also produced an increase in g_{conv}. Again, the only exception to this observation occurred for F3A with $l = 30$, $\delta_{\text{BB}} = 1$ and $m_{\text{BB}} = 1$.

F3B Variations in the multimodality and the roughness of the Rastrigin Function have been investigated in function F3B. In this case, an increase in A produces larger local optima in the function, and therefore a rougher function. An increase in f increases the frequency of the local optima, producing an increase in the multimodality of the function. The values for g_{conv} produced from the parametric study for the variations of F3B consid-

Table 3.11 g_{conv} values observed for F3A.

		m_{BB}				m_{BB}		
δ_{BB}	l	1	2	$l-1$	l	1	2	$l-1$
1		57	63	75		191	212	219
2	5	65	78	-	20	201	237	-
l		83	-	-		229	-	-
1		63	108	107		385	260	325
2	10	60	115	-	30	268	284	-
l		68	-	-		269	-	-

Table 3.12 g_{conv} values observed for F3B.

		A			A	
f	l	10	20	l	10	20
2	5	68	101	20	191	228
5		64	88		145	225
2	10	113	210	30	314	435
5		113	225		225	338

ered are presented in Table 3.12. The results indicate that an increase in the roughness of the function, or an increase from $A = 10$ to $A = 20$, results in a much larger value of g_{conv}. From Table 3.12, it can be seen that an increase in the multimodality of F3B, produced by an increase in f, generally resulted in a slight decrease in the values of g_{conv} to most efficiently solve the problem. This result was consistent over all the cases considered, apart from a slight increase in the value of g_{conv} for F3B with $l = 10$ and $A = 20$.

F3C Changes to the salience of the variables for the Rastrigin function have been considered in the function F3C. For Cases 1 and 4, all variables have an equal contribution to the fitness function. It can be seen from Table 3.13 that these two functions have very similar values of g_{conv} for all problem sizes considered.

Table 3.13 g_{conv} values observed for F3C.

	Case			
l	1	2	3	4
5	80	112	144	63
10	139	161	196	135
20	233	245	298	225
30	346	267	349	336

For the two cases with varying salience between variables, Cases 2 and 3, from Table 3.13 it can be seen that the more dominant the variables that are present in the fitness function, the higher the values of g_{conv}. In general, the value of g_{conv} for Case 2 is higher than that found for Cases 1 and 4 for all dimensions considered, with g_{conv} for Case 3 higher again.

3.2.2.2 F4

For both F4A and F4B an increase in the difference in each decision variable's effect on the fitness function values is produced by an increase in the value of p, as seen in the functional form of F4A in Figure 3.16. From Table 3.14 it can be seen that for most of the functions considered, an increase in p produced an increase in g_{conv}.

The results for F3B suggest that an increase in the roughness of a function produced an increase in g_{conv}. This result was not observed as convincingly for changes in the A parameter for F4A and F4B. The effect of A on the characteristics of F4A was relatively obvious in the two dimensional plots shown in Figure 3.16, and it was expected that as l increased with the problem size, the roughness produced by $A = l$ would also increase. However this was not the case, as the roughness produced by the product of the cosine terms decreased as l increased, similarly to the case shown in Figure 3.12. Consequently, the result was that a change in A had little effect on the characteristics of F4A and F4B in higher dimensions, and therefore there was no definitive effect on the values of g_{conv} for changes in A, as seen in Table 3.14.

The effect of the introduction of extra epistatic interactions between the decision variables in the fitness function can be seen by comparing the results for F4A and F4B with the same values for A and p in Table 3.14. The results indicate that in most cases, an

Table 3.14 g_{conv} values observed for F4A and F4B.

		F4A			F4B		
		p			p		
A	l	0	1	2	0	1	2
1	5	121	62	83	60	70	77
l		59	78	89	56	62	87
1	10	109	142	162	79	87	110
l		98	166	145	101	81	124
1	20	121	128	159	121	122	206
l		138	125	155	113	119	201
1	30	236	283	300	168	241	222
l		281	297	313	153	237	201

increase in the number of interactions produced a slight decrease in the value of g_{conv}. The result was more convincing that the effect of A outlined above, as only 4 of the 24 comparable functions tested produced a larger g_{conv} for F4B compared to F4A.

3.2.2.3 F6

The salience of the decision variables for F6A and F6B was changed by both the A and p parameters. An increase in A produced an increase in the dominance of each decision variable over the variables preceding it in the solution string, and an increase in p increased the order of the dominance of the final variable in the solution string. The resulting values of g_{conv} for changes to A and p for F6A and F6B can be seen in Table 3.15. The results indicate that, in general, an increase in the salience of the variables, through either an increase in A or p, produces an increase in g_{conv}.

The effect of introducing interactions into F6 can be seen by comparing the results for F6A without interactions, and F6B, which has had pair-wise interactions introduced into fitness function. From Table 3.15, it can be seen that there is very little difference between the value of g_{conv} for F6A and F6B with the same values for A and p, indicating that the introduction of epistatic interactions between the decision variables for F6 did not have a

Table 3.15 g_{conv} values observed for F6A and F6B.

		F6A		F6B			F6A		F6B	
		p		p			p		p	
A	l	2	3	2	3	l	2	3	2	3
2	5	80	116	84	109	20	319	323	338	321
3		117	127	130	124		403	350	339	344
2	10	166	174	179	175	30	529	483	473	462
3		166	168	167	163		459	493	499	510

significant effect on the best performing GA parameter values.

3.2.3 Observed Effect of Characteristics on the Number of Generations

3.2.3.1 Roughness and Multimodality

An increase in the roughness of a fitness function was considered for functions F3B, F4A and F4B, through changes to the A parameter. The degree of 'roughness' of a function is related to the global structure of a function; a very rough function has no global structure, and therefore there is less information in the fitness function to guide the GA to find better solutions; however, for a function with less roughness it will be easier for the GA to proceed to the global optimum of the function, provided there is structure in the fitness function to lead toward the better solutions. In general, it was observed that an increase in the roughness of a function produced an increase in g_{conv}. The implication of this result is that a smaller population size is more effective for functions that have a greater roughness. For rougher functions, the GA is relying on mutating the best solutions, as there is no structure in the fitness function to benefit the search, and a larger population size will only waste function evaluations. The effect of a change in the roughness of F4, produced by a change in the A parameter for F4A and F4B was not as substantial as the effect seen by a change in the roughness of F3B. The reasons for this are outline above in Section 3.2.2.2, where changes in A had little effect on the problem characteristics in higher dimensions.

Changes to the multimodality of a function were considered for function F3B, by making changes to the f parameter. The result observed was that an increase in the multi-

modality of the function produced a slight decrease in the value of g_{conv}. For a given FE, a decrease in g_{conv} indicates that a larger population size performed better for functions with more local optima. This result can be explained by the fact that a larger population size will contain more solutions distributed over the search space, and therefore will be less likely to become trapped in the local optima that are present. Therefore, a larger population size is more likely to and allow the search to proceed to the better regions of the search space.

3.2.3.2 Salience of Variables

Variations in the salience of one or more of the decision variables was considered for all three of the Optimal Generation Functions identified in Section 3.1. For all the variations of the test functions considered, it was observed that an increase in the salience of the decision variables produced a increase in g_{conv}. The consequence of this result is that for an available number of FE, a smaller population size is better for functions that have a higher salience of at least one decision variable. This may not have been the most expected result, as it might be assumed that a larger population size would be more beneficial for a more salient problem, as an increased diversity in the population would avoid the occurance of genetic drift once the more salient decision variables have converged. However, mutation can be used to inject diversity into the population, and when only a small percentage of the decision variables have a significant effect on the fitness function value, function evaluations are wasted when a large population size is used.

This result is in agreement with the modelling results of Thierens and Goldberg (1994) and Thierens et al. (1998), where it is proposed that for a uniformly distributed function, g_{conv} will scale $O(\sqrt{l})$, and a function with exponentially scaled variables g_{conv} will scale $O(l)$. Therefore, as the problem size grows more generations are expected for functions with higher salience, as observed in the results presented here.

3.2.3.3 Epistatic Interactions

The effect of epistatic interactions between the decision variables has also been considered for all three of the Optimal Generation Functions. The results for F3A in Table 3.11 suggested that an increase in g_{conv} occurred with an increase m_{BB} or δ_{BB}, however, in general, g_{conv} decreased when pair-wise interactions were introduced to F4B when compared to F4A. A change in the degree of epistatic interaction for these functions may also produce a change in the roughness of the function, which may account for the conflicting observations. The results presented in Table 3.15 for the introduction of interactions into

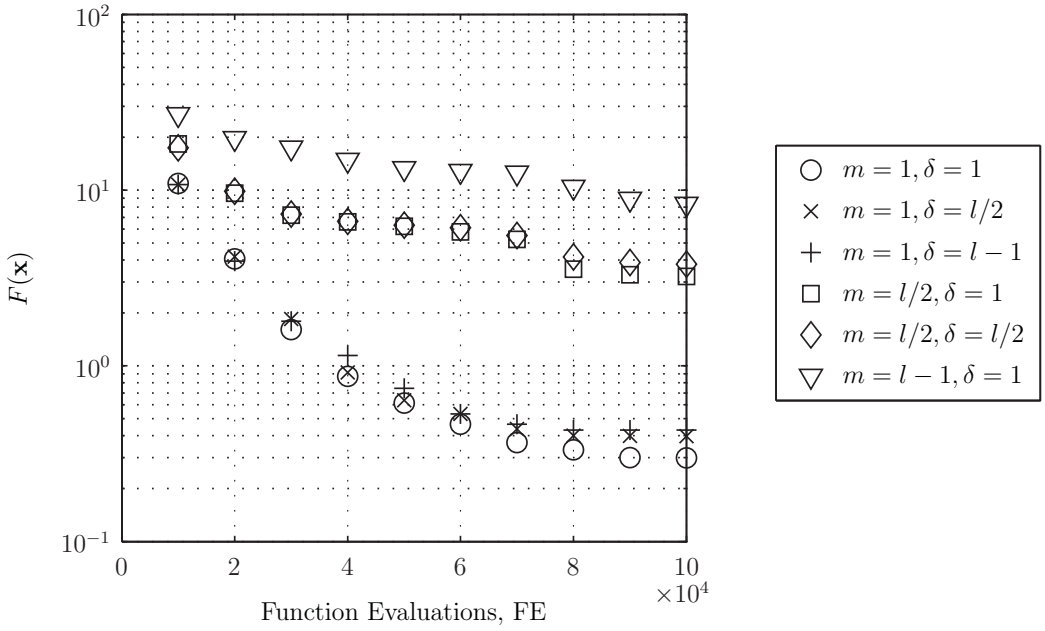


Figure 3.18 The effect of interactions on the solution found for F3A with $l = 30$.

F6 indicate that for this case, the interactions have very little effect on the observed value of g_{conv}. Therefore, the results presented in Section 3.2.2 suggest that a change to the degree of interaction between variables did not have a significant effect on g_{conv} for the cases considered.

While the results suggest the degree of interaction between the decision variables had little effect on g_{conv}, this characteristic did have a large effect on the solution quality found. In Figure 3.18 the best solution found from the average of the 30 different initial population runs for the parametric study for different FE has been plotted for each variation of interactions considered for F3A in $l = 30$ dimensions. It must be noted that the solutions plotted for different FE do not necessarily come from the same set of GA parameter values. For example, the solution plotted for F3A with $m_{\text{BB}} = l - 1$ and $\delta_{\text{BB}} = 1$ at 10 000 function evaluations was produced from a GA run with a population size of $N = 10$, however, the solution plotted for $FE = 100\,000$ for the same function was produced from a GA run with a population size of $N = 800$.

From Figure 3.18, it can be seen that the number of interactions, m_{BB}, has a large impact on GA performance, where for more interactions, the GA found it much more difficult to find better solutions. For the function with $m_{\text{BB}} = 1$ and $\delta_{\text{BB}} = 1$, the GA found a solution of $F(\mathbf{x}) = 0.30$ after $FE = 100\,000$. In contrast, for the function where

m_{BB} is increased to $m_{\text{BB}} = l - 1$, the best parameter values for the GA could only find an average best solution of $F(\mathbf{x}) = 8.33$. δ_{BB} also had a slight impact on GA performance, where the GA was able to find slightly better solutions when the distance between the interacting variables was shorter. However, the difference was negligible when compared to the impact of an increase in m_{BB}.

3.2.3.4 Problem Size

Every variation of every function considered has been optimised in problem sizes of $l = 5$, 10, 20, and 30. The results presented in Section 3.2.2 indicate that of all the problem characteristics considered, problem size has by far the largest effect on the value of g_{conv}. This is an expected result, as the GA will require more generations to effectively process the larger number of decision variables, and is in agreement with a number of theoretical modelling results, for example the population sizing rule (Harik et al., 1999) outlined in Section 2.4.3, in the form of the number or size of building blocks.

3.2.4 Discussion of Characteristics Results

The hypothesis tested in this section was that, if a function is an Optimal Generation Function, the value of g_{conv} is related to the characteristics of the function. The results presented in Section 3.2.2 validate this hypothesis for the cases considered, where a number of the function characteristics considered produced a change g_{conv} observed from the parametric study.

Both the roughness and multimodality resulted in predictable changes in g_{conv}, however, changes to the roughness of the function produced the largest changes in g_{conv}. Similarly, highly salient decision variables had a large effect on the most suitable value for g_{conv}, where the more salient one or more of the decision variables were, the larger the value of g_{conv}.

Somewhat surprisingly, introducing epistatic variables did not have a significant effect on the value of g_{conv} for the cases considered. However, this function characteristic did have a large effect on the solution quality found. Both changes to the number of interactions, and distance between the interactions in the solution string, affected the solution quality found. However, the number of interactions had a much larger influence on the ability of the GA to find good solutions compared to the distance between the interactions. This was also a somewhat surprising result, as the GA used in the parametric study used a one point crossover operator, which would be expected to process adjacent interactions very well, but would be expected to be quite disruptive to interactions between

variables far apart in the solution string. However, from Figure 3.18 it can be seen that the GA found solutions of very similar quality for these two cases, with the solution found for functions with epistatic interactions between adjacent decision variables only slightly better than the solution found for similar functions with interactions between variables further apart in the solution string.

3.3 SUMMARY

The first part of this chapter has investigated the hypothesis that under tournament selection, there exists a constant number of GA generations to most efficiently solve a given optimisation problem. A suite of test functions has been considered, each for a range of problem sizes. Two different function types have been identified, each having an identifiable GA calibration method for making the most efficient use of the function evaluations that are available. The function types identified were: Maximal Generation Functions, where the most efficient method to solve the problem uses a small population size updated as many times as possible; and Optimal Generation Functions, where there is an optimal number of generations to efficiently solve the problem, irrespective of the number of function evaluations available.

Therefore, the Optimal Generation Functions that have been identified have validated the initial hypothesis, where if the test function has a characteristic that benefits from being solved with a larger population size (such as interactions between decision variables, or significantly different contributions from the decision variables to the fitness function value), a constant number of generations exists that will locate the best solutions possible for the problem.

The second part of this chapter has investigated the hypothesis that the constant number of generations to most efficiently solve a problem is related to the characteristics of the fitness function. The function characteristics considered were those identified to effect the different function types in the first section, namely: multimodality or roughness, salience, epistatic interactions and problem size. A number of variations of the functions belonging to the Optimal Generation Functions identified were considered, each with different, controllable characteristics.

The second hypothesis tested in this chapter was also validated for the cases considered, with strong relationships identified between all the characteristics considered and the observed number of generations from the parametric study. The only exception was the effect of changing the degree of epistatic interactions between the variables, which did not significantly change the number of generations until convergence. However, this

characteristic was observed to have a large effect on the solution quality found.

Therefore, this chapter has presented two important results regarding GA calibration:

1. that for functions with certain characteristics, there exists a constant number of generations to most efficiently solve the problem, irrespective of the convergence criteria; and
2. that the actual number of generations is related to the characteristics of the problem.

Hence, it is proposed that the characteristics of the function can be used to predict the number of generations to most efficiently solve an optimisation problem.

For the problems considered in this chapter, the fitness function characteristics can be inferred by the mathematical form of the function. However, for many 'real-world' optimisation problems, such as WDS optimisation problems, the characteristics of the fitness function are largely unknown. For fitness functions such as this, the following chapter develops a number of statistics that can be applied to provide information about the characteristics of the fitness function.

Chapter 4
Development of Fitness Function Statistics

In Chapter 3, it was determined that the number of generations before the GA population converged for certain types of fitness functions considered was a function of the characteristics of the function. However, for WDS optimisation, as well as many other optimisation problems, the fitness function is derived from results of a simulation model, and the characteristics of the fitness function are largely unknown. For cases such as this, it is proposed in this chapter that statistics computed from a sample of fitness function values can be used to provide information about the characteristics of the fitness function.

A number of studies (Kinnear, 1994; Naudts, 1998; Naudts and Kallel, 2000) have indicated that fitness function statistics are a poor indicator of GA performance. However, these studies did not consider the calibration of the algorithm, and hence the convergence observed was rather arbitrary. For example, a GA with one set of parameters may perform poorly, however with different parameters the algorithm may find the optimum every time. This research proposes that the most suitable GA parameters are a function of the characteristics of the problem, and the fitness function statistics provide information about these. This approach is different to that generally adopted when making use of these statistics, which typically attempts to relate fitness function statistic results to how closely a GA will converge to the optimum solution. However, this is not only a function of the fitness function characteristics, but also the GA parameter values and the convergence criterion.

The fitness function characteristics that were identified to affect the values of g_{conv} in Chapter 3 were roughness and multimodality, salience of the decision variables and problem size. The problem size is known for any fitness function, and is given by the solution string length. The roughness of the function can be estimated by a correlation

measure, such as the autocorrelation measure (Weinberger, 1990). However, concerns regarding the most common method used to calculate this measure, by the use of a 'time series' of fitness function values, were outlined in Chapter 2. To more accurately characterise fitness function characteristics, a spatial correlation measure is proposed and tested in Section 4.1. The method proposed by Seo et al. (2003) to estimate the salience of variables by the use of MI is tested in Section 4.3.

The degree of epistatic interactions was not observed to affect the most suitable GA calibration in Chapter 3, however, this characteristic had a large effect on the solution quality found. If the interaction between variables could be detected, then the solution string could be rearranged to allow for more efficient processing of the best combination of values in the GA population. A number of common statistics for estimating epistasis, such as the epistatic variance, cannot detect the degree of interaction, only the presence of an interaction (Naudts, 1998). The gene epistasis measure proposed by Seo et al. (2003) is tested on simple test functions, and limitations in estimating the necessary parameters in a real parameter space are identified. A new sampling method is introduced in Section 4.2.2 to address these limitations, and the sampling method is tested on a number of test functions with known characteristics.

4.1 Spatial Correlation

The spatial autocorrelation technique is based on the statistic known as Moran's I (Moran, 1948), and has been adopted to overcome the problems associated with using a series of fitness function values to represent a l–dimensional search space. For the spatial autocorrelation, the arrangement of the function values is expressed using a weighting function. A spatial weighting function consists of a set of rules for assigning values to pairs of sample values to represent their arrangement in space. The most common spatial weighting function for a data set specifies that the weight for a pair of points i and j is one if i and j are nearby in the search space, and zero otherwise (Odland, 1988).

Typically, the weighing function is used to locate which pairs of sample points are a similar Euclidean distance apart from each other in the search space, to allow their function values to be correlated to provide information about the surface. The advantage of the weighing function is that it relates the actual position of points in the search space to each other. This is different from the traditional time series approach, which uses the number of steps between points where the function is sampled. This has the potential to be misleading, as only a small portion of the search space may be covered and the potential

exists for values to be a large number of steps apart in the generated 'time series', but actually to be very close to each other in the landscape.

The weighting function can be sequentially altered to provide more information about the landscape further away than the closest neighbours. By changing the weighting function to identify pairs of data points further and further apart from each other, the correlation between function values can be calculated for a number of distances, and a typical autocorrelation curve of distance against correlation can be produced.

Once the weighting function has been defined, the spatial correlation (R_s) is computed by:

$$R_\mathrm{s}(d) = n \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij,d} \left(x_i - \overline{x}\right) \left(x_j - \overline{x}\right)}{\sum_{j=1}^{n} w_{ij,d} \sum_{i=1}^{n} \left(x_i - \overline{x}\right)^2}, \tag{4.1}$$

where n is the number of samples; $\overline{x}$ is the mean value of the function values; x_i is the i^{th} function value sampled; and $w_{ij,d}$ is the value of the weighting function, for values i and j that are d apart in the search space. R_s has an expected value of $-1/(n-1)$ (Odland, 1988), which tends toward zero as the number of samples increases. If the x_i values are independent of the x_j values at neighbouring locations, the calculated value of R_s should equal this expected value, within the limits of statistical significance. In this case, the function values are unrelated to each other on average, and it would be expected that there is no information in the fitness landscape to guide the GA toward better solutions. Values of R_s that exceed the expected value indicate positive spatial autocorrelation, for which fitness values of x_i tend to be more similar to neighbouring values compared to the average of the samples. When this is the case, there may be some useful structure in the landscape to guide the algorithm toward better solutions. Correlation values below the expectation indicate negative spatial autocorrelation, where pairs of values are dissimilar. A negative correlation may also provide useful information to guide the optimization algorithm, as this indicates that on average, function values are significantly different from one another.

The weighting function used here correlates objective function values that are similar distances apart in the objective function search space, which is not necessarily the fitness landscape the GA is operating on. However, in this work, it is assumed that the fitness function is a reasonable approximation to the fitness landscape, the assumption being that the distance between points is a suitable surrogate for the GA fitness landscape, as it is more likely that a GA operator will produce new solutions that are close to the solutions that were used in the operation. Also, in the previous chapter it was found that character-

istics of the fitness function itself affected the most suitable value of g_{conv}, implying the fitness function provides a useful approximation to the true fitness landscape.

4.1.1 Methodology

To investigate the accuracy of the existing and proposed correlation measures, a number of experiments have been performed. In order to compare the accuracy of the correlation measures, functions with a known autocorrelation have been used. The Wiener–Khinchin Theorem (Bracewell, 1965) provides that the autocorrelation of a function, $R(d)$, is given by the Fourier Transform of the power spectrum of the Fourier Series of the function. The theorem can be used to determine the exact form of the correlation function for a series of sine waves, as if:

$$f(x) = \sum_{i=1}^{m} A_i \sin(f_i x + \phi_i), \tag{4.2}$$

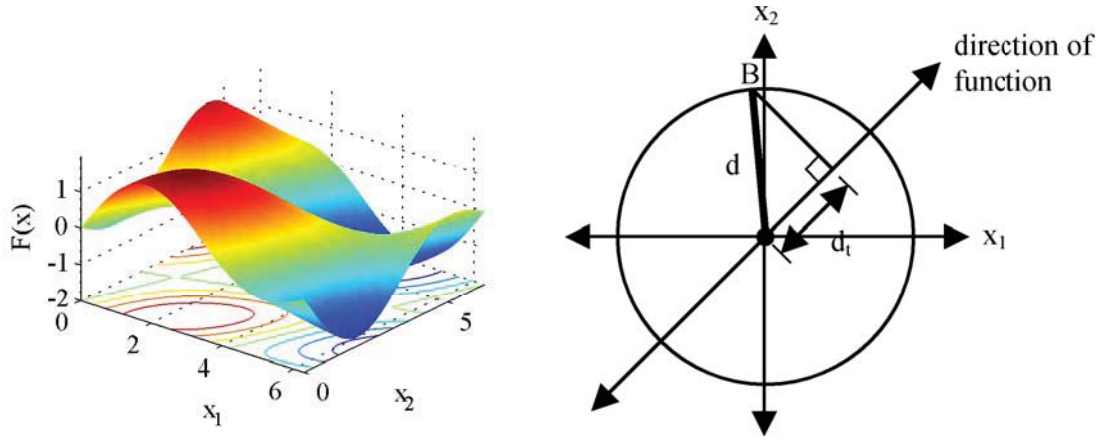
then the autocorrelation function is given by (Bracewell, 1965):

$$R(d) = \frac{\int_{-\infty}^{\infty} f(x) f(x+d) dx}{\int_{-\infty}^{\infty} (f(x))^2 dx}, \tag{4.3}$$

$$= \frac{1}{\sum_{i=1}^{m} A_i^2} \sum_{i=1}^{m} A_i^2 \cos(f_i d), \tag{4.4}$$

where: m is the number of sine terms that compose the function; d is the correlation lag, or distance between points; and A_i, f_i, and ϕ_i are the amplitude, frequency, and phase shift of each term i, respectively. Using this relationship, the true autocorrelation of the function is known, and therefore a direct comparison between the exact correlation structure of a fitness function, and that estimated by the correlation statistics, can be made.

The above relationship is only defined for functions in one dimension, however optimization problems are generally highly multi-dimensional. To test both the temporal and spatial correlation measures, the autocorrelation function in Equation 4.4 has been extended into two dimensions. To achieve this, each sine function in the Fourier Series has been rotated into the second dimension, thus projecting each sine wave along a line trajectory in two dimensions. Each sine function can be rotated by a different angle, producing complex multi-modal functions. As the function is still comprised of superimposed sine waves, the Fourier Transform of the function provides the true autocorrelation of the function. Note that the autocorrelation function does not take the form of an exponential decay, as suggested by Weinberger (1990). This can be explained by the fact



(a) Two 1D sine waves rotated to produce a 2D function.

(b) The method used to compute the correlation of the 2D functions.

Figure 4.1 Extension of the Wiener–Khinchin Theorem into two dimensions.

that the functions are not first order autoregressive functions, which was assumed in that work.

An example of the superposition used can be seen in Figure 4.1(a), comprised of two simple sine waves; $y = \sin(x)$ with a line of trajectory of $x_2 = x_1$, and $y = \sin(2x)$ with a line of trajectory of $x_2 = -x_1$. Even from this simple example, it can be seen that complex, multi-modal functions can be constructed in this fashion. However, by using the approach outlined above, the autocorrelation of this function is known.

Similarly to the one dimensional case, the autocorrelation along the line of the rotated function is known, as defined by Equation 4.4. However, it is necessary to compute the average correlation in all directions, not just along the line of trajectory of the function. Perpendicular to the line of trajectory of the function, the contribution from each sine wave to the fitness function value is the same, therefore the autocorrelation due to each individual wave perpendicular to its line of trajectory is $R = 1$. In directions between these two extremes, the correlation is also between the corresponding correlation values. The value of the correlation function for a given distance can therefore be computed by taking the average value in all directions for a given correlation distance.

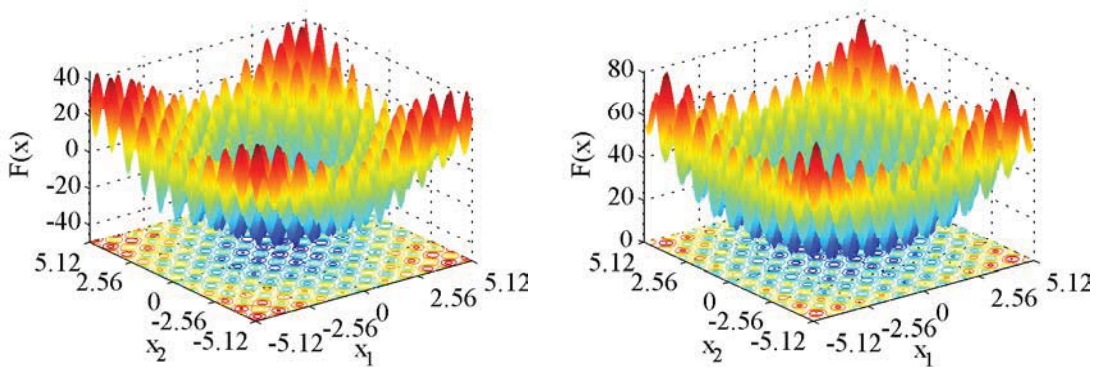
The method used to compute the correlation can be seen in Figure 4.1(b), where the correlation is computed for the distance, d. To compute the correlation in the direction of B, the distance, d, is projected to give a distance, d_t, along the line of the function. Therefore, the correlation computed using d_t in Equation 4.4 produces the correlation for the function in the direction of B for a distance of d. From Figure 4.1(b), it can be seen

that perpendicular to the line of the function, $d_t = 0$, producing $R = 1$, as expected. The correlation for distance d is computed by taking the average of the correlation values calculated at a distance d for a sample of points in all directions for all sine waves. 128 samples have been used to compute the average autocorrelation for a given distance in this work. While a two dimensional function such as this has a much lower dimensionality than the highly dimensional problems typically tackled with GAs, the methodologies tested are based on the Euclidean distance between two points, and therefore can be directly extended to higher dimensions.

Using this technique, the correlation computed using the temporal and spatial autocorrelation functions can be compared with the true autocorrelation of the function. Fourier Series functions composed to approximate the variations of the Rastrigin Function used for function F3B in Chapter 3 have been used as the functions to test the correlation statistic. Two tests can be conducted by making use of these functions; 1) the accuracy of the temporal and spatial correlation statics can be compared against the known autocorrelation, and 2) the results from different functions can be compared to determine if the statistic can identify changes to the function characteristics.

As the functions of known autocorrelation are comprised of sine waves, the x^2 term in the Rastrigin Function must be approximated by a sine wave. Given that the range of the decision variables was $[-5.12, 5.12]$, a frequency of $f = 2\pi/(5.12 - -5.12) = 0.6136$ must be used. Similarly, $A = \frac{5.12^2}{2} = 13.1072$ has been implemented to provide the correct amplitude. In order for the resulting sine wave to produce the 'big bowl' structure, a phase shift of $\phi = -\pi/2$ was also used. A subsequent sine wave is then used to provide the roughness and multimodality on the function, and both waves are also applied in the second dimension. Therefore a total of four sine waves are used to produce the function shown in Figure 4.2(a), and this is compared against the true Rastrigin Function with parameters $A = 10$ and $f = 2$, shown in Figure 4.2(b). Note that the minimum of the approximated function is not $F(\mathbf{x}) = 0$, as is the case for the true Rastrigin Function. However as the correlation statistic is based on the difference between function values, this will not affect the results. As was the case for function F3B, values of $f = 2$ and 5 have been tested for the roughness sine term, along with $A = 10$ and 20.

For the temporal autocorrelation statistic, a random walk has been used, as proposed by Weinberger (1990). If a random sample of points is taken in the search space to compute the spatial correlation statistic, the resulting distribution of distances between the sample points is close to normal. Therefore, there are many samples of fitness function values that can be used to compute the correlation for points that are approximately half



(a) Fourier Series Approximation (b) True Rastrigin Function

Figure 4.2 The Rastrigin function and Fourier Series approximation to the Rastrigin Function used to compute the correlation statistics.

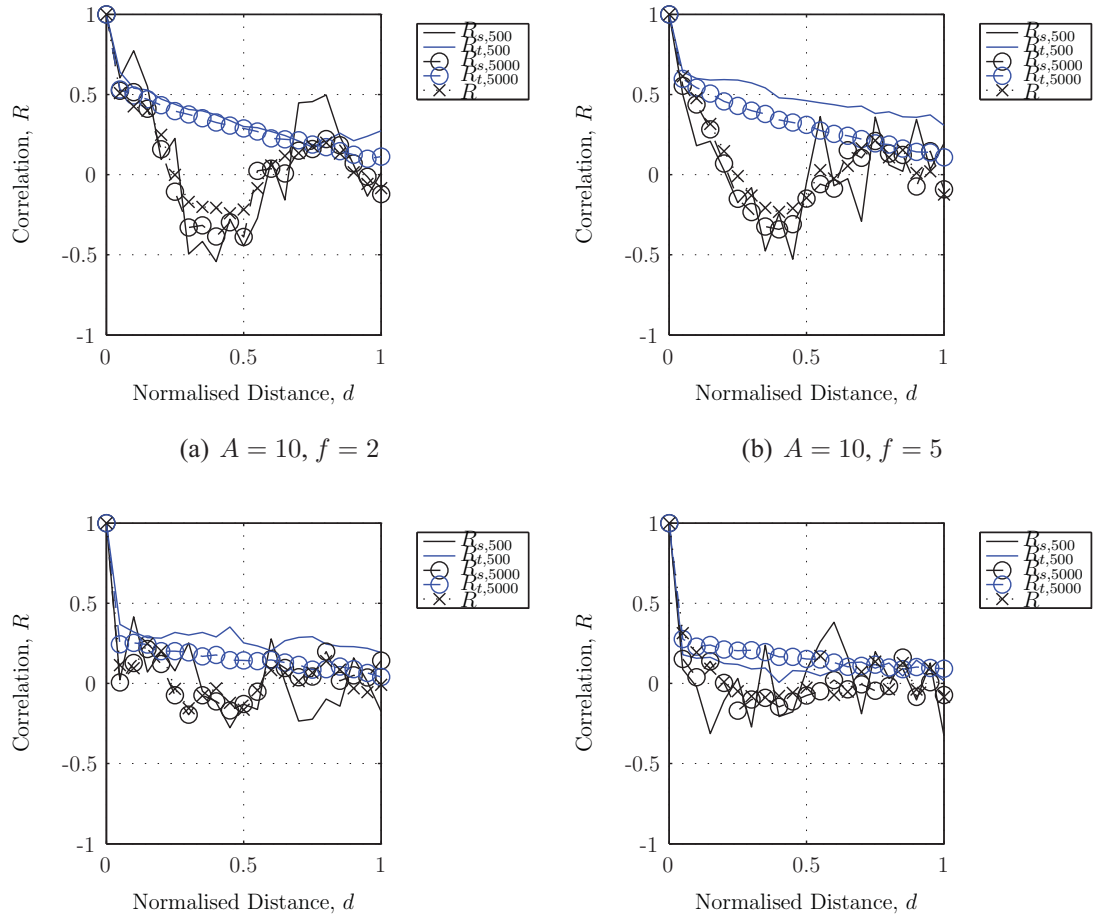
the extent of the search space away from each other, but many fewer are available to be used to compute the correlation between points that are very close together or very far apart in the search space. To characterize a search space, we may only be interested in points that are relatively close together, where some correlation is still detectable. Hence, a random sampling method may be very inefficient in this instance, as most pairs of points to be correlated are further apart in the search space than the correlation length, or the detectable correlation in the fitness function.

A structured sampling approach has been adopted to overcome this problem, and to sample the necessary data to compute the spatial correlation. This approach involves randomly generating an initial n_c samples, as given by:

$$n_c = n/\left(\frac{1}{\text{step size}} + 1\right), \tag{4.5}$$

where n is the total number of samples required, n_c is the number of samples used for each correlation computation, and step size is the percentage of the extent of the search space to use as the distance between points for each correlation value. From each of the n_c initial sample points, another sample point is generated one step size away, producing n_c function values to be correlated for the initial distance of one step size. Then, another set of points is generated two step sizes from the initial n_c sample points, to be correlated for the next distance. The process continues until the whole search space has been covered. In this way, the specified number of samples, n, is still used, however only n_c samples are used to compute each correlation value.

For the analyses conducted, a step size of 0.05% of the extent of the search space has



(a) $A = 10$, $f = 2$

(b) $A = 10$, $f = 5$

(c) $A = 20$, $f = 2$

(d) $A = 20$, $f = 5$

Figure 4.3 Temporal (R_t) and spatial (R_s) correlations compared to the true correlation (R).

been adopted. As the range of each decision variable is $x_i = [-5.12, 5.12]$, the step size corresponds to a distance of $d = 0.05\sqrt{(5.12 - -5.12)^2 + (5.12 - -5.12)^2} = 0.724$. The sample sizes tested were $n = 500$ and 5000 samples.

4.1.2 Results

Results for both the temporal and spatial correlation statistics, as well as the analytic autocorrelation, are presented for the four functions considered (Figure 4.3). R_s indicates the correlation computed using the spatial correlation statistic, and R_t indicates the correlation computed using the temporal correlation statistic. R provides the true correlation of the function, as given by Equation 4.4 extended into two dimensions.

For points very close together in the search space, all the methods considered produce

correlation values very close to the true autocorrelation. For distances further apart than the first two or three steps, the temporal methods can be seen to overestimate the true correlation value. This result can be attributed to using the order that points are sampled in as a surrogate for the distance between two points, as for points further than a few step sizes apart, it is unlikely that this approach will provide a reasonable representation of the true distance between points. The overestimation of the autocorrelation of a function will in turn suggest a longer correlation length than the function actually has. This overestimation may be a significant concern when using the temporal correlation statistic for function characterisation, as it provides misleading information about the characteristics of the function.

Generally, the spatial correlation statistic provides a very close approximation to the true autocorrelation for the functions considered. It can be seen from Figure 4.3 that for $n = 500$ samples, the spatial correlation statistic produces a very noisy approximation to the true correlation. In this case, each correlation value is being computed using $n_c = 500/(\frac{1}{0.05} + 1) = 23$ samples, producing large errors in each calculation. However, when the sampling is increased to $n = 5000$ samples, then $n_c = 5000/(\frac{1}{0.05} + 1) = 238$ samples are used to compute each correlation value, and accurate approximations to the true autocorrelation can be seen for all the functions considered.

The other information that can be obtained from these results is whether the statistics can actually provide useful information about changes in the characteristics of the function. The correlation computed using the spatial correlation statistics with $n = 5000$ for the four functions considered is presented in Figure 4.4. The results indicate that the spatial correlation statistic can in fact provide very useful information about changes in the roughness of a function, where the correlation computed with $A = 10$ is significantly higher than that computed for the functions with $A = 20$. The statistic is less able to predict changes in the multimodality of the function, produced by a change from $f = 2$ to $f = 5$. However the increase in multimodality does produce a slight decrease in the correlation detected. It was found in Chapter 3 that the increase in multimodality only produced a slight change in the value observed for g_{conv}, and therefore this result is acceptable.

4.1.3 Discussion

From these analyses, it can be concluded that the traditional temporal correlation statistic is unable to represent a two–dimensional space as a one dimensional series of function values. This approach leads to the overestimation of the true correlation of a function.

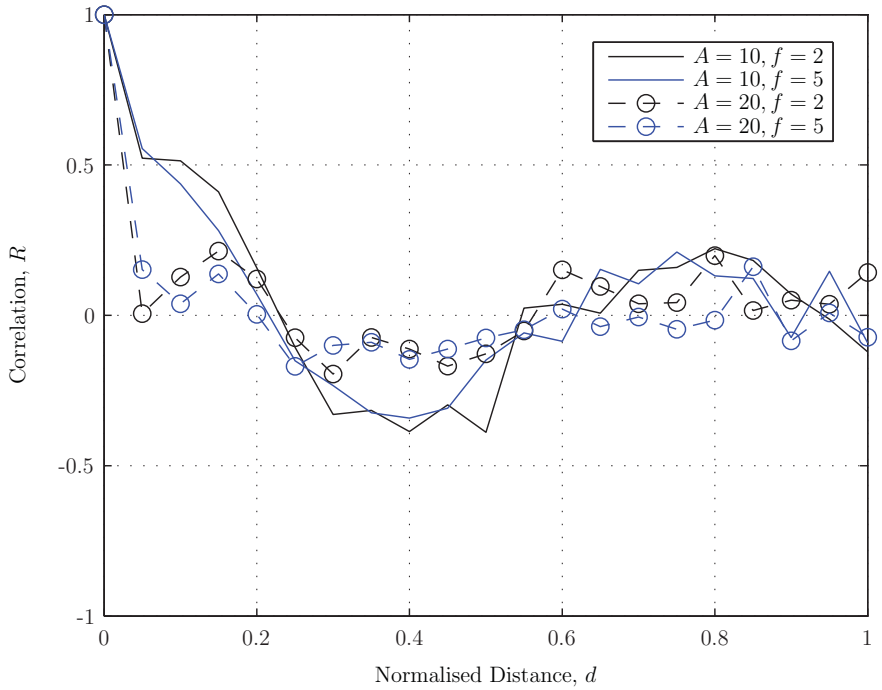


Figure 4.4 Comparison of the spatial correlation function computed for variations of the approximated Rastrigin Function

The proposed spatial correlation statistic provides a more accurate representation of the search space, and therefore a more accurate estimation of the true correlation of a function. Hence, the spatial correlation statistic with structured sampling is proposed as an appropriate method to provide useful information about the structure of a fitness function. Not only is the statistic accurate, but it can also provide relevant information about changes to the fitness function that produced changes in the number of GA generations before convergence.

4.2 Epistatic Interactions

As outlined in Section 2.4.1.2, there are a number of concerns regarding the application of traditional epistasis variance measures to quantify the epistasis of a fitness function, namely (Naudts and Kallel, 2000): i) the statistic can only provide information about the presence of epistasis, not a measure of the degree of epistasis; and ii) the epistasis measure is sensitive to nonlinear scaling of the fitness function, something a GA with tournament or ranking selection is entirely insensitive to. Therefore, in this section the gene epistasis measure proposed by Seo et al. (2003), also outlined in Section 2.4.1.2, is

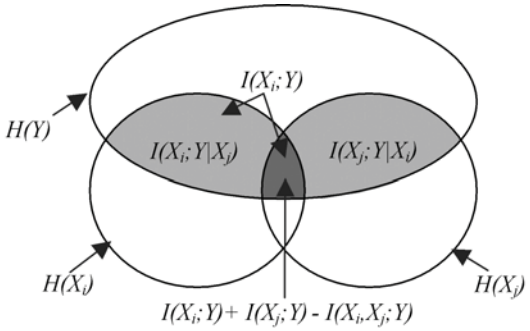


Figure 4.5 Representation the interaction between decision variables in terms of Mutual Information.

tested on two simple problems; one with epistatic interactions, and one without. The main drawback of the gene epistasis measure is that it requires the joint MI, $I(X_i, X_j; Y)$, to be estimated. Computing this value directly requires the three dimensional pdf of a sample of data, $p(x_i, x_j, y)$, and therefore a large data set to provide a reasonable estimate, due to the relatively high dimensionality.

4.2.1 Testing the Gene Epistasis Measure

To avoid the need to estimate a three dimensional pdf, the interaction between decision variables can also be estimated using the difference between the MI, $I(X_i; Y)$, and the conditional MI, $I(X_i; Y|X_j)$. The relationship between these two terms can be seen in Figure 4.5. If the relationship between X_i and Y is independent of the value of X_j, then $I(X_i; Y) = I(X_i; Y|X_j)$. Otherwise, the degree of interaction between decision variables can be estimated by the size of the non-overlapping region in Figure 4.5, given by $I(X_i; Y) - I(X_i; Y|X_j)$. However, this approach is not without its own shortcomings, as in order to compute the interaction between decision variables using the conditional mutual information, $I(X_i; Y|X_j)$, the residuals of the fitness function value given one decision variable, $Y|X_j$, must be estimated. This is not a straightforward procedure, as the relationship between Y and X_j is generally unknown.

To investigate the impact of these practical aspects on the accuracy of the gene epistasis measure, both approaches outlined above (estimating the full three dimensional pdf and estimating the residuals of Y given X_j) have been tested experimentally. To compute the mutual information terms, the pdfs of a sample of data must be estimated. The most common approach adopted for non-parametric density estimation is the histogram, and this approach has been adopted in this work. Nonparametric density estimation methods such as this can be used to provide a direct approximation to the distribution of a random

Table 4.1 Gene epistasis test functions.

	Function
F1	$x_1 - x_2$
F2	$(x_1 - x_2)^2$

variable based on structure in the data, without needing to parameterise a model that may or may not be suitable for the distribution of the data. The classical frequency histogram is formed by constructing a set of non-overlapping intervals, or bins, and counting the number of samples that fall into each bin (Scott, 1992). There are two parameters that determine a histogram, the bin width, h, and the starting point of the first bin, as this will determine which bin a given sample is allocated to. The bin width, h, is commonly referred to as a smoothing parameter, as it controls the amount of smoothness in the estimator for a given sample size, n. The bin width reference rule for the asymptotically optimal bin width for normally distributed, multivariate data has been adopted to determine the size of the histogram bins (Scott, 1992):

$$\hat{h_i} = 3.5\hat{\sigma}_i n^{-1/(2+l)}, \tag{4.6}$$

where $\hat{h_i}$ is the bin width in dimension i, $\hat{\sigma}_i$ is the standard deviation of the sample data in dimension i, n is the number of samples in the data set, and l is the dimension of the data. The reference rule is based on finding the best trade-off between a large h to reduce variance, and a small h to reduce bias, producing the minimum Mean Squared Error (MSE) of the estimates. The approach proposed by Moddemeijer (1989) to explicitly account for the bias and variance in a histogram estimate for computing entropy and MI has been adopted in this work.

Two simple test functions have been selected to test the accuracy of the gene epistasis measure for detecting an interaction between decision variables. The functions considered are given in Table 4.1. F1 is the difference between two variables, hence no interaction between the random variables is expected. F2 is calculated from the squared value of the difference between the two random variables, therefore for this function a strong interaction between variables is expected. To compute the gene epistasis, each random variable has been generated from the distribution $x_i \in N(0, 1)$. The values presented for all experiments have been computed using $n = 5000$ samples, and the results presented have been averaged over 30 different random data sets.

Table 4.2 Interactions by mutual information and joint probability.

	F1	F2
$I(X_1;Y)$	0.320	0.071
$I(X_2;Y)$	0.333	0.074
$I(X_1,X_2;Y)$	1.300	0.456
$I(X_1;Y)+I(X_2;Y)-I(X_1,X_2;Y)$	-0.648	-0.310

4.2.1.1 Method 1: Gene Epistasis by Joint Mutual Information

This approach to computing the gene epistasis involves computing the joint MI directly using Equation 4.7:

$$I(X_i,X_j,Y)=\sum_{x_i\in X_i}\sum_{x_j\in X_j}\sum_{y\in Y}p(x_i,x_j,y)\log\frac{p(x_i,x_j,y)}{p(x_i,x_j)p(y)}. \tag{4.7}$$

From Equation 4.7, it can be seen that the three dimensional pdf, $p(x_i,x_j,y)$, must be estimated in order to compute $I(X_1,X_2;Y)$. The joint MI can then be used to compute the interaction by comparing it to the sum of the individual MI terms, $I(X_1;Y)+I(X_2;Y)$.

The results for computing the interaction between decision variables using the joint MI can be seen in Table 4.2. As expected, $I(X_1;Y)\approx I(X_2;Y)$, as both random variables have a similar contribution to the dependent variable, Y. It would be expected that the difference between the sum of the MI terms and the joint MI would be higher for F2 than F1, due to the interaction between the variables. This is reflected in the results, with a difference of -0.31 for F2 and -0.648 for F1. The theoretical minimum value for this difference is zero, occurring when there is no interaction between the decision variables, and $I(X_1;Y)+I(X_2;Y)=I(X_1,X_2;Y)$. However, for both functions, the value for the joint MI, $I(X_1,X_2;Y)$, is significantly higher than the sum of the individual MI values, $I(X_1;Y)+I(X_2;Y)$. It is likely that the results in Table 4.2 can be attributed to a poor estimate of the three dimensional pdf that is required to determine the value of $I(X_1,X_2;Y)$, even for these simple functions and a relatively large sample size of $n=5000$.

4.2.1.2 Method 2: Gene Epistasis by Fitness Function Residuals

The previous results suggest that it is not practical to estimate the three dimensional pdf, $p(x_i,x_j,y)$, as even for the very simple functions considered, the interaction between variables cannot be estimated accurately. However, by estimating the MI between one variable X_i, and the fitness function value given the effect of another variable, $Y|X_j$,

Table 4.3 Interactions by mutual information and residuals.

	F1	F2
$I(X_1;Y)$	0.320	0.071
$I(X_1;Y\|X_2)$	1.991	0.255
$I(X_1;Y) - I(X_1;Y\|X_2)$	-1.671	-0.184

($I(X_i;Y|X_j)$) the effect of both variables, X_i and X_j, on the function value, Y, can be estimated without the need to estimate the three dimensional pdf.

$Y|X_j$ can be computed by the difference between the best estimate of Y given the values of X_j (Y_{X_j}) and the original values of Y. However, it is difficult to approximate the series Y_{X_j} accurately, as generally the relationship between Y and X_j is unknown. A simple linear regression could be used to estimate Y_{X_j}, however, if the relationship between Y and X_j is highly nonlinear, this approach will provide poor estimates of Y_{X_j}, and therefore potentially misleading values for $I(X_i;Y|X_j)$.

To avoid assuming a form of the regression equation, the series Y_{X_j} has been estimated using nonparametric kernel regression. The theoretical regression function is defined to be (Scott, 1992):

$$r(x) = E(Y|X=x) = \int yp(y|x)dy = \frac{\int yp(x,y)dy}{\int p(x,y)dy}, \tag{4.8}$$

where $E(Y|X=x)$ is the conditional expectation of y for a given observation, x. The nonparametric kernel regression estimator with kernel K of Equation 4.8 is:

$$\hat{r}(x) = \frac{\frac{1}{2}\sum_{i=1}^{n} y_i K_{h_x}(x-x_i)}{\frac{1}{2}\sum_{i=1}^{n} K_{h_x}(x-x_i)}. \tag{4.9}$$

Then, the series $Y|X_j$ can be computed as:

$$(Y|X_j)_i = Y_i - \hat{r}(X_j). \tag{4.10}$$

To compute $Y|X_2$ for the two test functions, a Gaussian kernel with reference bandwidth (Scott, 1992) has been adopted. The results for computing the interaction between F1 and F2 using the residuals of Y given X_2 can be seen in Table 4.3. The values presented are the average values from 30 different random data sets, computed using $n = 5000$ samples.

From the results, it can be seen that for both functions, $I(X_1;Y|X_2) > I(X_1;Y)$. For F1 with independent variables, it would be expected that $X_1 \approx Y|X_j$, as without the effect of X_2 the value of Y is equal to X_1. Therefore, $I(X_1;Y|X_2) \approx I(X_1;X_1) = H(X_1)$.

By definition $I(X_1;Y) \leq H(X_1)$, therefore the result that $I(X_1;Y|X_2) > I(X_1;Y)$ is reasonable for F1 in this case. However, this results in a negative interaction term, computed by $I(X_1;Y) - I(X_1;Y|X_2)$, and clearly does not represent the region of interaction in Figure 4.5. For F2, it would be expected that $I(X_1;Y|X_2) < I(X_1;Y)$ due to the interaction between X_1 and X_2. However, a similar result to that observed for F1 can be seen in Table 4.3, and $I(X_1;Y|X_2) > I(X_1;Y)$, even for this function with an interaction between decision variables.

From the results for these two simple test functions, it can be seen that there are practical shortcomings in computing the gene epistasis measure proposed by Seo et al. (2003). Major limitations in calculating the statistic have been identified; either a three dimensional pdf cannot be accurately estimated with a reasonably sized data set, or the series $Y|X_j$ cannot be determined, and does not reflect the interaction between decision variables. If the relationship between variables for these two very simple cases cannot be determined, it is highly unlikely that the measure will be useful for high dimensional optimisation problems. To overcome these problems and allow the degree of interaction between decision variables to be estimated, a new sampling method is proposed, leading to a new statistic, the separability measure.

4.2.2 Separability Measure

While the results from the previous section suggest that the gene epistasis measure may not be suitable to be used in practice, MI may still be a useful property for determining if there is an interaction between pairs of decision variables.

Essentially, it should be possible to detect the relationship between decision variables by sampling the effect of a change in each decision variable on the dependent fitness function value. A sampling method is proposed to produce two series of function values, which can be used to determine the magnitude of the interaction between a pair of decision variables. The sampling method is outlined in the following section, before it is experimentally tested under the same conditions used for the gene epistasis measure above.

Consider a function value based on the value of two variables, $f(x_1, x_2)$. The effect of x_1 on the function value can be determined by comparing the original function value with that obtained when a different value is used for x_1, $f(x_1+\Delta x_1, x_2)$. Δx_1 is not necessarily a small change in x_1, but is used to indicate the function value computed using a different value for x_1, and the same value of x_2. Similarly, the effect of x_2 can be investigated by computing $f(x_1, x_2 + \Delta x_2)$. The effect of both decision variables can be computed

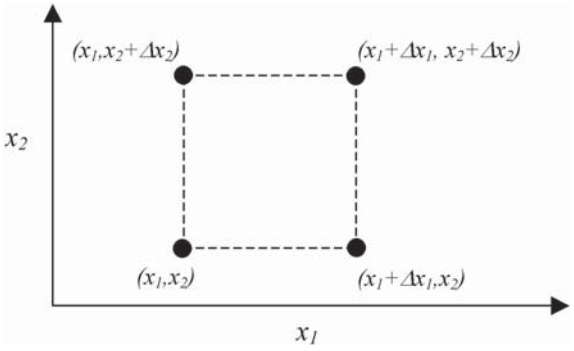


Figure 4.6 Sampling method used for the separability measure.

by changing both variables at once, $f(x_1 + \Delta x_1, x_2 + \Delta x_2)$. If there is no interaction between decision variables, then the sum of the change in each decision variable on their own should equal the change in both decision variables at once:

$$\begin{aligned} f_s(x_1 + \Delta x_1, x_2 + \Delta x_2) &= f(x_1, x_2) + (f(x_1 + \Delta x_1, x_2) - f(x_1, x_2)) \\ &\quad + (f(x_1, x_2 + \Delta x_2) - f(x_1, x_2)), \\ &= f(x_1 + \Delta x_1, x_2) + f(x_1, x_2 + \Delta x_2) - f(x_1, x_2), \end{aligned} \quad (4.11)$$

where f_s is the function value if f is completely separable. This relationship is depicted in Figure 4.6. Next, the value of $f_s(x_1 + \Delta x_1, x_2 + \Delta x_2)$ computed using Equation 4.11 can be compared with the true value of $f(x_1 + \Delta x_1, x_2 + \Delta x_2)$, to determine if there is an interaction between the decision variables. A series of function values can be generated using this approach for different decision variable values, Y_{12}, for the true fitness function values, and $Y_{\mathrm{s},12}$ for the function values if the function is completely separable. If the functions are completely separable, then $Y_{12} = Y_{\mathrm{s},12}$, and $I(Y_{12}; Y_{\mathrm{s},12}) = H(Y_{12}) = H(Y_{\mathrm{s},12})$. Therefore, a measure of the interaction between decision variables can be obtained using the proposed separability measure, λ_{ij}:

$$\lambda_{ij} = 1 - \frac{I(Y_{ij}; Y_{\mathrm{s},ij})}{H(Y_{ij})}. \quad (4.12)$$

If the function is completely separable in variables X_i and X_j, then $I(Y_{ij}; Y_{\mathrm{s},ij}) = H(Y_{ij})$ and $\lambda_{ij} = 0$. Otherwise, $I(Y_{ij}; Y_{\mathrm{s},ij}) < H(Y_{ij})$ and $0 < \lambda_{ij} \leq 1$. $\lambda_{ij} = 1$ occurs when $I(Y_{ij}; Y_{\mathrm{s},ij}) = 0$, i.e., Y_{ij} is completely independent of $Y_{\mathrm{s},ij}$, indicating a very strong interaction between the two variables. The statistic can be applied to problems in more than two dimensions by considering each pair of variables in turn, and holding the other variables at constant values for each sample of Y_{ij} and $Y_{\mathrm{s},ij}$, but using different values for

Table 4.4 Separability measure results.

	F1	F2
λ_{12}	0.054	0.926

each different sample of Y_{ij} and $Y_{\mathrm{s},ij}$, to provide a suitable representation of the search space.

The proposed separability measure has been applied to the two simple functions in Table 4.1, to determine if it can detect the interaction between the variables. As four function evaluations are required to compute one sample, the number of samples has been reduced to $n = 1250$ to ensure the same number of function evaluations are made that were made for the gene epistasis measure experiments.

The results in Table 4.4 indicate that the separability measure can detect the interaction in F2, with a value of $\lambda_{12} = 0.926$, and correctly determines that the random variables are separable for F1, with a very low value of $\lambda_{12} = 0.054$. There is some error introduced into the estimates by the density estimation procedure, as the separability measure for F1 is not exactly zero.

To demonstrate the accuracy of the separability measure on more complex fitness functions than the simple test functions in Table 4.1, the statistic has been applied to the different variations of the test functions considered for F3A in Chapter 3. The 6 variations of the function in $l = 5$ dimensions have been considered, with $m_{\mathrm{BB}} = l/2$ used as $m_{\mathrm{BB}} = 2$ where necessary, and similarly for the value of δ_{BB}. If a value has been computed to be $\lambda < 0$, the value has been presented as $\lambda = 0$ in the results, as these values are only produced by slight errors in the density estimation process.

The results for the separability measure applied to F3A with $m_{\mathrm{BB}} = 1$ and $\delta_{\mathrm{BB}} = 1$ are presented in Table 4.5. The results indicate that the statistic has accurately detected a strong interaction between variables x_2 and x_3 with $\lambda_{2,3} = 0.699$, and correctly determines that there are no other interactions between the variables. Note that the location of the interaction between the variables is randomly determined before the samples are collected, so it could be present between any of the adjacent decision variables. The results for F3A with $m_{\mathrm{BB}} = 1$ and $\delta_{\mathrm{BB}} = 2$, as well as $m_{\mathrm{BB}} = 1$ and $\delta_{\mathrm{BB}} = 4$, are presented in Table 4.6 and Table 4.7, respectively. The separability measure has also correctly classified these functions, with the interaction detected between x_3 and x_5 in Table 4.6, and between x_1 and x_5 in Table 4.7.

Similar results are observed for the remaining variations of F3A that were considered

Table 4.5 Separability measure for F3A with $m_{\text{BB}} = 1$, $\delta_{\text{BB}} = 1$.

	x_2	x_3	x_4	x_5
x_1	0.0	0.0	0.0	0.0
x_2		0.699	0.0	0.0
x_3			0.0	0.0
x_4				0.0

Table 4.6 Separability measure for F3A with $m_{\text{BB}} = 1$, $\delta_{\text{BB}} = 2$.

	x_2	x_3	x_4	x_5
x_1	0.0	0.0	0.0	0.0
x_2		0.0	0.0	0.0
x_3			0.0	0.694
x_4				0.0

Table 4.7 Separability measure for F3A with $m_{\text{BB}} = 1$, $\delta_{\text{BB}} = 4$.

	x_2	x_3	x_4	x_5
x_1	0.0	0.0	0.0	0.697
x_2		0.0	0.0	0.0
x_3			0.0	0.0
x_4				0.0

Table 4.8 Separability measure for F3A with $m_{\text{BB}} = 2$, $\delta_{\text{BB}} = 1$.

	x_2	x_3	x_4	x_5
x_1	0.0	0.0	0.0	0.0
x_2		0.699	0.0	0.0
x_3			0.699	0.0
x_4				0.0

in Chapter 3, with interactions between x_2 and x_3, as well as between x_3 and x_4, for F3A with $m_{\text{BB}} = 2$ and $\delta_{\text{BB}} = 1$ in Table 4.8, interactions between x_1 and x_3, as well as between x_2 and x_4, for F3A with $m_{\text{BB}} = 2$ and $\delta_{\text{BB}} = 2$ in Table 4.9, and between all adjacent variables in Table 4.10 for $m_{\text{BB}} = 4$ and $\delta_{\text{BB}} = 1$. Therefore, the results presented in this section indicate that the proposed seperability measure can accurately determine the location of any epistatic interactions for these more complex functions.

As the proposed statistic is based on the distribution of data produced by the difference from changes to pairs of decision variables, it can be accurately applied to any fitness function. Higher order interactions can be identified by inspecting the results for pair-wise interactions. For example, if an interaction is detected between variables x_i and x_j, x_k

Table 4.9 Separability measure for F3A with $m_{\text{BB}} = 2$, $\delta_{\text{BB}} = 2$.

	x_2	x_3	x_4	x_5
x_1	0.0	0.699	0.0	0.0
x_2		0.0	0.694	0.0
x_3			0.0	0.0
x_4				0.0

Table 4.10 Separability measure for F3A with $m_{\text{BB}} = 4$, $\delta_{\text{BB}} = 1$.

	x_2	x_3	x_4	x_5
x_1	0.729	0.0	0.0	0.0
x_2		0.732	0.0	0.0
x_3			0.739	0.0
x_4				0.718

and x_j, as well as x_i and x_k, it would be expected that there is an interaction of order $k_{\text{BB}} = 3$ between these three variables. By only considering pairs of variables at a time, it allows very accurate results to be produced, as it is unlikely that for higher dimensional problems the effect of one decision variable on the fitness function value will be able to be detected over the combined effect of all other decision variables from a random sample of data. However, this can also be a disadvantage, as it dramatically increases the computational requirements to compute the statistic, as the fitness function must be sampled for each combination of pairs of decision variables. In higher dimensions, the number of combinations of decision variables increases quickly, as the number of combinations of decision variables to be sampled is $l(l-1)/2$.

4.3 Decision Variable Salience

So far in this chapter, statistics have been developed to provide information about two of the function characteristics that were considered in Chapter 3, the roughness and multimodality of the function, as well as the interactions between variables. The remaining characteristic that must be quantified is the salience of the decision variables, where a highly salient variable has a much larger effect on the value of the fitness function than the others. The results presented in Section 4.2.2 indicate that MI can be a useful statistic for determining the relationship between variables if the sample data are collected carefully. The major difference between the methodology for the proposed separability measure and the gene epistasis measure of Seo et al. (2003) is that only two series of data need to be considered, as opposed to three, and due to the curse of dimensionality, it becomes too difficult to estimate the necessary properties in three dimensions.

It is proposed in this section that MI could also be used to determine if there are any salient variables in the search space. The normalised MI, NI, is proposed for this purpose:

$$NI(X_i, Y) = \frac{I(X_i, Y)}{H(Y)}. \tag{4.13}$$

The value for the statistic ranges from $0 \leq NI \leq 1$. If $I(X_i, Y) = H(Y)$ then $NI(X_i, Y) = 1$, and X_i contains the same information as Y, so variable X_i completely describes Y and any other decision variables have no effect on the fitness function value. On the other hand, if $I(X_i, Y) = 0$, then X_i has no relationship to Y. Therefore, by comparing the values of $NI(X_i, Y)$ for all decision variables, X_i, any salient variables that have a much larger effect on the fitness function value can be detected.

To investigate if the normalised MI can detect the different contribution of decision

Table 4.11 Salience results for F3C.

	Case			
	1	2	3	4
x_1	0.001	0.045	0.008	0.006
x_2	0.004	0.050	0.051	0.012
x_3	0.008	0.006	0.008	0.020
x_4	0.007	0.004	0.000	0.016
x_5	0.009	0.000	0.003	0.011

variables on the fitness function values, the statistic has been applied to the variations of F3C considered in Chapter 3. The functions have been tested in $l = 5$ dimensions, and $n = 5000$ has been used to compute the statistic in each case. The results presented have been taken from an average of 30 random data sets.

Table 4.11 presents the results for the normalised MI between each decision variable and the fitness function value for the four cases of F3C considered in Chapter 3. Cases 1 and 4 have similar contributions from each decision variable, with the only difference being that each sub-function of the Rastrigin Function has been squared for Case 4. From Table 4.11, it can be seen that the normalised MI between each variable and the fitness function is very similar for each case, correctly indicating that each variable has a similar contribution to the fitness function value. Case 2 of function F3C has the first sub-function of the Rastrigin Function squared, so it would be expected that the term $(x_1 + x_2)/2$ would have a much larger effect on the fitness function value than the remaining decision variables. This is reflected in Table 4.11, with $NI(X_1, Y) = 0.045$ and $NI(X_2, Y) = 0.05$, an order of magnitude higher than the next highest value of NI. The value for $NI(X_2, Y)$ is slightly higher, because it will also contribute to the fitness function value in the $(x_2 + x_3)/2$ term. For Case 3 the first two terms of the fitness function are squared, as opposed to only the first term for Case 2. The normalised MI values indicate that x_2 has by far the largest contribution to the fitness function value, with $NI(X_2, Y) = 0.051$. This can be explained by the fact that x_2 is involved in both squared terms for Case 3 of F3C, $(x_1 + x_2)/2$ and $(x_2 + x_3)/2$, and therefore has a much larger contribution to the fitness function value than x_1 or x_3. The results also correctly indicate that x_1 and x_3 have a larger effect on the fitness function value for Case 3 than x_4 and x_5.

The results presented in Table 4.11 correctly reflect the relative relationship between each decision variable and the fitness function value for the cases of F3C considered. However, the values of normalised MI are very low for all cases. This can be explained

Table 4.12 Salience results for F3C without epistatic interactions.

	Case			
	1	2	3	4
x_1	0.044	0.670	0.155	0.047
x_2	0.048	0.003	0.145	0.051
x_3	0.045	0.004	0.001	0.045
x_4	0.041	0.000	0.000	0.050
x_5	0.044	0.004	0.000	0.053

by the pair-wise interactions that were introduced between the decision variables, as $NI(X_i, Y)$ does not provide any information about changes to x_{i+1}, which will also be affecting the values of y for changes in x_i. To demonstrate this phenomenon, the normalised MI statistic has been applied to the four cases of F3A without the pair-wise interaction between the decision variables. The results from this analysis are presented in Table 4.12, where the relationship between the decision variables and the fitness function value are much clearer. For Cases 1 and 4, $NI(X_i, Y) = 0.045$ for all decision variables, indicating that each variable has a similar effect on the fitness function value. For Case 2, x_1 has by far the largest effect on the fitness function values with $NI(X_1, Y) = 0.67$. For Case 3, both x_1 and x_2 dominate the fitness function with $NI(X_1, Y) \approx NI(X_2, Y) \approx 0.15$, as expected.

The results presented in Table 4.11 and Table 4.12 indicate that the normalised MI can be used to determine if any highly salient variables exist in the fitness function value. As the effect of only one decision variable at a time is considered in the normalised MI statistic, the combination of two variables on the fitness function value was not detected as well, as indicated by the lower values calculated in Table 4.11 compared to Table 4.12. However, the results indicate that differences between the normalised MI values computed between each of the decision variables and the fitness function value can be used to provide very useful information about the salience of the variables for a given fitness function.

4.4 SUMMARY

Chapter 3 identified a number of fitness function characteristics that affected the number of generations before a GA population converged when solving a given fitness function. This chapter has focused on developing statistics to quantify the characteristics that were

identified. The first characteristic considered in this chapter was the roughness and multimodality of a fitness function. The spatial correlation measure was proposed to quantify these characteristics, and was found to compare very favourably to the true autocorrelation for functions with a known autocorrelation. Not only was the spatial correlation statistic accurate, but it was also able to identify changes in the roughness, and to a lesser extent multimodality, which were observed to affect the value of g_{conv} in Chapter 3.

After testing the gene epistasis measure, the separability measure was proposed to identify epistatic interactions between decision variables. By testing the statistic on functions with different numbers of interactions, and distance between the interactions, it was found that the measure was able to very accurately determine the location of any epistatic interactions introduced into the fitness function. The proposed measure worked by identifying interactions between pairs of variables, and by inspecting all pairs of interactions, higher order interactions can also be identified.

The final statistic proposed in this chapter was the normalised MI to determine the presence of any highly salient decision variables in the fitness function. The results presented in Section 4.3 indicate that the normalised MI is a suitable method for comparing the effect of each decision variable on the fitness function value, as the results agreed with the expected results for the functions with known characteristics considered.

Therefore, this chapter has presented statistics that can be used to quantify the characteristics of fitness functions that were identified to effect the most efficient number of GA generations in Chapter 3. The following chapter is dedicated to developing a relationship between g_{conv} and the values computed by the three statistics proposed in this chapter, to allow g_{conv} to be predicted for functions with unknown characteristics.

Chapter 5
Predicting the Number of Generations Before Convergence

In Chapter 3, it was found that for a given fitness function with certain characteristics, the number of generations before the population converged, g_{conv}, did not change, irrespective of the convergence criteria. It was also observed that the value of g_{conv} changed with changes in the characteristics of the fitness function. A number of fitness function statistics to provide useful information about these characteristics were developed in Chapter 4. However, Merz (2004) recognised that while fitness function measures may provide some useful information about problem characteristics, the link between GA theory and function properties has not yet been well developed. The aim of this chapter is to progress this understanding, by developing a relationship between the values calculated by the fitness function statistics and g_{conv} for a range of test functions. In the following section, the theoretical basis for the proposed analyses is outlined, before the methodology used to derive an empirical relationship is presented. This is followed by the results obtained using the derived relationship, and a validation of the relationship by comparing the predicted and observed values of g_{conv} for a range of test functions.

5.1 Background

In this chapter, it is proposed that the fitness function characteristics can be used to predict when a GA population will converge. This hypothesis appears reasonable, as from experience we know that the best GA parameters are different, depending on the characteristics of the fitness function. However, the proposed hypothesis also has a more theoretical basis, derived from quantitative genetic theory.

A number of GA studies (Lobo, 2000; Mühlenbein and Schlierkamp-Voosen, 1993; Rogers and Pruegel-Bennett, 1999; Thierens and Goldberg, 1994; Thierens et al., 1998)

have made use of the Genetical Theory of Natural Selection (Fisher, 1930), which, in GA terminology, states that the change in the average fitness function value of the population from the parents to the children solutions produced by the selection operator is equal to the selection pressure multiplied by the standard deviation of the fitness function values. The rather intuitive result of this relationship is that there will be no more improvement in the fitness function values when there is no more diversity in the population.

The selection operator will decrease the variance in the fitness function values, by replacing the worst solutions by better ones. As outlined in Section 2.3.3, Quantitative genetic theory indicates that this decrease will be a constant value, k, each generation (Falconer, 1981). This results in an exponential decay of the variance in the fitness function values in terms of the number of generations. Therefore, the time until the GA converges to one solution is dependent on the rate of decay of the population standard deviation, and can be computed by:

$$\begin{aligned}
\sigma_{\text{P,g}} &= \sigma_{\text{P,0}} k^g \\
\frac{\sigma_{\text{P,g}}}{\sigma_{\text{P,0}}} &= k^g \\
\log\left(\frac{\sigma_{\text{P,g}}}{\sigma_{\text{P,0}}}\right) &= g\log(k) \\
g\log(k) &= \log(\sigma_{\text{P,g}}) - \log(\sigma_{\text{P,0}}) \\
g &= \frac{\log(\sigma_{\text{P,g}}) - \log(\sigma_{\text{P,0}})}{\log(k)}
\end{aligned} \tag{5.1}$$

where $\sigma_{\text{P,g}}$ is the standard deviation of the fitness function values after generation g, $\sigma_{\text{P,0}}$ is the standard deviation of the fitness function values of the initial population, and k is a constant with value $0 \leq k \leq 1$. Therefore, if an estimate can be obtained for the value of k, then an estimate can be obtained for the number of generations before convergence occurs, g_{conv}.

Quantitative genetic theory also provides an insight in the components that make up the variance in the fitness values. Falconer (1981) cites a number of components that comprise the total fitness variance, namely:

- Genotypic variance, (σ_G^2);
- Breeding or Additive variance (σ_A^2);
- Dominance variance (σ_D^2);
- Interaction variance (σ_I^2); and
- Environmental variance (σ_E^2).

The total variance in the fitness function values is the sum of these components, in the form (Falconer, 1981):

$$\begin{aligned}\sigma_{\mathrm{P}}^2 &= \sigma_G^2 + \sigma_E^2 \\ &= \sigma_A^2 + \sigma_D^2 + \sigma_I^2 + \sigma_E^2. \end{aligned} \tag{5.2}$$

These components of variance are analogous to characteristics of fitness functions that can now be quantified by the fitness function statistics that were proposed in Chapter 4. σ_A^2 can be considered to be variance in the decision variable values. σ_D^2 is the variance provided by dominant, or highly salient, variables; which can be predicted using the normalised MI statistic. σ_I^2 is the variance produced by epistatic interactions between the decision variables, as estimated by the separability measure, λ. σ_E^2 is the variance produced by the environment, which can be interpreted as the shape of the fitness function, and useful information about this aspect of the variance of the fitness function can be provided by the spatial correlation measure, R_{s}.

Based on these observations, it is proposed in this chapter that the fitness function statistics can be used along with the standard deviation of the values of the decision variables in the population, σ_{pop}, to predict the change in the standard deviation of fitness function values due to selection, k. From the predicted value of k, the number of generations before the variance in the fitness function values decreases to zero can then also be predicted. Therefore, the following section is dedicated to quantifying the relationship between the information provided by the fitness function measures and the change in standard deviation due to selection, k.

5.2 The Relationship Between Fitness Function Characteristics and Population Variance

An empirical approach has been adopted to develop a relationship between k and the information provided by the fitness function statistics. Firstly, the methodology adopted is outlined, including the test functions selected, and how the information provided by the fitness function statistics has been used. This is followed by a results section, outlining the input determination process, the derivation of the functional form of the relationship, and finally the relationship that has been obtained.

Table 5.1 Parameters values used for the test functions.

Parameter	Values
l	2, 4, 10, 30, 50, 80, 100
A_i	0, 0.5, 1, 2
f_1	1
f_2	2, 4, 8, 16
f_3	3, 5, 9
p	0, 1

5.2.1 Methodology

5.2.1.1 Test Functions

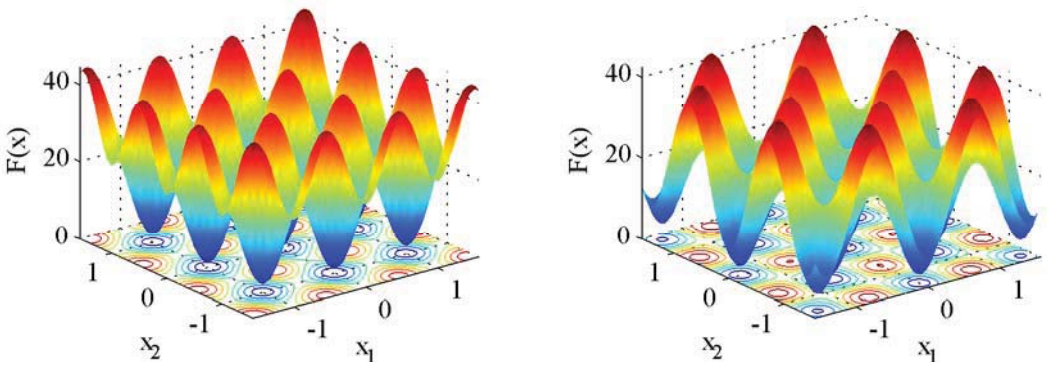
Functions of the form of a Fourier Series have been selected as the test functions. These functions have been selected as a wide range of characteristics can be generated, and complex functions can be obtained by the superposition of a number of terms with different amplitudes and frequencies.

Up to three cosine terms have been used to generate the test functions. The functional form of the test functions is given in Equation 5.3:

$$F(\mathbf{x}) = \sum_{i=1}^{3} \sum_{j=1}^{l} j^p A_i \cos(f_i x_j), \tag{5.3}$$

where A_i and f_i are the amplitude and frequency of each term i. The range considered for the decision variables was $x_i \in [0, 2\pi]$. The first cosine term has been used to control the global structure of the fitness function. To do so, each of the amplitudes shown in Table 5.1 has been implemented, along with a frequency of $f_1 = 1$. This allows the global structure to change from a 'big bowl' with $A_1 = 2$, to totally flat, with $A_1 = 0$. The remaining two terms for each test function have been used to control the degree of roughness and multimodality. The amplitude and frequency values considered for terms two and three are given in Table 5.1. It can be seen from Table 5.1 that terms two and three can have similar frequencies, to produce complex function characteristics, where the cosine terms are additive for some of the search space, and negating each other in other areas of the search space. The p parameter in Equation 5.3 has been included to allow the salience of the decision variables to be controlled, with values of $p = 0$ and 1 considered.

Along with the original form of the test function without epistatic interactions between decision variables, interaction cases of $m_{\text{BB}} = 1, \delta_{\text{BB}} = l - 1$, $m_{\text{BB}} = 2, \delta_{\text{BB}} = l/2$, and $m_{\text{BB}} = l/2, \delta_{\text{BB}} = 2$ have been considered. Interactions between decision variables were



(a) No rotation (b) With rotation

Figure 5.1 The effect of applying the rotation matrix.

produced by rotating each of the terms of the test function in Equation 5.3 using the transformation matrix, M, proposed by Salomon (1996). The transformation matrix is a pure rotation, and does not change the function's structure (Salomon, 1996). Any of the decision variables that have the transformation matrix applied are no longer separable, and therefore an epistatic interaction is produced between the variables. The rotation matrix is composed to produce a $45\,^{\circ}$ rotation of the variables. If there is to be an epistatic interaction between a number of variables, then their values are transformed using the matrix such that, $\mathbf{x}_{\mathrm{T}} = \mathbf{x}M$, where $\mathbf{x}$ contains the values of the variables to be transformed. The effect of applying M can be seen in Figure 5.1, where the original function can be seen in Figure 5.1(a), and the same function with the rotation applied can be seen in Figure 5.1(b). The rotation seen in Figure 5.1(b) was produced using Equation 5.4:

$$\begin{aligned}\begin{bmatrix} x_1 & x_2 \end{bmatrix}_T &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} \sin(45^\circ) & \cos(45^\circ) \\ \cos(45^\circ) & -\sin(45^\circ) \end{bmatrix} \\ &= \begin{bmatrix} x_1\sin(45^\circ) + x_2\cos(45^\circ) & x_1\cos(45^\circ) - x_2\sin(45^\circ) \end{bmatrix}. \end{aligned} \tag{5.4}$$

The interaction between variables produced by the rotation matrix can be seen very clearly in Equation 5.4. The effect of the rotation can be seen in Figure 5.1(b), where the local optima no longer line up along the same value of a decision variable, which is the case in Figure 5.1(a). For example, a number of local maxima occur for $x_1 = 0$ in Figure 5.1(b), which also contains the global minimum. However, in Figure 5.1(a) without the transformation matrix applied, only local minima occur along the line of $x_1 = 0$, and therefore it is much easier for the GA to solve the problem.

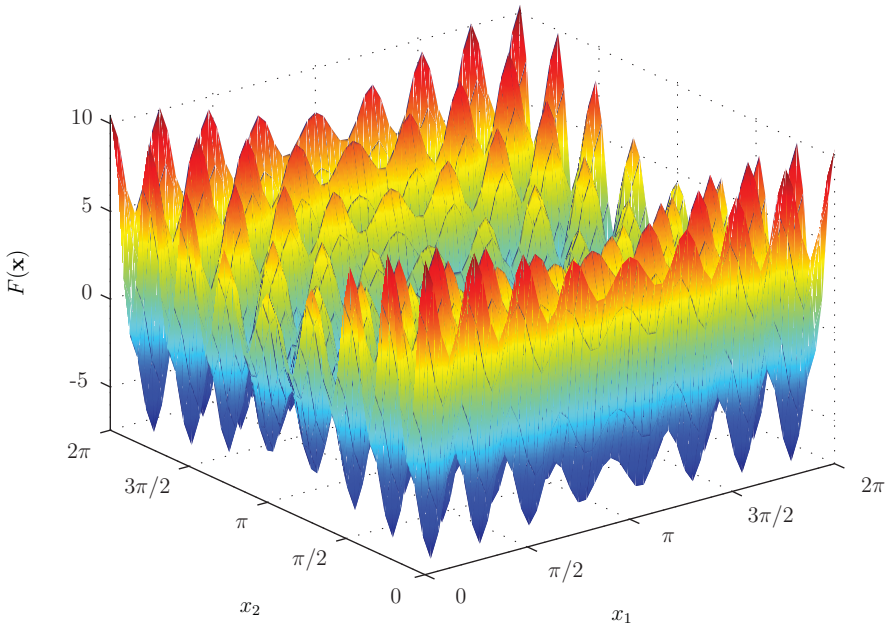


Figure 5.2 Example of the Fourier Series test functions used.

An example test function is shown in Figure 5.2 which has been produced using parameter values: $l = 2$, $A_1 = 0.5$, $A_2 = 2$, $A_3 = 1$, $f_1 = 1$, $f_2 = 8$, $f_3 = 9$, and $p = 1$. Every combination of the function parameters considered in Table 5.1 has been tested, along with each of the different interaction cases, however, combinations that produce the same function characteristics have not been re-evaluated. Examples of parameter combinations that produce the same function characteristics include changes to the frequency of a term when $A_i = 0$, different interaction cases that are the same for the lower dimensional functions considered, or functions with $A_i = 1$ or $A_i = 2$ and the same values for f_i, producing the same function characteristics, only the function values are twice the magnitude for $A_i = 2$. Therefore, a total of $18\,672$ functions were used to collect the data used determine the relationship between fitness function characteristics and k.

5.2.1.2 Determining the Decay in Population Variance

The decay in population variance has been computed for each of the test functions outlined above. A total of $n = 5\,000$ samples in the search space have been used to represent a population of solutions. The population standard deviation, σ_{pop}, has been computed based on the population variance statistic proposed by Beyer and Deb (2001):

$$\sigma_{\text{pop}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{l}(x_{s,i,j} - \overline{x}_{s,j})^2}, \tag{5.5}$$

where x_s are sampled decision variable values, standardised to into the range $[0, 1]$. This standardisation is used to ensure that one decision variable that happens to occur over a larger range does not dominate the σ_{pop} computed. Initially, the values are randomly sampled from a uniform distribution over the search space, to represent the initial parent solutions in a GA population. Tournament selection is then applied to the sampled solutions, producing the children solutions. The variance in the population will decrease after the selection operator is applied, as after selection the worst solutions in the population are replaced by a number of copies of the better solutions in the population. σ_{pop} is then recomputed from the decision variable values of the children solutions using Equation 5.5. The decay in the standard deviation of the population due to selection, k, is then computed by:

$$k = \frac{\sigma_{\text{pop,children}}}{\sigma_{\text{pop,parents}}}. \tag{5.6}$$

To remove any variability produced by the random sampling, the average over 30 different random data sets have been taken as the k value for each test function to develop the relationship in Section 5.2.2.

5.2.1.3 Statistics Based on the Fitness Function Measures

Each of the measures proposed in Chapter 4 have been applied to each of the test functions outlined in Section 5.2.1.1. In order to make use of the correlation plots that are produced by the spatial correlation measure, a number of statistics have been computed. The three statistics are the correlation length, R_{l}, the average correlation, R_{av}, and the total positive correlation in the fitness function, R_{T}.

The correlation length has been used in a number of studies to investigate the behaviour of GAs. Along with the correlation length, however, two more statistics that provide information about the roughness of a fitness function have been developed. The first of these statistics is the average correlation in fitness function over the correlation length, which provides information about the roughness of the fitness function for solutions near each other in the search space. It was observed in Chapter 4 that an increase in the roughness of the fitness function did not affect the correlation length, however, the the value for R_{av} did decrease with an increase in roughness, indicating that this is a very useful statistic for the characterisation of fitness functions. The second statistic is the total positive correlation in the search space, R_{T}. This statistic provides a very useful insight into the global structure in the search space. For example, if $R_{\text{T}} > R_{\text{av}}$, it is likely that points far apart in the search space are positively correlated, indicating that a 'big bowl' structure exists for the fitness function being analysed. However, if $R_{\text{T}} = R_{\text{av}}$, then the

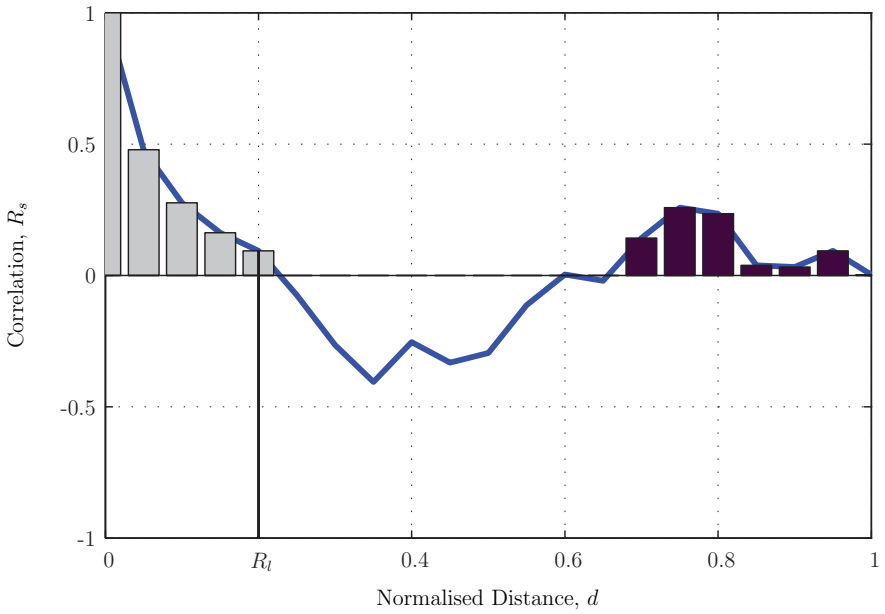


Figure 5.3 The computation of the correlation statistics from the spatial correlation measure.

plot of correlation against distance in the search space does not turn positive again after the correlation length, indicating that the function is likely to be globally flat over the whole search space.

The spatial correlation function computed for the Rastrigin Function with pair-wise interaction between decision variables in Section 4.1 can be seen in Figure 5.3. The value of R_{l} can be seen as the normalised distance before the spatial correlation calculated is no longer positive, within the limits of statistical significance. The limit of statistical significance has been taken as $\frac{1}{\sqrt{n}}$ in this work. Therefore, for the function considered in Figure 5.3, $R_{\text{l}} = 0.2$. The value for R_{av} is computed as the average area under the correlation plot for distances less than the correlation length, as indicated by the grey bars in Figure 5.3. The value for R_{av} for the function considered in Figure 5.3 is therefore $R_{\text{av}} = 0.10$, taken as the sum of the height of each bar multiplied by its width, in this case 0.05. The total positive correlation in the fitness function, R_{T}, is computed in a similar fashion, however, in this case, the area under the whole positive section of the correlation plot is used, as indicated by both the grey and black bars. Therefore, the value for R_{T} in Figure 5.3 is $R_{\text{T}} = 0.14$.

To make use of the salience measure proposed in Section 4.3, the difference between the maximum NI and minimum NI has been used. This statistic provides information about the maximum dominance of one variable over another for a given fitness function.

The equation used to compute the dominance statistic, D, is given in Equation 5.7:

$$D = \max_{i \in 1...l} NI(X_i, Y) - \min_{j \in 1...l} NI(X_j, Y). \tag{5.7}$$

While the number and order of any interactions for the test functions is known from the variables that have been rotated by the transformation matrix, the separability measure has been applied to ensure that the interactions can be detected in practice. Based on the array of interactions produced by applying the separability measure to each pair of decision variables, higher order interactions that are present in some of the test functions have been located. Higher order interaction were identified by inspecting the results for the pair-wise interactions. For example, if an interaction is detected between variables x_i and x_j, x_k and x_j, as well as x_i and x_k, then there is an interaction of order $k_{\text{BB}} = 3$ between these three variables. This analysis has been extended to identify the highest order of interaction present in each of the test functions. Therefore, k_{BB} has been taken as the size of the biggest building block identified, and m_{BB} has been taken as the number of unique building blocks in the fitness function.

5.2.2 Relationship Results

Before the relationship between the fitness function measures and the k values can be developed, the data collected from the test functions have been screened to include only the functions that are Optimal Generation Functions. The flowchart given in Section 3.1.3 has been used to undertake this screening. If the correlation length for a test function was computed to have $R_{\text{l}} \leq 0.05$, or if $R_{\text{av}} < 0.05$, then the function has been deemed to have no structure in the fitness function, and is therefore not an Optimal Generation Function. Typically, this occurred for functions that were comprised of terms with the highest frequencies considered along with the highest amplitudes, or if $A_1 = 0$. Similarly, functions that had both $k_{\text{BB}} = 1$, indicating no epistatic interactions, and $D < 0.05$, indicating that the function has no highly salient variables, were also deemed to not be Optimal Generation Functions. Therefore, of the original $18\,672$ test functions considered, $9\,347$ functions were determined to be Optimal Generation Functions to be used to develop the relationship.

5.2.2.1 Input Determination

In order to identify the statistics that had the largest influence on the k value, an input determination technique has been implemented. The Partial Mutual Information (PMI) (Sharma, 2000) between the values of k, or the output, and each of the statistic values, or

inputs, has been used as the input determination technique. The dimension of the problem has also been included as an input. The process of PMI is similar to the method used to estimate the gene epistasis by fitness function residuals in Section 4.2. Firstly, the MI is calculated between each input, the statistic values, and the output (i.e.; the values of k). The input that has the highest value of MI is regressed against the output, using the non-parametric kernel regression outlined in Section 4.2. The resulting series comprises the residual values of k, the values of k without the influence of the most significant input. The process is then repeated, in this case by computing the MI between each of the remaining inputs and the output, in the form of the residuals of the k values, known as the PMI. The input that produces the highest PMI value is then regressed against the residuals of the k values. The process is continued until there are no more significant inputs.

For this work the significant inputs have been determined using 95% confidence level, indicated in brackets Table 5.2. The 95% confidence levels have been produced from 1000 bootstraps of the data. This involved computing the MI between a random reordering of the data, which will remove any true interaction between the parameters. This process was repeated 1000 times, and the MI value that was greater than the MI computed for a random reordering between the input and output values 95% of the time is selected as the significance level.

The results from the input determination procedure using the data for the $9\,347$ test functions is presented in Table 5.2. The analysis has been performed with reference bandwidth and a Gaussian kernel. A base 2 logarithm has been used to compute the entropy, therefore the units of the PMI values is 'bits'. From Table 5.2, it can be seen that all the statistics computed have a strong relationship with k, apart from those produced by the separability measure, namely m_{BB} and k_{BB}. This result reinforces the observation in Chapter 3, where the number or distance between the epistatic interaction had very little effect on the g_{conv} value. Based on the input selection results, the statistics computed from the separability measure have not been included in the relationship between k and the remaining fitness function statistics. While it has not been included in the prediction of the value of k, the information provided by the separability measure may still be useful in re-arranging the solution string, to align the interactions between decision variables, giving the GA the best chance to find better solutions before the population will converge.

5.2.2.2 Functional Form of the Relationship

The information provided by the fitness function statistics is very similar to the parameters involved in the theoretical population sizing equation proposed by Harik et al. (1999).

Table 5.2 Input selection results.

Rank	Parameter	$PI(X_i, Y)$	
1	R_{av}	0.899	(0.177)
2	R_{T}	0.825	(0.113)
3	l	0.384	(0.101)
4	R_{l}	0.288	(0.083)
5	D	0.212	(0.089)
6	m_{BB}	0.076	(0.079)
7	k_{BB}	0.077	(0.086)

Therefore, a similar functional form has been adopted to that given in Equation 2.11, where each input is raised to a power, and the product of each input term is taken. The population sizing equation proposed by Harik et al. (1999) was focused on maximising the final solution quality found by the GA, and hence terms similar to m_{BB} and k_{BB} were included in the equation. However, as discussed above, these parameters were not found to influence the convergence of the algorithm considered in this work.

The relationships proposed by Thierens and Goldberg (1994) and Thierens et al. (1998) have also been included in the functional form of the relationship used in this work. Their results suggest that g_{conv} is between $O(\sqrt{l})$ and $O(l)$, depending on the salience of the decision variables. Therefore, the l and D parameters have been included in the form $l^{(\alpha+\beta D)}$.

Hence, the general form of the function that has been selected to predict the decay in population variance due to selection is:

$$k = 1 - aR_{\text{l}}^{b} R_{\text{T}}^{c} R_{\text{av}}^{d} l^{(e+fD)}, \tag{5.8}$$

where: k is the decay of population variance, R_{l} is the correlation length, R_{T} is the total correlation, R_{av} is the average correlation, l is the problem size, D is the dominance statistic, and a, b, c, d, e and f are all parameters to be calibrated to the data collected from the test functions. a has been constrained such that $a > 0$, and will not produced a value greater than 1 for the second term in Equation 5.8. This constraint has been used to ensure that the resulting prediction of k is reasonable, i.e.; $k \in [0, 1]$.

A non-linear generalised reduced gradient search method has been used to determine the parameter values in the relationship, as it would be expected that the search space characteristics are smooth, with few local optima, and gradient methods are more suitable for these purposes. The initial starting points for the gradient search have been determined

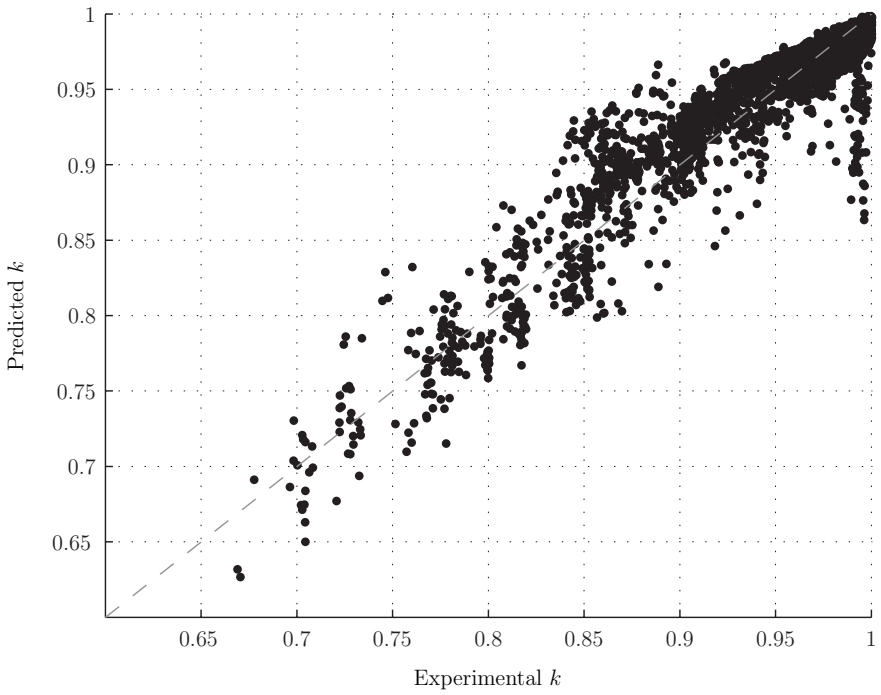


Figure 5.4 Predicted and experimental values of k for the Fourier Series test functions.

from the regression coefficients between each of the inputs and the output, and a number of different starting points were used with the search method in order to have the best chance of determining the globally optimal set of coefficients.

Based on these analyses, the following relationship has been fitted to the data collected from the test functions:

$$k = 1 - 2.266 \frac{R_{\text{l}}^{1.168} R_{\text{T}}^{2.145}}{R_{\text{av}}^{1.074} l^{(0.506+0.346D)}}. \tag{5.9}$$

The k value experimentally computed from a sample of solutions from each test function plotted against the k value predicted by Equation 5.9 can be seen in Figure 5.4. The dashed grey line indicates the line of perfect prediction. The figure shows that very good agreement is achieved between the functional form used in Equation 5.9 and the experimentally collected data. A coefficient of determination of $R^2 = 0.8118$ is obtained for this relationship. All of the coefficients of determination presented in this section are produced from the regression line through the origin, to ensure there is no bias included in the goodness of fit tests.

5.3 VALIDATION OF THE RELATIONSHIP

An empirical relationship between the information provided by the fitness function statistics proposed in Chapter 4 and the decay in variance in a GA population due to selection has been developed in the first part of this chapter. The remainder of this chapter is dedicated to testing the generality of the relationship identified. Firstly, the relationship is applied to test functions that have a different functional form to the Fourier Series that have been used to develop the relationship. Secondly, the predicted value of k is used to determine an estimate for g_{conv}, which is then compared against the experimentally observed value of g_{conv} for a number of test functions.

In order to undertake these tests, the variations of the three Optimal Generation Functions considered in Chapter 3 are used. These function have been selected as the very large GA parametric studies required to obtain a experimental estimate of g_{conv} were undertaken in Chapter 3. The following section compares the predicted and experimental value of k for these functions, before the predicted value of k is used to determine an estimate for g_{conv}, to be compared against the values that were experimentally observed in Chapter 3.

5.3.1 Predicting the Decay in Population Variance

A similar approach to that used to determine the experimental value of k for the Fourier Series functions in Section 5.2.1.1 has been used to determine the value of k for the test functions considered in Chapter 3. A sample size of $n = 5\,000$ in the search space has been used to represent a population of solutions to compute the population standard deviation, σ_{pop}, before and after selection. Each decision variable has been standardised to be in the range $[0, 1]$. Again, the average over 30 different random sample data sets have been used to compute the k values presented.

In the remainder of this section, the predicted value of k using Equation 5.9 is plotted against the experimentally observed value for each of the three families of Optimal Generation Functions considered in Section 3.2.

5.3.1.1 F3

F3 considered variations of the Rastrigin Function, with different degrees of epistatic interactions (F3A), roughness and multimodality (F3B) and salience of the variables (F3C). The predicted and experimental values of k for all variations of the Rastrigin Function considered are shown in Figure 5.5. It can be seen from Figure 5.5 that generally Equation 5.9 produces a slight overestimate of the experimental values, however, a very good

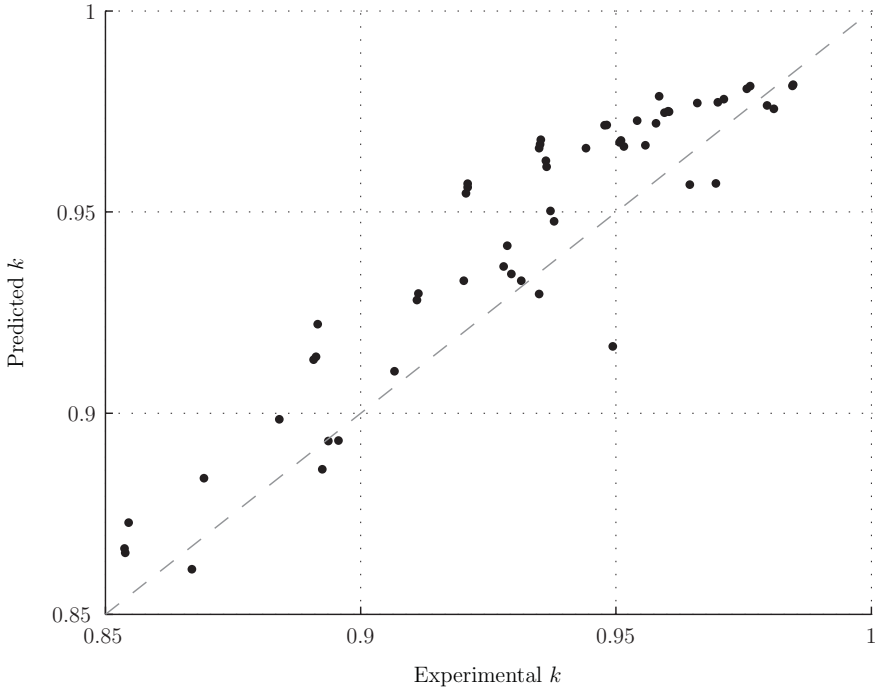


Figure 5.5 Predicted and experimental values of k for F3.

fit is obtained for these functions, with a coefficient of determination of $R^2 = 0.848$.

5.3.1.2 F4

Variations of Griwank's Function (F4) were considered in Section 3.2. The values predicted and experimental k values for both F4A and F4B are shown in Figure 5.6. From Figure 5.6 it can be seen that the k values produced by Equation 5.9 consistently produced a slight overestimation of the experimental values, with every point on Figure 5.6 above the line of perfect prediction. However, the overestimation is only slight, and a very strong relationship with $R^2 = 0.920$ is obtained.

5.3.1.3 F6

Function F6 was composed by a summation of the decision variable values multiplied by their location in the solution string, with the final decision variable highly salient, as it was raised to a higher power. The results from the comparison of the predicted and experimental values of k for F6 can be seen in Figure 5.7. While there is generally good agreement between the predicted and experimental values of k, the relationship is not as close for F6 compared to that observed for F3 and F4. This is reinforced by a lower coefficient of determination of $R^2 = 0.686$. The effect of problem size on the predicted

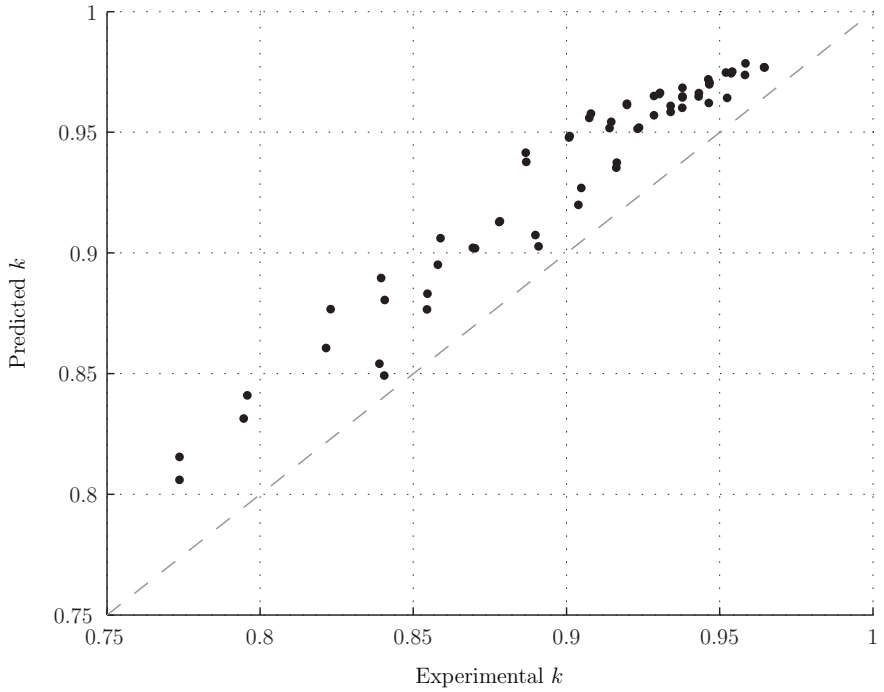


Figure 5.6 Predicted and experimental values of k for F4.

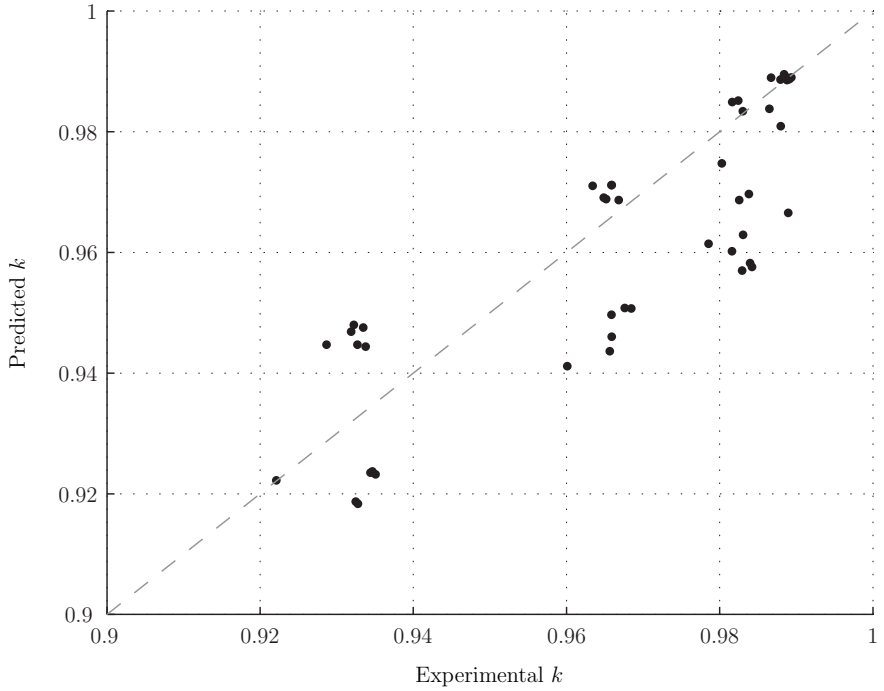


Figure 5.7 Predicted and experimental values of k for F6.

values of k is more evident in the predictions for F6, with each cluster of points seen on Figure 5.7 corresponding to each problem size tested, $l = 5$, 10, 20, and 30.

5.3.2 Predicting the Number of Generations Before Convergence

The previous section demonstrated that the relationship proposed in Equation 5.9 can generalise to functional forms other than the Fourier Series functions that it was developed on, with good predictions of the experimentally observed values of k for the test functions considered. The aim of this section is to take the predicted values of k to obtain the more useful prediction of g_{conv}, and compare the predicted values against those observed experimentally.

The functions considered in Section 3.2 have again been used as the test functions, as the experimental values of g_{conv} have been determined in that section. To determine each value of g_{conv} experimentally, large parametric studies were undertaken, with a total of $12\,960$ GA runs required to determine g_{conv} for each function, represented by a single dot on each of the figures in the following sections.

5.3.2.1 Determining the Initial and Final Population Variance

Equation 5.1 indicates that the number of generations before the GA population will converge to a single value can be determined from the k value, provided two terms are quantified, the initial and final population standard deviation. It would be expected that g_{conv} occurs when there is no more variance in the population, occurring when $\sigma_{\text{pop,g}} = 0$. However, in practice, the term $\log(0)$ cannot be computed, and an approximation to signify when the population has adequately converged must be adopted. In this work, the GA has been deemed to have converged once $\sigma_{\text{pop,g}} = 0.001$.

The other parameter that must be estimated in Equation 5.1 is the initial standard deviation of the population, $\sigma_{\text{pop,0}}$. As each variable is scaled to the range $[0, 1]$ to compute the standard deviation, and the samples have been drawn from a uniform random distribution, the expected value of the variance for one decision variable is $\sigma^2 = \frac{(1-0)^2}{12} = \frac{1}{12}$. The approach used to compute the population variance, given in Equation 5.5, involves summing the variance over each decision variable and hence the initial standard deviation in the population, $\sigma_{\text{pop,0}}$, is given by:

$$\sigma_{\text{pop,0}} = \sqrt{\frac{l}{12}}. \tag{5.10}$$

By substituting $\sigma_{\text{pop,g}} = 0.001$ and Equation 5.10 into Equation 5.1, the value of g_{conv} can be obtained from the value of k predicted by Equation 5.9. The following sections

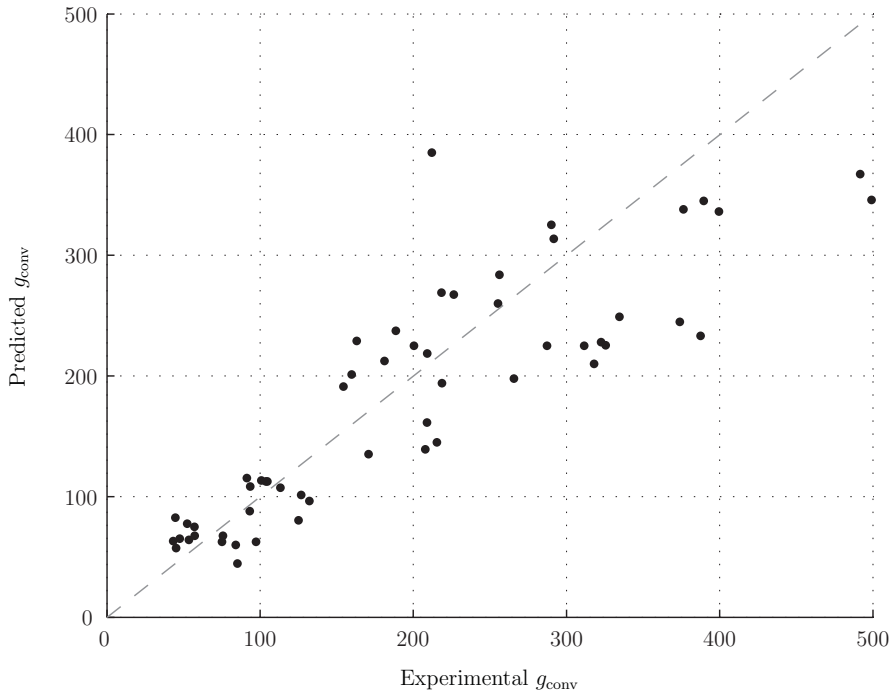


Figure 5.8 Predicted and experimental values of g_{conv} for F3.

provide the results for the predicted and experimental values of g_{conv} for each variation of the three Optimal Generation Functions considered in Section 3.2.

5.3.2.2 F3

F3 considered changes to the degree of epistatic interactions (F3A), roughness and multimodality (F3B), and salience (F3C), of the Rastrigin Function. A plot of the g_{conv} values predicted using Equation 5.9 and Equation 5.1 can be seen on the y-axis of Figure 5.8, with the value determined experimentally from the parametric study undertaken in Section 3.2 plotted on the x-axis. The grey dashed line is the line of perfect prediction. It can be seen from Figure 5.8 that while there is some scatter in the larger values of g_{conv}, generally close predictions of the experimental g_{conv} values are obtained. The coefficient of determination for the data presented in Figure 5.8 is $R^2 = 0.706$.

5.3.2.3 F4

As there was some slight overestimation of the k values for F4 (Griwank's Function) in Figure 5.6, it is not surprising that there is also some overestimation in the predicted values of g_{conv} compared to the experimentally observed values, as seen in Figure 5.9. However, unlike the k predictions, the slight overestimation is not as prominent in the

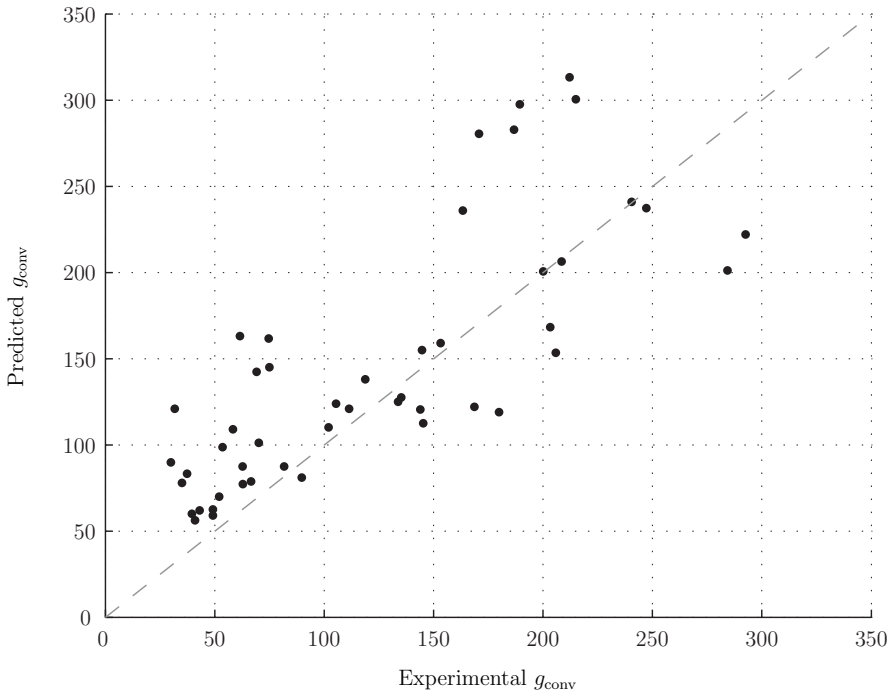


Figure 5.9 Predicted and experimental values of g_{conv} for F4.

g_{conv} predictions, as for some of the variations of F4, the predicted value is slightly below the line of perfect prediction. An $R^2 = 0.607$ is obtained for the F4 results seen in Figure 5.9.

5.3.2.4 F6

As the g_{conv} values are derived from the k values, it is not surprising that for F6, a grouping of the values of g_{conv} for different problem sizes is again observed in Figure 5.10. Some of the predicted values of g_{conv} can be seen to produce a significant overestimation when compared to the experimental values around the experimentally observed value of $g_{\text{conv}} \approx 200$. This cluster of functions all have $l = 10$. Apart from the over prediction for these few functions, there is generally good agreement between the predicted and the experimental values, with a coefficient of determination of $R^2 = 0.668$.

5.4 Discussion

The results presented in Section 5.2 indicate that a very strong relationship was identified between the statistics produced from two of the three fitness function statistics proposed in Chapter 4 and the change in population variance due to selection. The final relationship

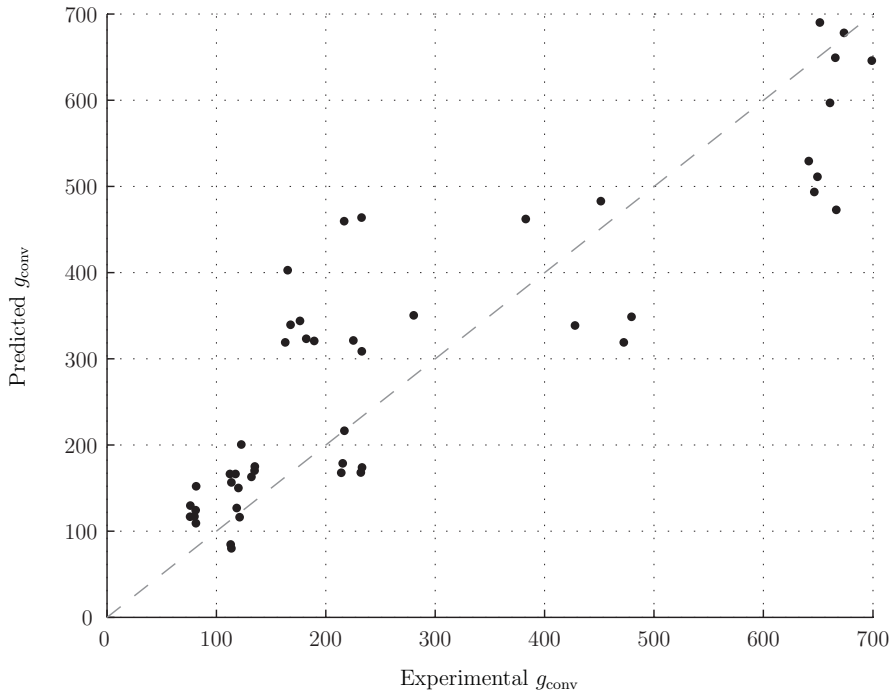


Figure 5.10 Predicted and experimental values of g_{conv} for F6.

identified, seen in Equation 5.9, produced a strong correlation between the predicted and experimental values of k, seen in Figure 5.4 with $R^2 = 0.8118$. The biggest errors in the predictions, seen for some functions with an experimentally observed $k > 0.98$, can be attributed to a slight underestimation of either the correlation length, R_{l}, or the total positive correlation in the fitness function, R_{T}.

The fact that this relationship has been identified is in itself an important result, as the relationship between fitness function characteristics and GA behaviour has not been quantified previously. Generally, fitness function statistics have been compared to the final solution quality found by a GA, and more often than not it appears that the statistics do not provide any useful information. However, this can be explained by a lack of consideration to the calibration of the algorithm. The results presented in this chapter demonstrate that the information provided by the proposed spatial correlation measure, along with the dominance statistic, can be directly related to the change in GA population variance due to selection.

The form of the relationship identified in Equation 5.9 indicates that the roughness of the fitness function has a large effect on the convergence of the GA. The correlation length has been used in a number of studies to investigate the behaviour of GAs. Along with the correlation length, however, two more statistics that provide information about

the roughness of a fitness function were required to accurately predict GA convergence. From the input selection results presented in Table 5.2, it can be seen that these two statistics had the strongest relationship with the k values, more so than the problem size, l, or the correlation length, R_{l}. This may explain why, along with a lack of consideration to the GA parameter values, previous studies relating the correlation length alone to GA behaviour have been largely unsuccessful.

In Section 5.2 a relationship between the proposed fitness function statistics and the decay of GA population variance was obtained, and validated on different function classes. Quantitative genetic theory provided the relationship to compute the number of generations before convergence from the decay in population variance, along with the initial and final standard deviation of the values in the population. While it is beneficial to have this theoretical basis, it is a significant result for the relationship developed to hold in practice, with all GA influences considered. The calculation of the fitness function statistics is based on a random sample of solutions, therefore some variability in the predicted values will arise from this sampling. Along with this variability in the predicted g_{conv} values, the experimentally observed values may also have a large degree of variability, as they have been determined from large scale parametric studies. The average of 30 GA runs with different initial populations has been used in an attempt to reduced this variability, however, the fact that the GA is a probabilistic process will also introduce some variability into the experimental results. In addition, even with the large number of GA runs that have been undertaken, only a limited number of potential population sizes have been tested, which may also add to the variation in the experimental results. However, even with the potential variability in the data, the results for predicting the observed g_{conv} in Figure 5.8, Figure 5.9 and Figure 5.10 indicate that Equation 5.9 provided a definite relationship to the experimentally observed g_{conv} for the range of test functions considered.

The fact that a useful relationship was obtained using the methodology proposed in Section 1.2 suggests that a number of the assumptions made, outlined in Section 2.5.2, were reasonable. The first important assumption was that the statistics applied to the fitness function were a useful approximation to the true fitness landscape. In other words, that the distance between points in the search space used for the spatial correlation measure is a suitable approximation to the solutions that are most likely to be produced by the GA operators of crossover and mutation. If this was not the case, the spatial correlation statistic would not have been related to the k value, as indicated by the input selection results presented in Table 5.2.

The second important assumption made in the methodology implemented was that

only the selection operator significantly changed the population variance each generation. This assumption implies that the crossover and mutation operators do not alter the population variance in a significant way. If additional variance was reintroduced into the population through these operators, it would be expected that the predicted value of g_{conv} would be lower than the observed experimental value, as the population will be predicted to converge quicker than in reality. However, the results in Section 5.3.2 do not display this, as in a number of cases in Figure 5.8, Figure 5.9 and Figure 5.10, the predicted g_{conv} is greater than the experimental value. Therefore, from these results, it can be concluded that the assumption is valid; that crossover and mutation operators selected for the GA used in this work did not have a significant effect on the observed population variance.

5.5 SUMMARY

This chapter has tested the final hypothesis of this thesis, that the number of GA generations before the population converges to one solution can be predicted using the information provided by fitness function statistics. A large data set of function characteristics and corresponding changes in population variance was collected for functions derived from a Fourier Series relationship with controllable roughness, epistatic interactions and salience characteristics. Based on theoretical modelling results, a deterministic relationship between the fitness function characteristics and the decay in population variance was identified.

The relationship was found to compare favourably for functions not belonging to the Fourier Series test functions used in the calibration of the relationship, displaying its generality. Based on the decay in population variance, the number of generations before convergence was calculated. The predicted number of generations based on the fitness function characteristics also compared favourably to experimentally observed values, again displaying the accuracy of the relationship identified.

It is a significant finding that fitness function measures can be directly related to GA parameter values, in this case the number of generations before the GA will converge. This is not surprising from a practical point of view, as based on experience GA practitioners know that the characteristics of the fitness function affect the most suitable GA parameters. Furthermore, a number of theoretical studies such as Harik et al. (1999) suggest that function characteristics, such as the signal to noise ratio and number of building blocks, affect the most suitable population size. However, fitness function statistics have generally been shown to be poor predictors of GA performance. The results presented in this chapter demonstrate firstly that fitness function measures can be used to accurately

characterise a fitness function, and secondly, that the information provided by these measures can be used to determine the most suitable GA parameter values, and not to predict GA convergence, as previously attempted.

The relationship identified is extremely beneficial in assisting with the calibration of GA parameters. By applying the spatial correlation measure and the dominance measure to a given fitness function, the number of generations before a GA will converge for the given fitness function can be predicted. For most WDS optimisation problems, along with most 'real world' optimisation problems, the optimal solution is never found, therefore by making use of the time available before a solution is required, the most influential GA parameter, the population size, can be determined. In the following chapter the prediction of the population size for a number of optimisation problems is considered with different values for the remaining GA parameters. The result from these analyses is a complete GA calibration methodology based on the characteristics of the fitness function.

Chapter 6

GA Calibration Methodology

In Chapter 5 it was demonstrated that fitness function statistics can be used to estimate the number of generations before the GA population can be expected to converge. Based on the estimated number of generations and the time available to solve a fitness function, the most important GA parameter, the population size, can be determined. However, this is only one of the GA parameter values that must be set before the GA can be applied to a given fitness function. The first part of this chapter is dedicated to using the results from the parametric study undertaken in Chapter 3 to identify relationships that can be used to set the remaining GA parameter values. Based on these observations, a complete GA calibration methodology utilising the characteristics of the fitness function is proposed in Section 6.2.

A second approach to determine the population size is proposed in Section 6.3. This approach is much simpler to apply than the first method proposed, as the fitness function statistics do not need to be computed. However, in this case the GA is assumed to converge due to genetic drift, and convergence due to selection pressure is not considered.

The remainder of this chapter is then dedicated to testing the GA calibration methodology proposed in Section 6.2 and Section 6.3 against other GA calibration methods that are available. The different calibration methods considered are outlined in Section 6.4.1, before the performance of each calibration method is evaluated over a wide range of fitness functions and convergence criteria in Section 6.4.4 and Section 6.4.5. A discussion of the results and concluding remarks on the performance and suitability of the different GA calibration methods considered in this chapter are made in Section 6.5 and Section 6.6, respectively.

6.1 The Relationship Between GA Parameters

It is likely that the most suitable GA parameter values are related to each other. For example, it might be expected that a small population size generally performs well with a large probability of mutation, and *vice versa*. To determine if any relationships exist between the best values of the GA parameters, the MI was computed between the values found for each pair of GA parameters. If the best values for two GA parameters are related to each other then there will be a high MI between their values, otherwise if the parameter values are independent of each other, the MI computed will be close to zero.

The two classes of functions identified in Chapter 3 have been considered separately to investigate the relationships between the GA parameters. The Optimal Generation Functions considered in Section 3.2 have been used for the Optimal Generation Function class, while the four problem sizes of the four Maximal Generation Functions tested in Section 3.1 have been considered for the Maximal Generation Functions.

Based on the results from the fitness function statistics, g_{conv} was predicted for each of the Optimal Generation Functions in Section 5.3.2. The best performing GA parameters after the predicted number of generations before the GA population will converge have been identified for each function. Similarly to the approach taken in Section 3.1.1, the best performing GA parameter values were determined from a Student's t-test with a 95% confidence level. For each function, the best parameter values were identified after $FE = g_{\text{conv}}N$ (to the nearest 1000 evaluations), for each population size considered in the parametric study, namely $N = 10$, 50, 100, 200, 400 and 800. For example, for function F3A with $l = 30$, $m_{\text{BB}} = 1$ and $\delta_{\text{BB}} = 1$, the predicted value of g_{conv} in Section 5.3.2 was $g_{\text{conv}} = 212$. For $N = 10$, the GA parameter combinations that produced statistically the best results after $FE = g_{\text{conv}}N = 2000$ were identified. The best combinations of GA parameter values that contained $N = 10$ were taken as the most suitable GA parameters to use with this population size for the predicted number of generations. All of the best performing GA parameter sets were identified for each population size considered in the parameteric study, for all 136 Optimal Generation Functions considered in Section 3.2. This analysis produced a total of 1858 GA parameter sets to be analysed for the Optimal Generation Functions.

For the Maximal Generation Functions there does not exist a g_{conv}, as these problems were most efficiently solved with as many generations as possible of a small population size. Therefore, the best GA parameter values can be considered after any FE before the optimal solution is located, unlike only at $FE = g_{\text{conv}}N$ for the Optimal Generation Functions. To produce a suitable set of GA parameter values, similar to that obtained

Table 6.1 GA parameter interactions for Optimal Generation Functions.

Rank	X_1	X_2	$I(X_1, X_2)$
1	N	p_m	0.557 (0.009)
2	N	e	0.160 (0.003)
3	p_m	e	0.155 (0.002)
4	p_m	c	0.056 (0.003)
5	p_m	p_c	0.032 (0.004)
6	N	p_c	0.028 (0.005)
7	N	c	0.020 (0.003)
8	e	c	0.015 (0.001)
9	p_c	e	0.012 (0.002)
10	p_c	c	0.007 (0.002)

Table 6.2 GA parameter interactions for Maximal Generation Functions.

Rank	X_1	X_2	$I(X_1, X_2)$
1	N	p_m	0.130 (0.022)
2	p_m	p_c	0.074 (0.035)
3	p_m	e	0.051 (0.014)
4	p_m	c	0.027 (0.021)
5	N	p_c	0.022 (0.015)
6	N	e	0.015 (0.007)
7	p_c	e	0.015 (0.013)
8	p_c	c	0.012 (0.011)
9	N	c	0.007 (0.010)
10	e	c	0.001 (0.005)

for the Optimal Generation Functions, the best GA parameters have been considered at five equal spacings before the optimal solution was found for F1, F2 and F5. For F7, the GA never converged to the optimal solution, therefore the maximum number of function evaluations used, $FE = 500\,000$, has been used as the upper limit. Hence, for F7, the best performing GA parameters have been considered after $FE = 100\,000$, $200\,000$, $300\,000$, $400\,000$, and $500\,000$. This analysis produced a total of 282 GA parameter sets to be analysed for the Maximal Generation Functions.

The MI was computed between every pair-wise combination of the GA parameters, to determine if there were any relationships between their best performing values. The results from this analysis are presented in Table 6.1 for the Optimal Generation Functions, and Table 6.2 for the Maximal Generation Functions. The tables rank each pair of GA parameters by their MI value, where a higher value indicates a stronger relationship. The values in brackets are the 95% confidence levels produced from 1000 bootstraps of the data for each pair of parameters. This involved computing the MI between a random re-ordering of the data, removing any true interaction between the parameters. This process was repeated 1000 times, and the MI value that was greater than the MI computed for a random reordering between the parameter values 95% of the time is presented in brakets in Table 6.1 and Table 6.2. Therefore, if the MI computed between two GA parameters is similar to the significance level, it can be concluded that the interaction between the parameters is not significant, and is due to chance alone.

The results for the Optimal Generation Functions, presented in Table 6.1, indicate that

there exists a very strong interaction between the best values for the population size and probability of mutation for the functions considered, with $I(N, p_m) = 0.557$. This result agrees with the observations made in Chapter 3, where a large probability of mutation is required for a small population size, while a larger population size performs best with a lower mutation rate. The interactions between the remaining pairs of GA parameters are much less significant than that between the population size and the probability of mutation, with the next highest mutual information being between the population size and the number of elite solutions, with $I(N, e) = 0.160$. However, all MI values are greater than the significance levels computed from bootstrapping the data, indicating that there is a weak relationship between all parameters.

The MI computed between each pair of GA parameters for the Maximal Generation Functions can be seen in Table 6.2. The MI computed between the GA parameters from this data set are much lower than those computed for the Optimal Generation Functions, with the highest MI term being $I(N, p_m) = 0.130$. This can be attributed to much less variation in the data for these functions, as for almost all of the GA parameter sets collected $N = 10$, along with generally very similar values for p_m and e. The relationships between the GA parameters and the frequency of occurrence of each of the GA parameter values considered for both the Optimal and Maximal Generation Functions are considered in more detail in the remainder of this section.

6.1.1 Population Size

Each population size considered in the parametric study has been used for the Optimal Generation Functions, where the relevant FE for each population size has been computed from the predicted value of g_{conv} as $FE = g_{\text{conv}}N$. For the Maximal Generation Functions, a histogram of the best performing population sizes is presented in Figure 6.1. As expected, it can be seen from Figure 6.1 that $N = 10$ is by far the best population size for the Maximal Generation Functions, with only a few occurrences for $N = 50$ and 100. The interaction between N and p_m of $I(N, p_m) = 0.130$ observed in Table 6.2 for the Maximal Generation Functions is produced by these few occurrences of larger population sizes occurring with low probabilities of mutation of $p_m = 0$ and $p_m = 0.2$.

6.1.2 Probability of Mutation

A number of previous studies (Schaffer et al., 1989, for example) have observed a relationship between the population size and the most suitable probability of mutation. The MI results presented in Table 6.1 and Table 6.2 reinforce this relationship, with the high-

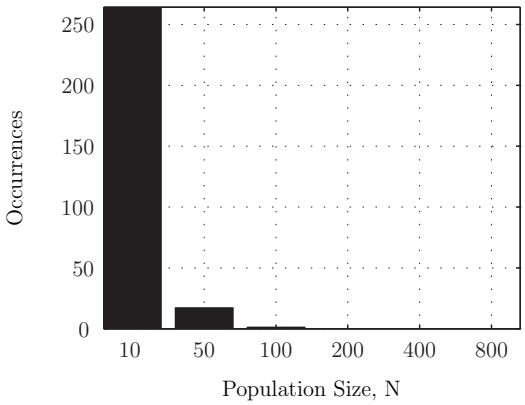


Figure 6.1 Best performing population sizes for Maximal Generation Functions

est values of $I(N, p_m) = 0.557$ and $I(N, p_m) = 0.130$, respectively. A boxplot of the different probabilities of mutation that occurred for each population size is presented in Figure 6.2 for the Optimal Generation Functions. From Figure 6.2, it can be seen that there is indeed a strong inverse relationship between N and p_m in this dataset, with large p_m only suitable for small N, and *vice versa*.

For the Maximal Generation Functions, a histogram of the number of occurrences of each probability of mutation considered in the parametric study can be seen in Figure 6.3(a). From Figure 6.3(a), it can be seen that all of the values of p_m considered perform well at least once, however $p_m = 0.4$ is by far the most common. The probabilities of mutation are not compared to the population size for the Maximal Generation Functions, as 93.6% of the GA parameter sets for these functions have $N = 10$, as seen in Figure 6.1.

From these results, it is clear that there is a relationship between the best performing values for the probability of mutation and the population size. Therefore, it is proposed that the probability of mutation be calculated from the population size, which is first determined by $N = FE/g_{\text{conv}}$ for an Optimal Generation Function, or $N = 10$ for a Maximal Generation Function. The grey dashed line in Figure 6.2 is the line $p_m = 5/N$, and it can be seen that this relationship provides a good fit to the centre of the box plot for each population size. This relationship also provides a close approximation to the most common $p_m = 0.4$ for the Maximal Generation Functions, as $p_m = 5/10 = 0.5$. However, this value for the probability of mutation was not included in the parametric study to be tested. As the probability of mutation used in this work is the probability of mutating each solution string in the population, this relationship suggests that there should be 5 mutations per generation, irrespective of the population size or dimension of the

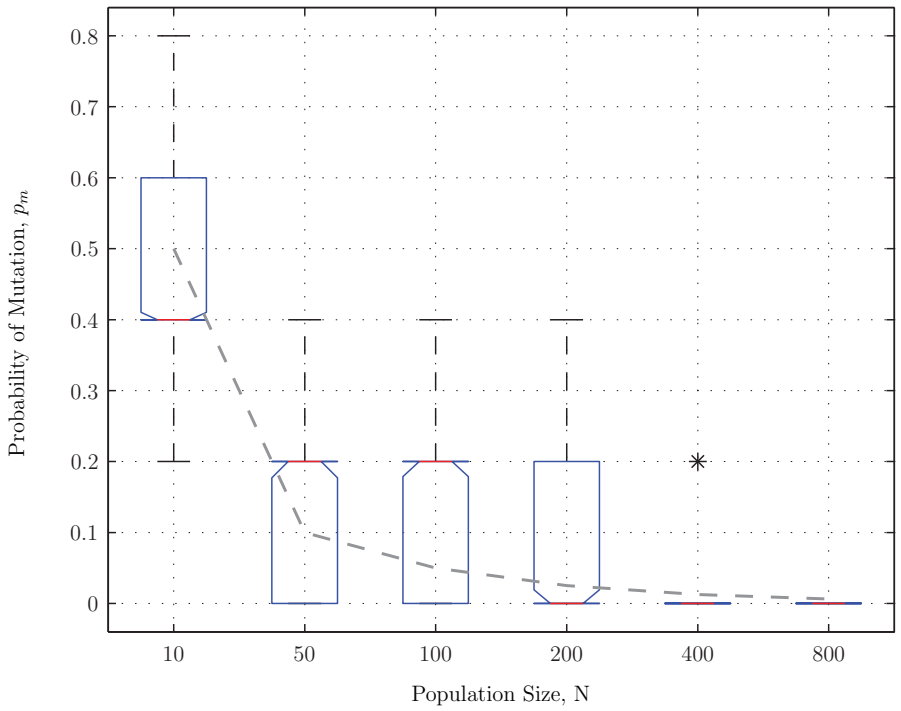


Figure 6.2 Population size against probability of mutation for Optimal Generation Functions

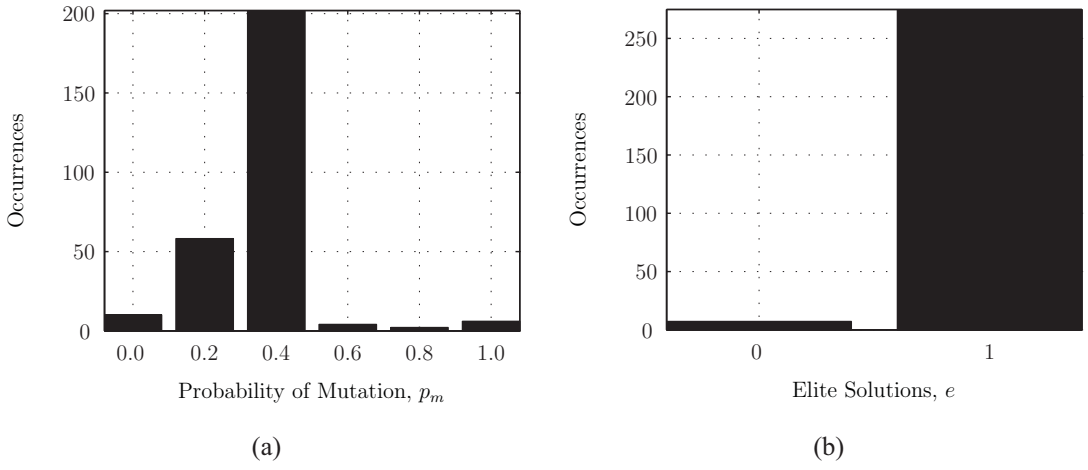


Figure 6.3 Best performing (a) mutation rates and (b) number of elite solutions for the Maximal Generation Functions.

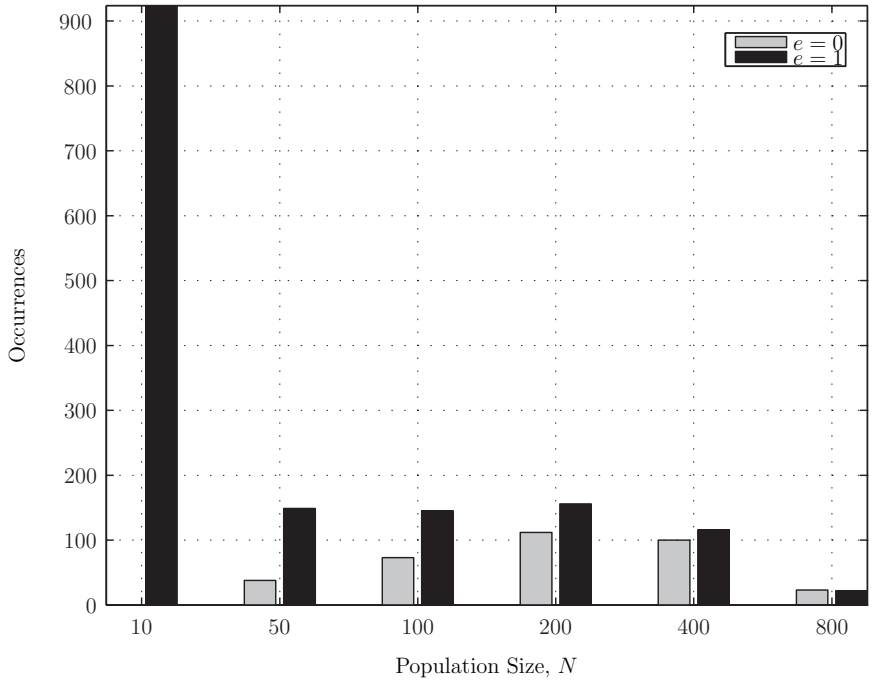


Figure 6.4 Number of elite solutions for each population size for Optimal Generation Functions

problem. It also indicates that there will always be some mutation to introduce diversity in the population, even if mutation occurs with a low probability for larger population sizes.

6.1.3 Elitism

The next strongest relationship identified between the GA parameters for the Optimal Generation Functions by the MI analysis presented in Table 6.1 was between the number of elite solutions and population size, and the number of elite solutions and the probability of mutation, with $I(N, e) = 0.160$ and $I(p_m, e) = 0.155$, respectively. Given that N and p_m were strongly related to each other, if one parameter is related to the elitism operator, it is not surprising that both parameters are related to the elitism operator.

The number of occurrences for a GA with ($e = 1$) and without elitism ($e = 0$) for different population sizes for the Optimal Generation Functions is presented in Figure 6.4. It can be seen from Figure 6.4 that $e = 1$ performs the best with all the population sizes considered in the parametric study. The difference is largest for the smaller population sizes, where good solutions are easily lost without an elitist operator.

The number of occurrences for a GA with and without elitism for different mutation

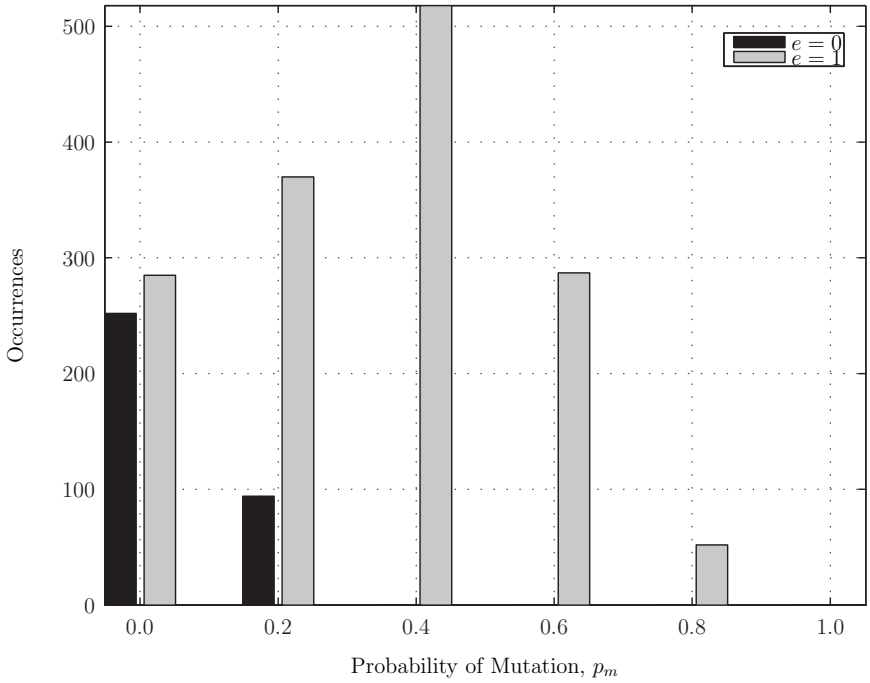


Figure 6.5 Number of elite solutions for each probability of mutation for Optimal Generation Functions

rates for the Optimal Generation Functions is presented in Figure 6.5. Similarly, it can be seen from Figure 6.5 that $e = 1$ performs the best with all the probabilities of mutation considered in the parametric study, with $e = 0$ only suitable for small mutation rates of $p_m = 0$ and 0.2, where good solutions in the population are less likely to be mutated, and subsequently lost.

It can be seen from Figure 6.3(b) that $e = 1$ performs much better for the Maximal Generations Functions, as the GA with an elitist operator produced the best results in 97.5% of the parameter sets. These results reinforce Markov Chain modelling results such as those presented by Lozano et al. (1999) and Suzuki (1995), that suggest that an elitist operator is useful, and required to ensure theoretical convergence to the optimal solution. Therefore, based on these results and the results from Markov Chain modelling studies such as these, $e = 1$ is used in the GA calibration methodolgy proposed in this work.

6.1.4 Standard Deviation of Crossover

For the Optimal Generation Functions, the GA parameter that had the strongest relationship with the standard deviation of the crossover operator was the probability of mutation,

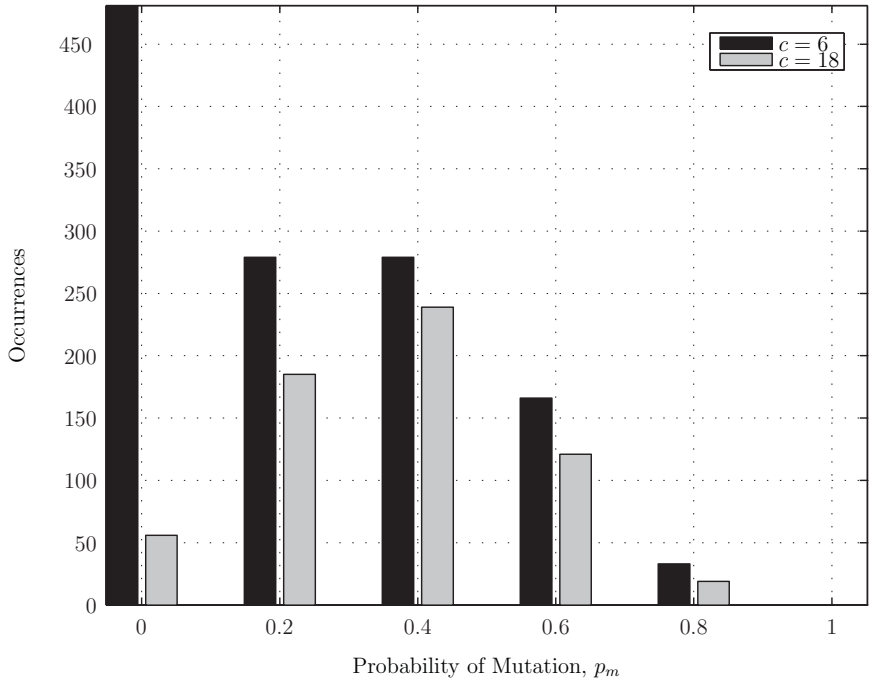


Figure 6.6 Standard deviation of crossover for each probability of mutation for Optimal Generation Functions

with $I(p_m, c) = 0.056$, as seen in Table 6.1. Both these GA parameters control the amount of variation of the population, as both a high probability of mutation and a smaller fraction of the distance between the parent values for σ will generally produce solutions that are further away in the search space from their parent solutions. Therefore, it is not surprising that these two GA parameters are related, as one or the other of these parameters could be used to control the degree of variation from generation to generation. However, the mutual information value between these two parameters was quite low, indicating the relationship is not very strong, but is still an order of magnitude higher than the significance level of 0.003.

The number of occurrences for each c considered for each mutation rate in the parametric study for the Optimal Generation Functions is presented in Figure 6.6. Similarly to the case for the elitism parameter, it can be seen from Figure 6.6 that $c = 6$ performed the best for all p_m considered. However, the largest difference between the two occurred for the lower mutation probabilities. In this case, the smaller value of c produces a wider distribution for the crossover operator, producing greater changes in the decision variable values in the population, taking over from mutation, which is occurring at low probabilities, if at all.

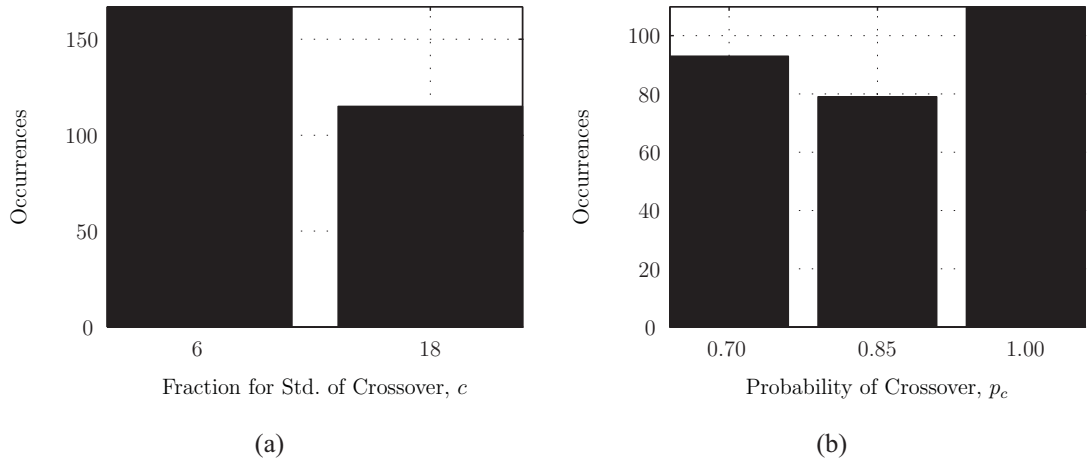


Figure 6.7 Best performing (a) fraction for the standard deviation of crossover and (b) probabilities of crossover for the Maximal Generation Functions.

The number of occurrences for the two fractions for the standard deviation of the crossover operator considered for the Maximal Generation Functions can be seen in Figure 6.7(a). From Figure 6.7(a), it can be seen that this parameter did not have a large effect on the performance of the GA, as the number of occurrences is relatively even, however, there were slightly more occurrences for $c = 6$. The MI scores presented in Table 6.1 indicate that there were no strong interactions between the standard deviation of the crossover operator and the other GA parameters, as the highest score for this parameter was $I(p_m, c) = 0.027$, only slightly higher than the significance level of 0.021. For the Maximal Generation Functions, new values are found through mutations of a small population, and therefore the crossover parameter values do not have a significant effect on GA performance.

The results from the MI analysis indicate that the standard deviation of the crossover operator parameter was not significantly related to the value of any of the other GA parameters, and that $c = 6$ performed the best for both the Optimal and Maximal Generation Functions. Therefore, $c = 6$ has been adopted in the GA calibration methodology.

6.1.5 Probability of Crossover

The final GA parameter value that must be determined before the GA can be applied is the probability of crossover. The MI results indicate that this parameter had the strongest relationship with the probability of mutation and the population size, with $I(p_m, p_c) = 0.032$ and $I(N, p_c) = 0.028$ for the Optimal Generation Functions, and $I(p_m, p_c) = 0.074$ and $I(N, p_c) = 0.022$ for the Maximal Generation Functions. While these MI values

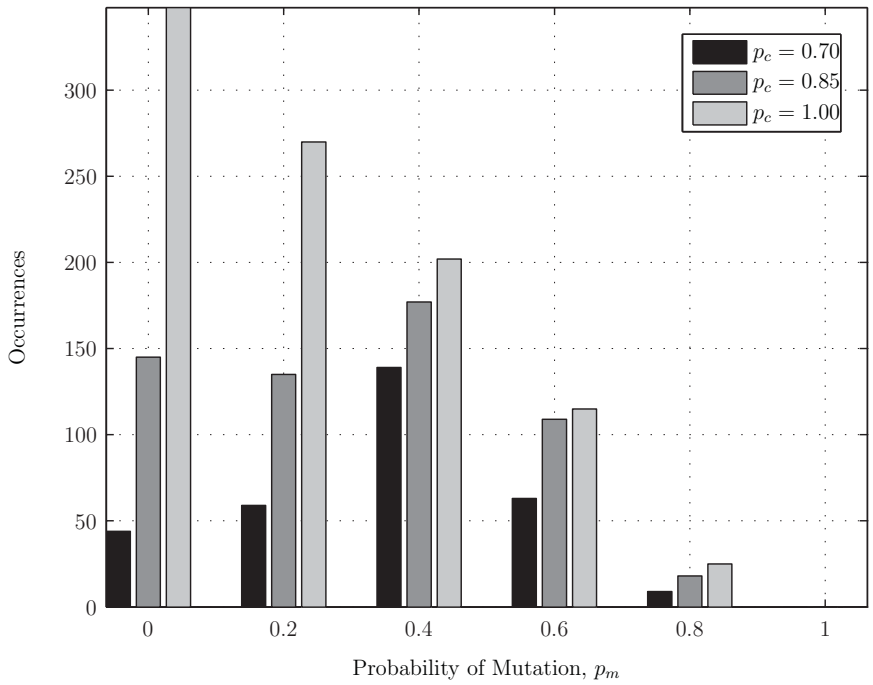


Figure 6.8 Probability of crossover for each probability of mutation for Optimal Generation Functions

are greater than their corresponding significance level, the MI values suggest that the relationship is very weak, and unlikely to effect the performance of the GA.

The number of occurrences of each probability of crossover for each probability of mutation considered in the parametric study undertaken in Chapter 3 is given in Figure 6.8 for the Optimal Generation Functions. It can be seen that $p_c = 1$ produced the best results, irrespective of the probability of mutation. Similarly to the case for the fraction of the distance between parent solutions for the distribution of the crossover operator variable, the largest differences are seen for the lower probabilities of mutation. This result can be better explained by the relationship with the population size. As observed above, there was a strong relationship between the probability of mutation and population size, where the lower probabilities of mutation performed best with the larger population sizes. For the larger population sizes, the crossover operator is used as the dominant operator to locate better solutions, as opposed to relying mainly on the mutation operator for problems that are best solved with smaller population sizes. Therefore, for the case where crossover is the more dominant operator, the probability of crossover has a greater influence on the performance of the GA.

This observation is reinforced in Figure 6.9, presenting the most common probabil-

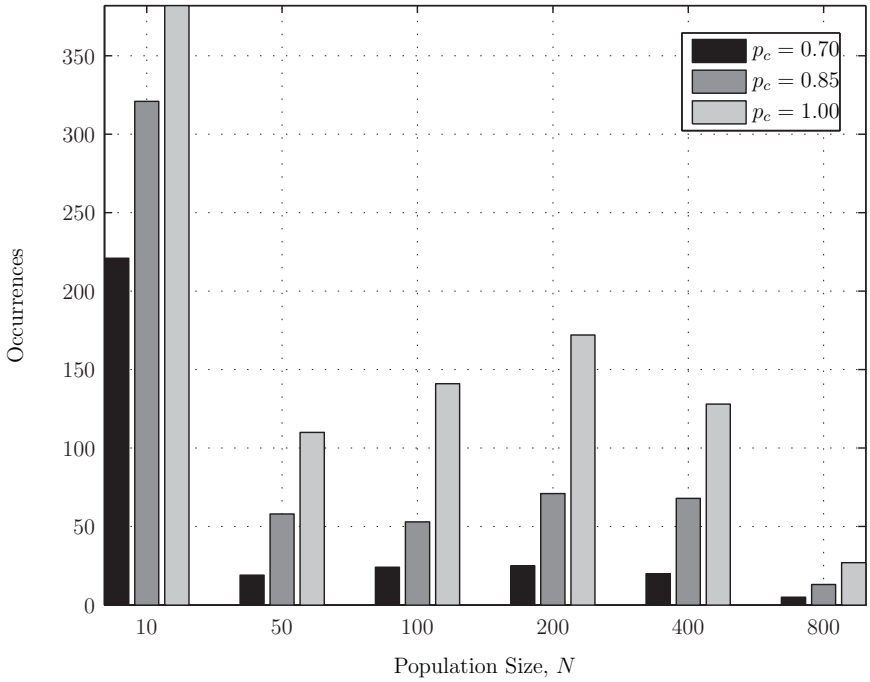


Figure 6.9 Probability of crossover for each population size for Optimal Generation Functions

ities of crossover for each population size for the Optimal Generation Functions. From Figure 6.9, it can be seen that the larger the value of p_c, the more frequently it produced the best results, irrespective of the population size. While $p_c = 1$ was always the best, for $N > 10$ there were only very few occurrences of $p_c = 0.7$, and not many more occurrences of $p_c = 0.85$. However, for $N = 10$, where mutation was relied on to do most of the searching, the value of p_c is not as important, as it can be seen from Figure 6.9 that all values for p_c performed well.

The frequency of occurrence for each probability of crossover tested for the Maximal Generation Functions can be seen in Figure 6.7(b). As was the case for the Optimal Generation Functions, $p_c = 1$ was the most common value in the best performing sets of GA parameter values for the Maximal Generation Functions. The explanation for this result is similar to that for the fraction of the distance between parents for the standard deviation of the crossover operator. For the Maximal Generation Functions, new values are found through mutations of a small populations, and therefore the crossover parameters do not have a significant affect on the GA performance.

These results indicate that $p_c = 1$ performed the best for both the Optimal and Maximal Generation Functions, and that the parameter was not significantly related to the

value of any of the other GA parameters. Therefore, $p_c = 1$ has been adopted for the GA calibration methodology. This result implies that every solution in the population is altered every generation by crossover, therefore making the most efficient use of FE available. Conversely, $p_c < 1$ will result in a number of solutions being re-evaluated every generation, provided a solution tracking operator is not implemented to ensure that only solutions that change from one generation to the next are evaluated.

6.2 GA Calibration Methodology

The previous section identified that the only strong relationship between the GA parameter values for the cases considered was between the population size and the probability of mutation. The MI results presented in Table 6.1 and Table 6.2 indicate that for the remaining combinations of pairs of GA parameters, the interaction was only slightly higher than that computed from a random reordering of the data. The results presented so far in this thesis are summarised in thesis section, providing a complete GA calibration method. A flowchart of the resulting methodology can be seen in Figure 6.10.

The type of fitness function must be determined before the population size can be determined. The classification flow chart developed in Section 3.1.3, presented in Figure 3.13, can be used to classify a given fitness function. By applying the fitness function statistics developed in Chapter 4, a function can be classified as a Maximal Generation Function if: $R_{\text{av}} < 0.05$ ($R_{\text{av}} > 0$ to allow for small random effects in the estimation of the statistic), indicating no structure in the search space; or if a function has a largest building block size of $k_{\text{BB}} = 1$, indicating that there were no interactions between the decision variables, as well as $D < 0.05$, indicating that each decision variable had a similar contribution to the fitness function value. If a function is classified as a Maximal Generation Function, the population size to be used is $N = 10$. Otherwise, a function is an Optimal Generation Function, and the number of generations before convergence, g_{conv}, should be computed by Equation 5.9 and Equation 5.1. From the time available to obtain a solution, the available number of fitness function values can be determined, then the population size to be used is given by $N = FE/g_{\text{conv}}$.

Once the population size has been determined, the probability of mutation can be determined. Section 6.1 identified a strong relationship between these two parameters, and it was determined that the best value for the probability of mutation was $p_m = 5/N$. The remaining GA parameters did not have a significant relationship between each other, and therefore the best values identified in Section 6.1 are used in the calibration methodology.

An elitist strategy is required in the Markov Chain modelling of GA populations to

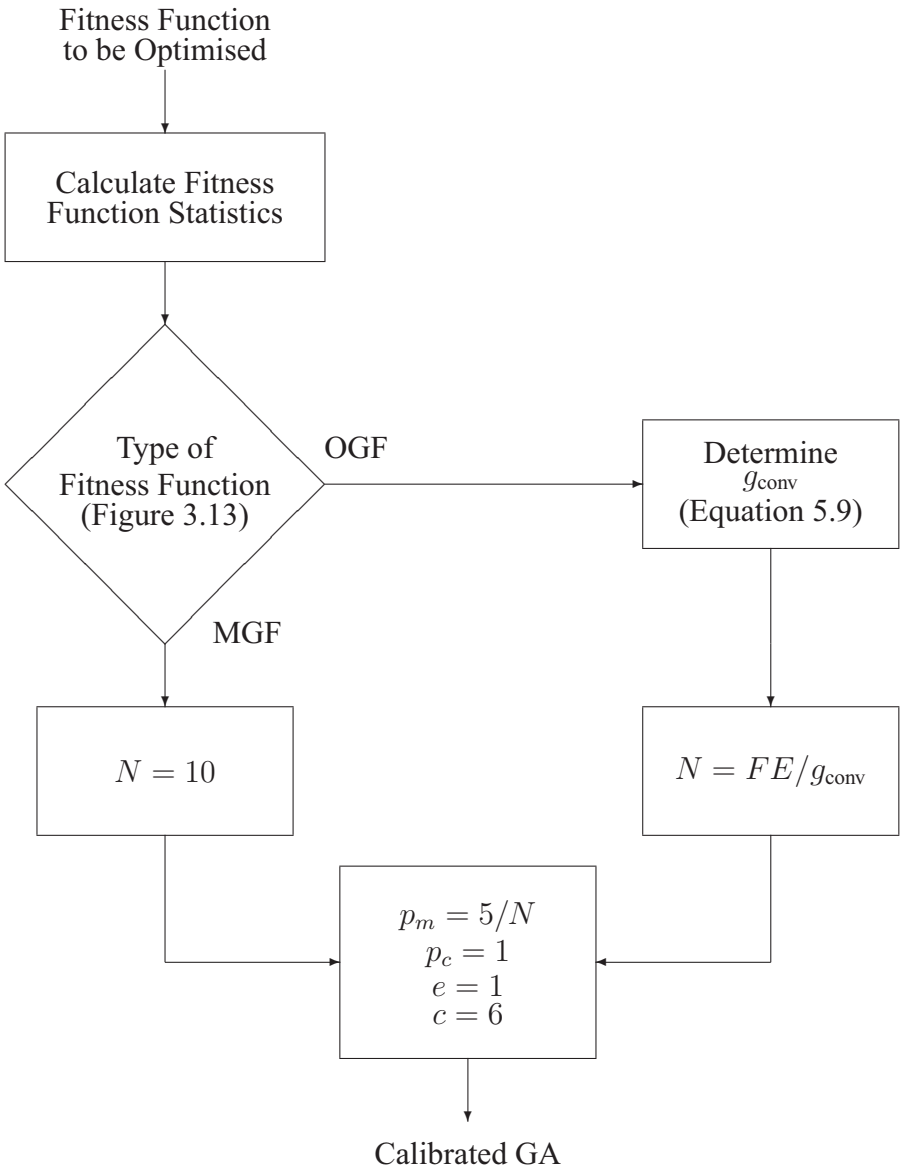


Figure 6.10 The proposed GA calibration methodology

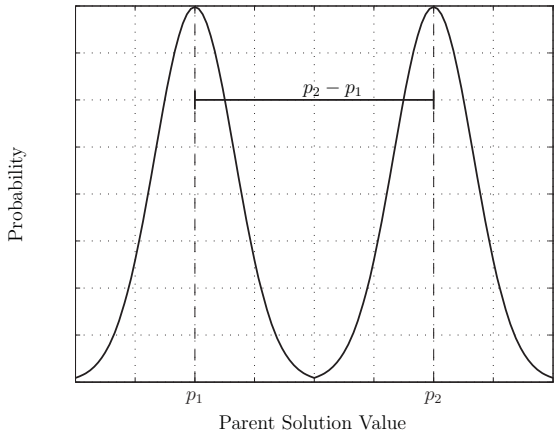


Figure 6.11 Probability distribution used for crossover with $\sigma = (p_1 - p_2)/6$

ensure convergence to the global optimum (Lozano et al., 1999; Suzuki, 1995), so it is not surprising that $e = 1$ performed well in the parametric study. $e = 1$ also ensures that the best solution in the population is not lost in the crossover or mutation process. $p_c = 1$ will make the most efficient use of the FE available, as every solution in the population is altered every generation, and therefore no FE are consumed by re-evaluating solutions. The final GA parameter to be calibrated produced the best performance when $c = 6$. A crossover distribution with a standard deviation based on this parameter value can be seen in Figure 6.11. A value of $c = 6$ will produce a distribution with three standard deviations from each parent solution, and therefore any solution between the two parent solutions can be produced by the crossover operator, with a decreasing probability as the distance from either of the parents is increased.

To summarise these results, the proposed GA calibration methodology, as seen in Figure 6.10, is:

- Calculate the fitness function statistics:
  - Spatial Correlation (Section 4.1);
  - Dominance Measure (Section 4.3);
  - Separability Measure (Section 4.2.2).
- Determine the type of function (Figure 3.13):
  - Maximal Generation Function; or
  - Optimal Generation Function;
    - Determine g_{conv} (Equation 5.9).
- Determine the population size:

  - if a Maximal Generation Function (MGF): $N = 10$;
  - if an Optimal Generation Function (OGF): $N = FE/g_{\text{conv}}$.
- $p_m = 5/N$;
- $p_c = 1$;
- $e = 1$;
- $c = 6$.

6.3 Convergence Due to Genetic Drift

A second GA calibration method is proposed, based on a similar method to determine the population size. However, rather than attempting to predict the convergence due to the selection pressure as a function of the fitness function characteristics, an upper limit to when the convergence will occur is identified, based on when the population will randomly converge due to genetic drift. As genetic drift is independent of the fitness function values, there is no need to estimate the characteristics of the fitness function. Rogers and Pruegel-Bennett (1999) developed analytic expressions for the change in population variance due to genetic drift for a range of selection schemes. The authors derived that for a generational GA with a tournament size of two, the decay in population variance from one generation to the next was $k = 1 - \frac{1}{N}$. Based on the same initial (σ_0) and final (σ_f) standard deviation of the population used to estimate the number of generations from k in Equation 5.1, the population size can be estimated directly by:

$$
\begin{aligned}
g_{\text{conv}} &= \frac{\log(\sigma_f) - \log(\sigma_0)}{\log(k)}, \\
\frac{FE}{N} &= \frac{\log(0.001) - \log\left(\sqrt{\frac{l}{12}}\right)}{\log\left(1 - \frac{1}{N}\right)}, \\
\frac{FE}{N}\log\left(1 - \frac{1}{N}\right) &= -3 - \log\left(\sqrt{\frac{l}{12}}\right). \qquad (6.1)
\end{aligned}
$$

Therefore, by solving the implicit equation in Equation 6.1, the largest population size that will converge due to genetic drift in the number of function evaluations that are available for a given problem size can be determined. As was the case for the GA calibration methodology based on the fitness function characteristics, this approach assumes that the GA operators, other than selection operator, do not significantly alter the population variance. The same values for the remaining GA parameters have been used for this calibration method as for the the proposed methodology above, namely $e = 1$, $p_m = 5/N$,

$p_c = 1$, and $c = 6$.

6.4 Comparison of GA Calibration Methods

The two proposed GA calibration methods have been compared with another GA calibration method, as well as 'typical' GA parameter values, to investigate the effect on GA performance over a range of functions. In this section the different GA calibration methodologies that have been tested are outlined, before the range of test functions that they have been applied to are presented. This is followed the results from this study, firstly presenting the results from the characterisation of the fitness functions, before the comparison of the solution quality found by the different GA calibration methods.

6.4.1 Outline of the Methodologies

Four different methods for determining the GA population size have been selected to be tested on a range of fitness functions. Other approaches for self adaptation of the population size are available, such as Arabas et al. (1994), Bäck et al. (2000), or Eiben et al. (2004). However, these approaches result in replacing the population size with one or more parameters to control the change in population size, hence they contribute to the GA calibration problem, not solve it. The first GA calibration method to be compared is the first methodology proposed in this thesis. This method is called 'Predicted' in the remainder of this chapter. The second GA calibration approach tested is that proposed in Section 6.3, and has been called 'Drift' in the analysis of the results.

The third method selected to be tested was the 'Parameterless' GA calibration method. This GA calibration methodology was first proposed by Harik and Lobo (1999), followed by a number of studies, including Lobo and Goldberg (2004), Reed and Yamaguchi (2004) and Minsker (2005). Reed and Yamaguchi (2004) applied this GA calibration methodology to real coded EAs, specifically Differential Evolution, and therefore a similar approach is adopted for comparison in this work. As is the case in this work, Reed and Yamaguchi (2004) assumed that the population size is the key parameter controlling the reliability and efficiency of EAs. Consequently, the approach involves doubling the population size after convergence is achieved, and standard values are used for the remaining EA parameters. The method starts with a small population size; a value of $N = 10$ was suggested (Reed and Yamaguchi, 2004). After the EA has converged, the population size is doubled and randomly reinitialised, and the best solution from the first GA run is injected into the population. The algorithm then starts again, until the larger population size

has converged, and then it is again doubled in size. The process continues until there is no improvement in solution quality from one run to the next, a desired solution quality is reached, or the maximum FE is reached.

The initial starting population used in this work was $N = 10$. As proposed by Reed and Yamaguchi (2004), the GA has been deemed to have converged after a minimum of $g = Nl$ generations, and after there is no longer at least a 1% improvement in the fitness function value from one generation to the next. After the GA has converged, the population size was doubled and reinitialised with random solutions, along with the best solution from the last GA run. The process was continued until the available number of FE have been made.

The GA calibration methodology proposed by Harik and Lobo (1999) suggests that to ensure building block growth in line with the schema theorem, the probability of crossover must be $p_c \leq (1 - s)/s$, where s is the selection pressure. The relationship has been applied to binary coded GAs, however, as it is derived from the schema theorem, it is also applicable to RCGAs (Goldberg, 1991; Wright, 1991). For this work, the parameter values proposed by Minsker (2005) to satisfy this relationship are adopted, namely $s = 2$ and $p_c = 0.5$. Minsker (2005) suggested that the bitwise probability of mutation should be the maximum value of $p_m = 1/N$ and $p_m = 1/l$, where $p_m = 1/l$ per bit used in that work is, on average, equivalent to $p_m = 1$ per string used in this work. Therefore, for the Parameterless GA calibration methodology, $p_m = 1$ has been adopted. For the remaining GA parameters $c = 6$ and $e = 1$ have been used.

The fourth GA calibration method tested was used to represent the GA parameter values used typically, and therefore is referred to as 'Typical'. The values adopted for this method were:

- $N = 100$;
- $p_m = 1$ per string;
- $p_c = 0.9$;
- $c = 6$;
- $e = 1$.

Each of these four GA calibration methods have also been used in a self-adaptive framework. The four methods described above have been used to determine the population size, and for each method $e = 1$ has been adopted. These parameters are applied at the population level, not the individual solution level, and therefore there is no clear way for the GA to self-adapt these parameter values. For the remaining parameter values, each solution in the population also included a value for p_m, p_c and c to be used by the GA for

that solution. When crossover occurred, the crossover parameter values to be used were randomly selected with an equal probability from one of the two parent solutions to be crossed over. In this way, the GA solves the parameterisation problem in parallel with the fitness function, removing the calibration problem from the user. The range for each parameter considered was $[0 \leq p_m \leq 1]$, $[0.5 \leq p_c \leq 1]$, and $[0.01 \leq c \leq 36]$.

Therefore, a total of eight GA calibration methods have been selected to be tested in the remainder of this chapter. The following section outlines the test functions that have been chosen for the comparison.

6.4.2 Test Functions

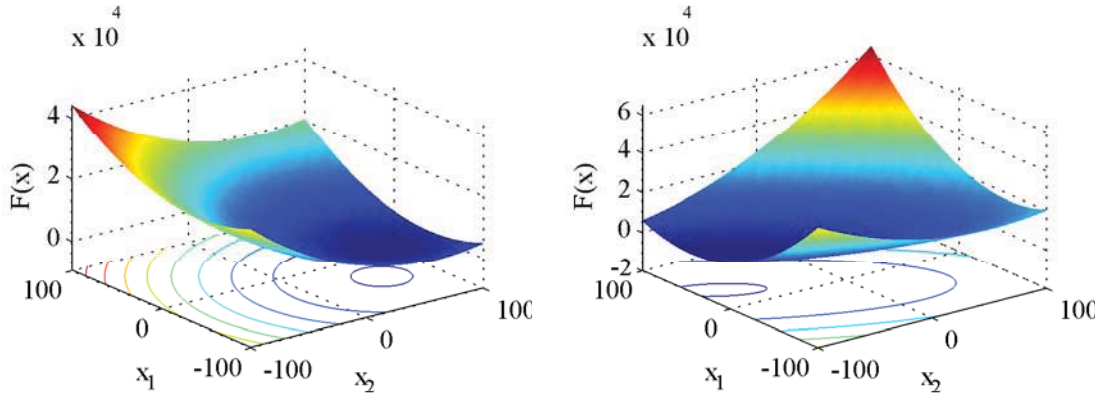
To avoid biasing the results toward the proposed GA calibration methodology, a set of fitness functions that have not been used in the calibration of the methodology have been selected. The test functions developed for the special session on real-parameter optimisation at the 2005 IEEE Congress on Evolutionary Computation (Suganthan et al., 2005) have been used as the test functions to compare the different GA calibration methods. These functions have been composed to posses a number of different properties, including (Suganthan et al., 2005):

- shifted functions (i.e. each variable has a different value at the optimum);
- rotated functions;
- varying structure in the fitness landscape;
- continuous and non-continuous functions;
- a range of the number of local optima;
- a range in the size and shape of local optima basins with respect to the basin of attraction for the global optimum;
- additively non-separable or non-decomposable;
- global optimum at the search boundary for some variables; and
- scalability into higher dimensions.

A summary of the functions used is presented in Table 6.3, and two–dimensional plots of each function are presented in Figures 6.12(a) to 6.21(b). Formal definitions of the functions can be found in Suganthan et al. (2005). Each function has been considered for problem sizes of $l = 10$, 30, and 50. For the different GA calibration methods, convergence times corresponding to $FE = 10^3$, 10^4, and 10^5 have been considered, along with 3×10^5 for each problem with $l = 30$, and 5×10^5 for each problem with $l = 50$. Each function has been optimised over the range $x_i \in [-5, 5]$ with the exception of f1–f6 and f14, which have the range $x_i \in [-100, 100]$, f8 over $x_i \in [-32, 32]$, f11 over

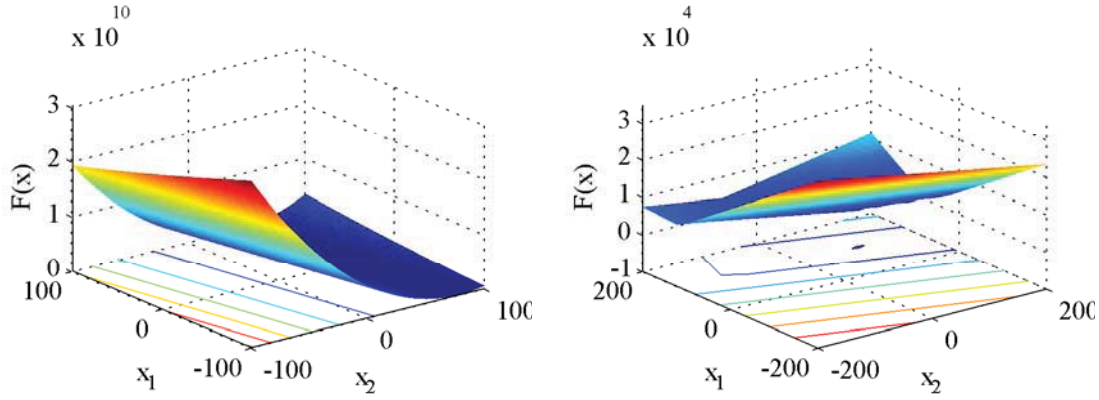
Table 6.3 Test functions for the GA calibration methods.

UNIMODAL FUNCTIONS	
f1	Shifted Sphere Function
f2	Shifted Schwefels Problem 1.2
f3	Shifted Rotated High Conditioned Elliptic Function
f5	Schwefels Problem 2.6 with Global Optimum on Bounds
MULTIMODAL FUNCTIONS	
Basic Functions	
f6	Shifted Rosenbrocks Function
f8	Shifted Rotated Ackleys Function with Global Optimum on Bounds
f9	Shifted Rastrigins Function
f10	Shifted Rotated Rastrigins Function
f11	Shifted Rotated Weierstrass Function
f12	Schwefels Problem 2.13
Expanded Functions	
f13	Expanded Extended Griewanks plus Rosenbrocks Function
f14	Shifted Rotated Expanded Scaffers Function
Hybrid Composition Functions	
f15	Hybrid Composition Function
f16	Rotated Hybrid Composition Function
f18	Rotated Hybrid Composition Function
f19	Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum
f20	Rotated Hybrid Composition Function with the Global Optimum on the Bounds
f21	Rotated Hybrid Composition Function
f22	Rotated Hybrid Composition Function with High Condition Number Matrix
f23	Non-Continuous Rotated Hybrid Composition Function



(a) 2–dimensional plot of f1 (b) 2–dimensional plot of f2

Figure 6.12 Functional form of f1 and f2.

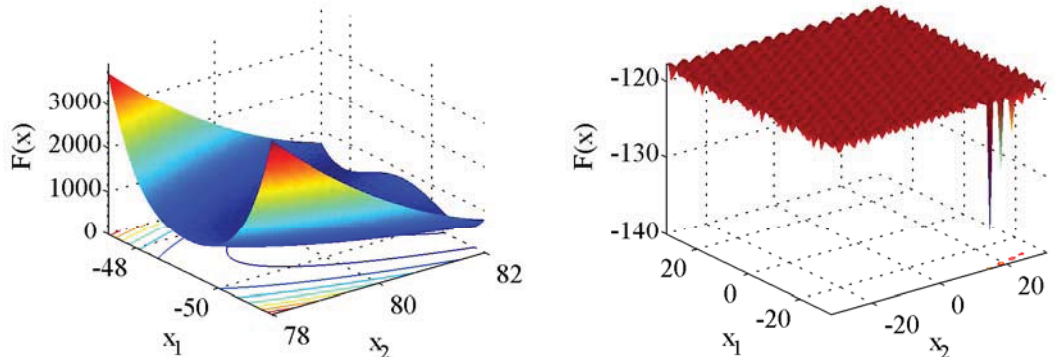


(a) 2–dimensional plot of f3 (b) 2–dimensional plot of f5

Figure 6.13 Functional form of f3 and f5.

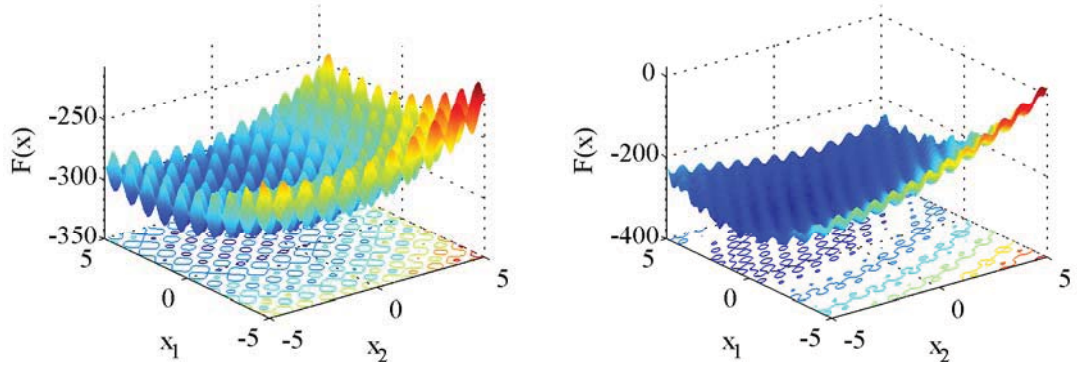
$x_i \in [-0.5, 0.5]$, and f12 over the range $x_i \in [-\pi, \pi]$. The analysis methods proposed by Suganthan et al. (2005) to compare the results from different calibration methods for the test functions have been adopted in this work. For every GA calibration methodology tested, each function in each problem size for each stopping criterion has been run 25 times with different seeds for the random number generator.

Functions that include random noise in the fitness function value (f4, f7, and f24) have not been included in the analysis, as functions such as these do not represent those that would be expected for WDS optimisation, where each set of decision variable values will produce a constant, deterministic, fitness function value. Similarly, the test functions that do not have a set search space (f7, f25) have not been considered in this work, as generally for WDS optimisation the objective is to locate the best solution within a desired range.



(a) 2–dimensional plot of f6 (b) 2–dimensional plot of f8

Figure 6.14 Functional form of f6 and f8.



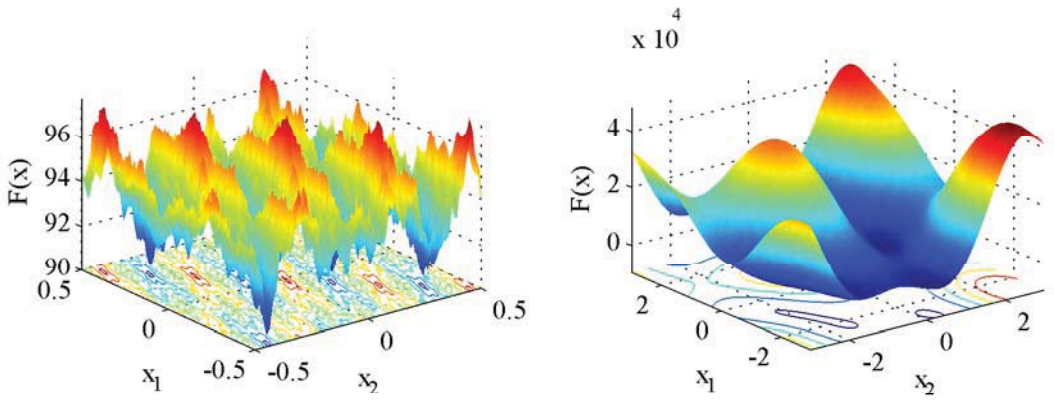
(a) 2–dimensional plot of f9 (b) 2–dimensional plot of f10

Figure 6.15 Functional form of f9 and f10.

Both of these function properties may also present a problem for the fitness function statistics, as the search space must be sampled within a specified area, and it is not clear how well statistics such as the spatial correlation measure will characterise noise in the fitness function values. Therefore, a total of 20 fitness functions have been used for the comparison.

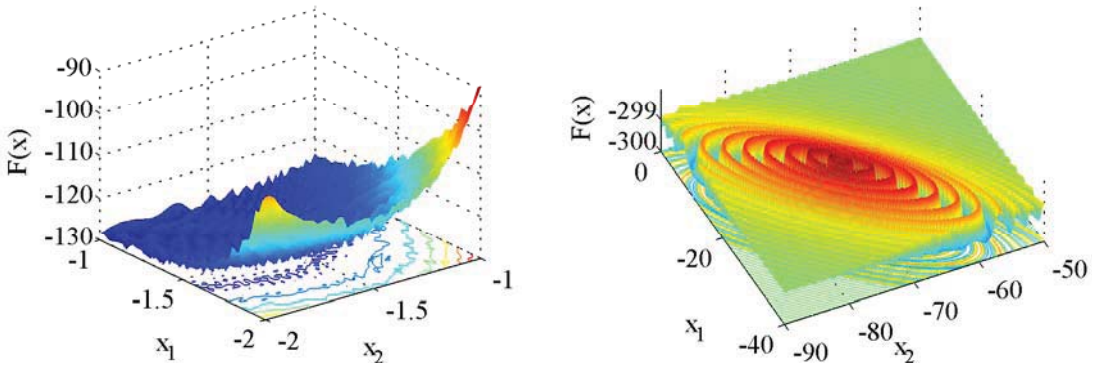
6.4.3 Function Characterisation

Each of the fitness function statistics proposed in Chapter 4 has been applied to the 20 test functions. The fitness function statistics have been computed using $n = 5\,000$, where the samples used to compute the separability measure are reused to compute the spatial corre-



(a) 2–dimensional plot of f11 (b) 2–dimensional plot of f12

Figure 6.16 Functional form of f11 and f12.

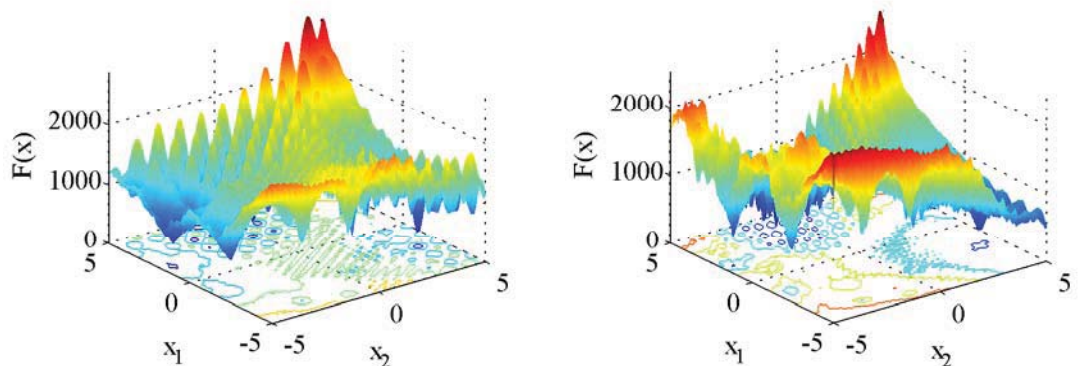


(a) 2–dimensional plot of f13 (b) 2–dimensional plot of f14

Figure 6.17 Functional form of f13 and f14.

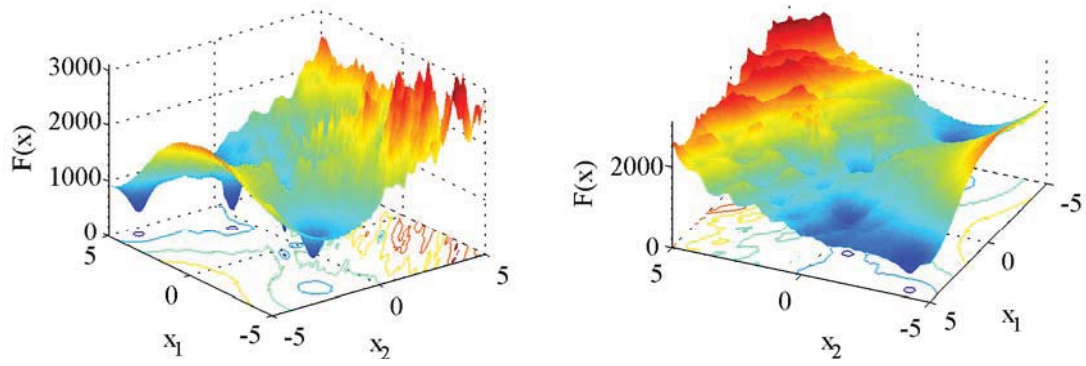
lation and the dominance of the decision variables. The values computed for each statistic for each function in $l = 10$ dimensions are given in Table 6.4, along with the predicted k, corresponding g_{conv} and N (to the nearest 10 solutions) for each FE considered.

Based on the values presented in Table 6.4, the Maximal Generation Functions are f1, f6, f8, f9, and f14. f8 and f14 do not have any structure in their search space to guide the optimisation process, as they have $R_{\text{av}} = 0$. This result is validated by the 2–dimensional plots of these functions, seen in Figure 6.14(b) for f8 and Figure 6.17(b) for f14, where the fitness function is very flat, and there is no structure to guide the GA toward better solutions. The fitness function statistics results for f1 and f9 suggest that these are separable functions with equal contributions from each decision variable to the fitness function value, which is verified by the equation for each of these fitness functions.



(a) 2–dimensional plot of f15 (b) 2–dimensional plot of f16

Figure 6.18 Functional form of f15 and f16.



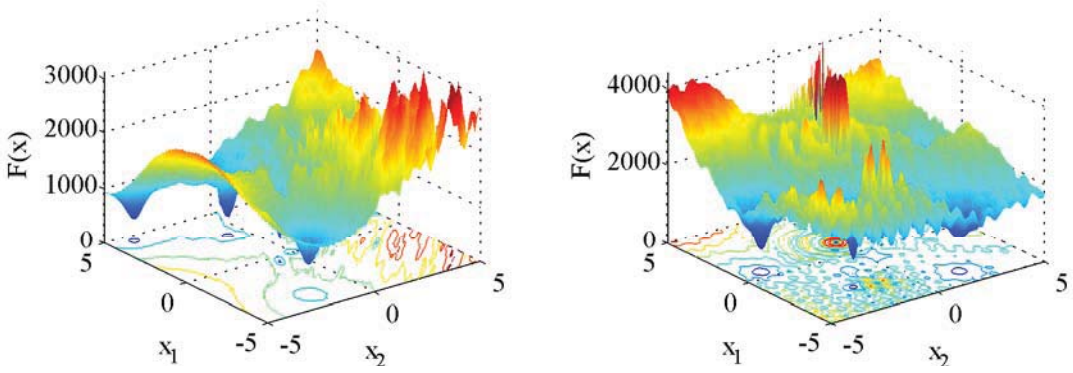
(a) 2–dimensional plot of f18 (b) 2–dimensional plot of f19

Figure 6.19 Functional form of f18 and f19.

The results presented in Table 6.4 suggest that f6 also falls into this class of fitness functions. However, f6 is Rosenbrocks Function with the optimal solution shifted in the search space. The function is:

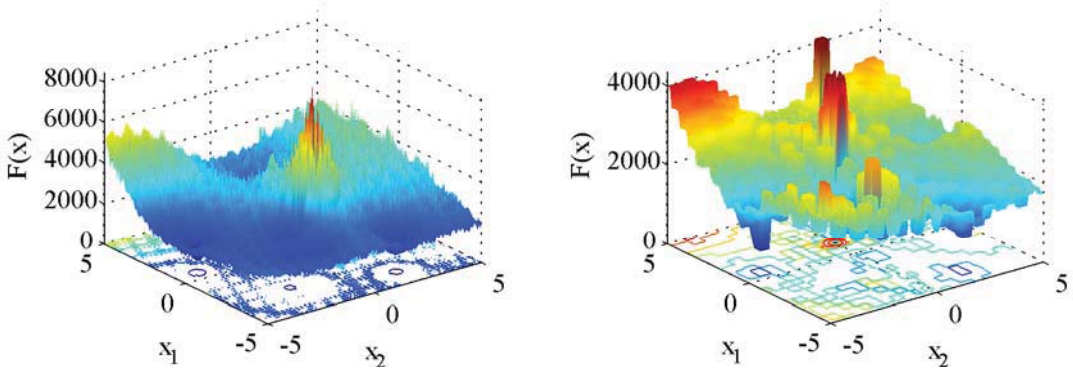
$$f_6(\mathbf{x}) = \sum_{i=1}^{l-1} \left(100\left(z_i^2 - z_{i+1}\right)^2 + \left(z_i - 1\right)^2\right), \tag{6.2}$$

where $z_i = x_i - o_i$, and o_i is value of x_i where the optimum occurs. If the summation



(a) 2–dimensional plot of f20 (b) 2–dimensional plot of f21

Figure 6.20 Functional form of f20 and f21.



(a) 2–dimensional plot of f22 (b) 2–dimensional plot of f23

Figure 6.21 Functional form of f22 and f23.

term is expanded, Rosenbrocks Function can also be written as:

$$\begin{aligned} f_6(\mathbf{x}) &= \sum_{i=1}^{l-1} \left(100\left(z_i^2 - z_{i+1}\right)^2 + (z_i - 1)^2\right), \\ &= \sum_{i=1}^{l-1} \left(100z_i^4 - 200z_i^2 z_{i+1} + 100z_{i+1}^2 + z_i^2 - 2z_i + 1\right). \end{aligned} \tag{6.3}$$

From Equation 6.3, it can be seen that most terms that make up f6 are if fact separable, it is only the $-200z_i^2 z_{i+1}$ term that involves a combination of the variables. As the statistics are computed over the whole range of the function, which for f6 is $x_i \in [-100, 100]$, the contribution to the fitness function value from the interacting terms will be relatively insignificant compared to that produced from the completely separable terms. The fitness

Table 6.4 Fitness function statistic values, and corresponding N for fitness functions with $l = 10$.

f	k_{BB}	m_{BB}	R_{av}	R_{l}	R_{T}	D	k	g_{conv}	N 10^3	10^4	10^5
1	1	0	0.2603	0.45	0.2603	0.0330	0.9389	-	10	10	10
2	6	6	0.2357	0.45	0.2357	0.0539	0.9433	117	10	90	860
3	4	8	0.2432	0.45	0.2549	0.0794	0.9365	104	10	100	960
5	3	12	0.2386	0.45	0.2386	0.0590	0.9428	116	10	90	860
6	1	0	0.2444	0.45	0.2482	0.0187	0.9452	-	10	10	10
8	10	1	0	0.05	0	0.0028	1	-	10	10	10
9	1	0	0.2092	0.45	0.2092	0.0364	0.9505	-	10	10	10
10	10	1	0.2474	0.45	0.2474	0.0593	0.9406	111	10	90	900
11	10	1	0.0595	0.25	0.0595	0.0030	0.9932	998	10	10	100
12	3	6	0.1830	0.45	0.1830	0.0480	0.9566	154	10	70	650
13	2	2	0.1780	0.45	0.2020	0.1790	0.9502	133	10	70	750
14	10	1	0	0.05	0	0	1	-	10	10	10
15	7	9	0.2233	0.45	0.2233	0.0561	0.9466	124	10	80	810
16	9	7	0.2257	0.45	0.2289	0.0569	0.9443	119	10	80	840
18	3	11	0.2556	0.45	0.2556	0.1414	0.9423	115	10	90	870
19	3	11	0.2496	0.45	0.2496	0.1308	0.9433	117	10	90	860
20	3	10	0.2542	0.45	0.2542	0.1293	0.9421	114	10	90	870
21	9	9	0.2482	0.45	0.2482	0.0860	0.9416	113	10	90	880
22	4	2	0.2426	0.45	0.2426	0.0393	0.9409	112	10	90	890
23	8	13	0.2370	0.45	0.2370	0.0870	0.9445	119	10	80	840

function statistics computed for f6 in $l = 10$ for different search space sizes can be seen in Table 6.5. The sampling has been centred around the global optimum solution, and the fitness function statistic values computed from ranges of ± 0.1, ± 1, ± 10 and ± 100 in each dimension are shown in Table 6.5. It can be seen that the fitness function characteristics are very similar when the fitness function values are sampled over the ranges ± 10 and ± 100 from the optimal solution. Similar correlation and dominance values are observed for a decrease in the search space size to ± 1 around the global optimum, however, for this smaller search space, the $m_{\text{BB}} = 9$ pair-wise interactions between the decision variables has been detected, with $k_{\text{BB}} = 2$.

In practice, the location of the global optimal solution is unknown, and therefore it is not possible to identify local changes to the fitness function characteristics, such as this, before the problem is solved. However, the fitness function statistics could be re-evaluated

Table 6.5 Fitness function statistic values for different sized search spaces for f6.

Range	k_{BB}	m_{BB}	R_{av}	R_{l}	R_{T}	D	k	g_{conv}
0.1	2	9	0.2022	0.4	0.3033	0.0270	0.8978	60
1	2	9	0.2102	0.4	0.2815	0.0374	0.9171	80
10	1	0	0.1936	0.4	0.2749	0.0444	0.9145	-
100	1	0	0.1870	0.4	0.3003	0.0453	0.8928	-

as the GA converged to smaller regions in the search space, and if the characteristics have changed, the GA parameter values could also be changed accordingly. It may be possible to make use of fitness function values that have already been evaluated by the GA, and therefore the computation effort involved in re-evaluating the fitness function characteristics would not be significant for computationally intensive fitness functions. While an approach such as this may be beneficial to the GA search, it has not been considered in this work.

Similar fitness function values were computed for the fitness functions with $l = 30$ and $l = 50$. The predicted values of k from Equation 5.9 for each function, corresponding values of g_{conv} calculated from Equation 5.1, as well as N rounded to the nearest 10 solutions for each FE, used as stopping criterion for each function, are presented in Table 6.6 for the functions with $l = 30$ and Table 6.7 for the functions with $l = 50$.

To compute the largest population size that will converge due to genetic drift, only the dimension of the problem and the number of function evaluations available are required, as given by Equation 6.1. The population sizes (rounded to the nearest 10 solutions) computed for each case considered in the comparison of the GA calibration methodologies are presented in Table 6.8. As the self-adaptive calibration method adds three more decision variables to the problem (p_m, p_c, and c) dimensions of $l = 13$, 33, and 53 are also included in Table 6.8. However, this slight increase in problem size did not influence the population size computed by Equation 6.1. The population sizes presented in Table 6.8 are much smaller than those computed from the fitness function statistics presented in Table 6.4, Table 6.6, and Table 6.7. This result indicates that it is predicted that the GA will converge much quicker due to the selection pressure than due to genetic drift, and therefore larger population sizes can be adopted to make more efficient use of the FE available to the GA.

Table 6.6 Predicted N for fitness functions with $l = 30$.

f	k	g_{conv}	N			
			10^3	10^4	10^5	3×10^5
1	0.9662	-	10	10	10	10
2	0.9681	230	10	40	440	1320
3	0.9646	200	10	50	490	1470
5	0.9679	230	10	40	440	1330
6	0.9707	-	10	10	10	10
8	1	-	10	10	10	10
9	0.9723	-	10	10	10	10
10	0.9667	220	10	50	460	1380
11	0.9961	1890	10	10	50	160
12	0.9755	300	10	30	340	1010
13	0.9733	270	10	40	370	1100
14	1	-	10	10	10	10
15	0.9700	240	10	40	410	1240
16	0.9688	230	10	40	430	1290
18	0.9687	230	10	40	430	1300
19	0.9691	230	10	40	430	1280
20	0.9684	230	10	40	440	1310
21	0.9676	220	10	40	450	1340
22	0.9666	220	10	50	460	1380
23	0.9692	240	10	40	420	1270

6.4.4 Overall Solution Quality Comparison

It should be noted that the function evaluations used to compute the fitness function statistics have not been considered in the function evaluations available to solve the problem. The fitness function statistics have been computed using $n = 5\,000$, and only need to be computed once for each function in each problem size. As each fitness function is solved 25 times for a number of different FE, the number of samples used to evaluate each function is insignificant compared to those used by the GA. For example, for a fitness function with $l = 50$ optimised 25 times for $FE = 10^3$, 10^4, 10^5, and 5×10^5, a total of $15\,275\,000$ fitness function evaluations were made.

The analysis methods proposed by Suganthan et al. (2005) to compare the results from different calibration methods for the test functions has been adopted in this work. For every GA calibration methodology tested, each function in each problem size for

Table 6.7 Predicted N for fitness functions with $l = 50$.

f	k	g_{conv}	N			
			10^3	10^4	10^5	5×10^5
1	0.9743	-	10	10	10	10
2	0.9756	310	10	30	320	1620
3	0.9731	280	10	40	360	1790
5	0.9755	310	10	30	330	1630
6	0.9781	-	10	10	10	10
8	1	-	10	10	10	10
9	0.9788	-	10	10	10	10
10	0.9745	300	10	30	340	1690
11	0.9970	2530	10	10	40	200
12	0.9813	400	10	20	250	1240
13	0.9800	380	10	30	260	1320
14	1	-	10	10	10	10
15	0.9771	330	10	30	300	1520
16	0.9761	320	10	30	320	1590
18	0.9764	320	10	30	310	1570
19	0.9766	320	10	30	310	1550
20	0.9761	320	10	30	320	1580
21	0.9753	310	10	30	330	1640
22	0.9744	290	10	30	340	1700
23	0.9766	320	10	30	310	1560

Table 6.8 Population Sizes Predicted due to Genetic Drift

l	FE				
	10^3	10^4	10^5	3×10^5	5×10^5
10	10	40	120	-	-
13	10	40	120	-	-
30	10	40	120	200	-
33	10	40	120	200	-
50	10	40	110	-	260
53	10	40	110	-	260

each stopping criterion has been run 25 times with different random number seeds. The difference in the solution found from the optimal solution for each fitness function for each of the 25 runs for each case considered have been sorted into increasing order, and the 1^{st} (Best), 7^{th}, 13^{th} (Median), 19^{th}, and 25^{th} (Worst) values found are presented in Appendix B, along with the mean and standard deviation of all 25 runs.

For each combination of l and FE tested, the performance of each GA calibration method over each of the 20 fitness functions has been ranked in order of decreasing ability to locate the best solutions. The ranking has been performed using a Student's t-test between the mean values found by the different methods, and if there is not a significant difference between two methods with a 95% confidence level, they have been given the same rank. The results from this analysis can be seen in Appendix C, where the different GA calibration methods have been sorted by their average rank over the 20 test functions used.

The average rankings for each case of FE and l considered in the comparative study, given in Appendix C, can be seen in Table 6.9. The GA calibration methods have been sorted by the mean of the average rankings over each case of FE and l considered, producing an overall ranking of the performance of the different GA calibration approaches considered.

For $FE = 10^3$, both the Predicted method and the Drift method use a population size of $N = 10$, therefore the results are the same for these two cases. It can be seen that overall, the proposed GA calibration methodology with set values for the GA parameter values performed best, with a mean ranking of 2.43. The genetic drift population sizing method also performed well, with a mean ranking of 2.79.

Not surprisingly, the Typical GA parameter calibration method performed worst, with a mean rank of 6.56, highlighting the importance of calibrating the GA parameter values for each individual problem. By allowing the GA to self-adapt the values for p_m, p_c and c, the typical GA parameter setting of $N = 100$ was able to improve the mean ranking of this method to 5.42.

The third best GA parameter calibration method overall was the Parameterless method with set values for the remaining GA parameters. However, with an overall rank of 4.11, it was consistently outperformed by the constant population size methods of Predicted and Drift. It can be seen from Table 6.9 that the Parameterless method with self adaptive parameters was the second worst performing calibration method, with a mean ranking of 5.48, which was even slightly outperformed overall by the constant population size with self adaptive parameters, with a ranking of 5.42.

Table 6.9 Overall rankings of the GA calibration methods

Parameter Setting	Mean	$FE = 10^3$ $l = 10$	10^3 30	10^3 50	10^4 10	10^4 30	10^4 50
Predicted - Set Values	2.43	1.90	1.98	2.52	1.95	3.00	3.27
Drift - Set Values	2.79	1.90	1.98	2.52	3.05	3.50	3.55
Parameterless - Set Values	4.11	3.10	2.67	2.38	4.75	3.60	3.05
Predicted - Self Adaptive	4.33	5.53	5.20	4.67	5.08	5.10	5.20
Drift - Self Adaptive	4.88	5.53	5.20	4.67	5.42	4.85	4.95
Typical - Self Adaptive	5.42	5.90	6.80	7.45	3.48	4.03	5.00
Parameterless - Self Adaptive	5.48	5.53	5.20	4.67	5.28	5.55	5.55
Typical - Set Values	6.56	6.62	6.97	7.10	7.00	6.38	5.42

Parameter Setting	Mean	FE =10^5 $l = 10$	10^5 30	10^5 50	3×10^5 30	5×10^5 50
Predicted - Set Values	2.43	3.52	1.85	1.70	2.55	2.48
Drift - Set Values	2.79	3.73	2.45	3.05	2.45	2.52
Parameterless - Set Values	4.11	4.75	5.15	5.00	5.10	5.65
Predicted - Self Adaptive	4.33	3.00	3.90	3.67	3.23	3.00
Drift - Self Adaptive	4.88	4.47	4.97	4.83	4.45	4.35
Typical - Self Adaptive	5.42	4.97	5.12	5.22	5.80	5.90
Parameterless - Self Adaptive	5.48	5.33	5.83	6.05	5.85	5.40
Typical - Set Values	6.56	6.22	6.72	6.47	6.58	6.70

Somewhat surprisingly, the calibration methods with set values for p_m, p_c and c consistent outperformed the corresponding methods with self-adaptive parameter values. The only exception to this was the typical GA parameter setting, where the ranking increased from 6.56 to 5.42 by allowing the parameter values to self adapt. This was most likely due to the probability of mutation of $p_m = 1$ (one mutation per solution) used for the typical setting, which was able to be reduced by the self-adaptive method when appropriate. This probability of mutation was also used for the Parameterless GA calibration method with set values, which may explain its relatively poor performance. However, if only the self-adaptive calibration methods are considered, and therefore the values of the GA parameters other than the population size and number of elite solutions are left to the GA, it can be seen from Table 6.9 that the Predicted and Drift methods still perform the best of the four methods used to determine the population size.

A typical convergence plot for the various GA calibration methods can be seen in

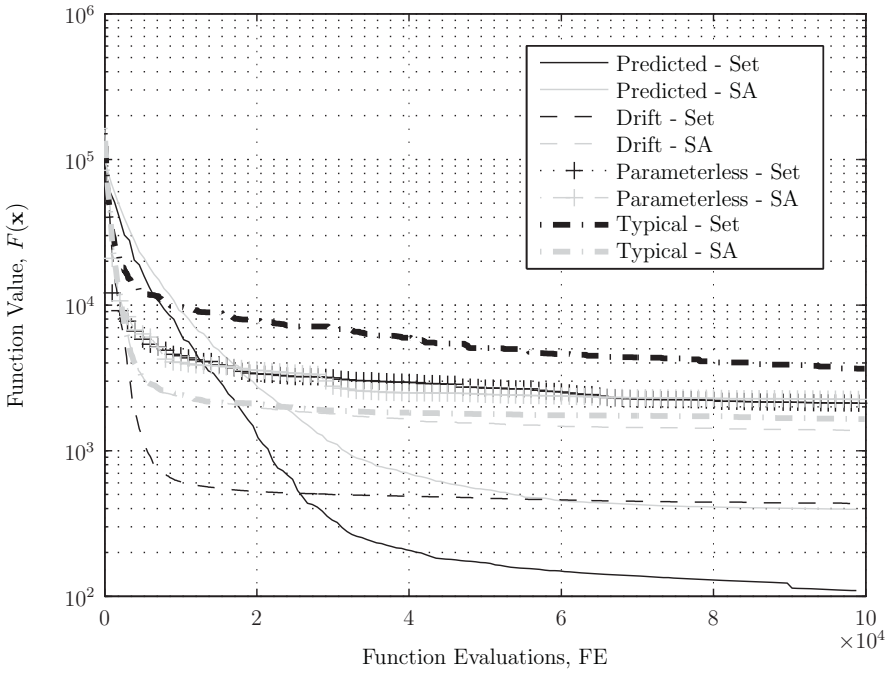


Figure 6.22 GA convergence for the different calibration methods, for f12 with $l = 10$

Figure 6.22. The plot shows the best solution found by each calibration method, averaged over the 25 independent GA runs undertaken, plotted against the function evaluations made, for f12 with $l = 10$. The Typical and Drift calibration methods with set parameter values have very similar population sizes of $N = 100$ and $N = 120$, respectively. The difference between these two convergence plots can be explained by the different p_m values used for each method, which were $p_m = 1$ and $p_m = 0.042$, respectively. From Figure 6.22, it can be seen that the Drift method with set parameters converged very quickly, but then is unable to find better solutions, while the Typical calibration method converged much more slowly but continued to slightly improve the solution found through the higher mutation rate. The results indicate that for f12, $p_m = 1$ is too disruptive, and the Drift method with $p_m = 0.042$ found a much better solution, averaged over the 25 GA runs. The effect of the different values of p_m is reinforced by the results for these two methods with self-adaptive parameters, which are very close when the GA is optimising the value of p_m along with the actual solution.

The Parameterless GA calibration method can be seen to converge very quickly in Figure 6.22, as in the first few thousand function evaluations very small population sizes of $N = 10$ and $N = 20$ are used. As the population size increased as the run progressed, the average best solution found from the 25 runs can be seen to still improve, however,

the rate of improvement is much slower. From Figure 6.22, it can be seen that there was very little difference between the self-adaptive and set parameter value methods for the Parameterless GA calibration method for f12. This may be due to the population size changing in accordance with the other parameters, reducing their influence on the solution found.

The convergence of the proposed GA calibration methodology can be seen in Figure 6.22 as the lines 'Predicted - Set' with set GA parameter values and 'Predicted - SA' with self-adaptive parameter values. This method adopts by far the largest population size of all the methods, with $N = 650$, and therefore can be seen to converge the slowest out of all the methods over the first $FE \approx 10^4$. However, there is more diversity in this larger population, and therefore the GA has the best chance of finding the best solutions during the permitted $FE = 10^5$. It can be seen from Figure 6.22 that the predicted method is still converging after the allowed $FE = 10^5$, however the most efficient part of the optimisation has already occurred. If more time was available to solve the problem, it would be expected that better solutions would be located with a larger population size, rather than relying on mutation to fine tune the decision variable values.

In comparison to the Predicted method, which was still locating better solutions after $FE = 10^5$, the population size determined from the Drift calibration method had more or less converged after $FE = 6 \times 10^4$. This result indicates that the GA has converged due to the selection pressure, rather than due to genetic drift, and therefore a larger population size than that predicted by the genetic drift model will converge in the FE available. The genetic drift method can be considered to provide a minimum bound on the most suitable population size to use, where even on a flat fitness function, the population can be expected to converge to a solution.

6.4.5 Function by Function Performance Comparison

The ranking of each of the GA calibration methods for each of the fitness functions, averaged over each case of FE and l considered, is provided in Table 6.10. It can be seen that the Predicted GA calibration method with set parameter values consistently ranks as the best approach, and if this calibration method did not have the best rank for a given function, it had the second best. The functions considered have a wide range of different properties, hence this result reinforces the belief that it is necessary to consider the characteristics of the fitness function in the calibration process, to provide the GA with the greatest chance of locating the best solutions.

The functions that the Predicted calibration method was outperformed on were f10,

Table 6.10 Average ranking of each GA calibration method for each fitness function.

Parameter Setting	1	2	3	5	6	8	9	10	11	12
Predicted - Set Values	1.6	2.3	1.8	2.4	1.3	3.7	1.1	3.0	1.9	1.4
Predicted - Self Adaptive	4.3	4.3	4.1	4.2	6.0	4.5	3.6	3.8	5.5	4.5
Drift - Set Values	1.7	2.3	2.2	3.1	2.2	4.1	4.2	1.7	1.2	1.9
Drift - Self Adaptive	5.1	6.0	5.4	6.0	4.9	4.8	4.2	4.2	4.4	5.2
Parameterless - Set Values	5.6	2.8	4.0	3.1	4.8	4.8	6.3	5.1	4.5	4.6
Parameterless - Self Adaptive	4.1	7.1	5.6	6.6	3.7	4.5	3.4	6.5	5.6	5.0
Typical - Set Values	7.8	5.6	7.1	4.9	7.4	4.4	7.9	7.4	7.9	7.9
Typical - Self Adaptive	5.8	5.6	5.7	5.6	5.7	5.2	5.4	4.4	5.0	5.6
Parameter Setting	13	14	15	16	18	19	20	21	22	23
Predicted - Set Values	2.8	2.1	3.4	2.6	3.2	3.2	3.1	2.5	2.2	3.0
Predicted - Self Adaptive	3.6	6.4	4.0	3.7	4.1	4.0	4.2	3.7	4.0	3.9
Drift - Set Values	2.2	5.0	3.0	1.9	3.5	3.3	3.3	3.4	2.0	3.5
Drift - Self Adaptive	4.8	2.8	4.0	4.7	5.8	5.7	5.5	4.5	5.2	4.7
Parameterless - Set Values	5.4	4.1	5.2	5.1	2.1	2.2	2.1	3.2	4.0	3.1
Parameterless - Self Adaptive	3.9	4.9	3.6	7.0	6.2	6.4	6.4	5.9	7.1	6.0
Typical - Set Values	7.6	6.7	7.8	6.4	5.0	5.3	5.3	6.5	6.0	6.3
Typical - Self Adaptive	5.7	4.0	5.0	4.7	6.0	5.9	6.1	6.3	5.5	5.5

f11, f15, f16, and f22 by the Drift method, and f18-f20 by the Parameterless method. In all cases, the best calibration methods adopted static values for the p_m, p_c and c parameters. Functions f15-f23 are the composite functions, composed from 10 simpler standard benchmark fitness functions. These functions have different characteristics in different regions of the search space, thus it is much harder for the fitness function statistics to accurately characterise these functions. Therefore, it is possible that the predicted number of generations before the population will converge based on the values of the fitness function statistics were accurate. However, the proposed GA calibration methodology does not perform poorly on these functions, and is the second best GA calibration method. It may be that better estimates of the local fitness function characteristics can be obtained for these more difficult functions from the smaller regions of the search space as the GA converges, similar to the case presented in Table 6.5 for f6, and the GA parameter values could be adjusted as more accurate values for the fitness function statistics were obtained.

The rankings presented in Table 6.10 were determined from a Student's t-test based on the mean and variance of the 25 GA runs for each GA calibration methodology for each case of each fitness function. Therefore, a calibration methodology achieved a higher

ranking than another method only if the difference was statistically significant. However, if one method happens to find some good solutions, as well as some poor solutions, over the 25 independent GA runs, a large variance is produced, and it is difficult to distinguish between the different calibration methods. In reality, a large variance is undesirable, as the solutions produced from one run to another can vary greatly, and it is unlikely that numerous GA runs will be undertaken to ensure the best solutions are identified in practice. The variances of the 25 GA runs for each GA calibration methodology for each function with $l = 50$ after $FE = 5 \times 10^5$ are presented in Table 6.11. It is not practical to combine the variances of different problems sizes or run times, as the variances are directly related to the fitness function values. However, the standard deviations for all GA calibration methods for all cases considered can be found in Appendix B.

The largest differences in the variance in the solutions found by the different GA calibration methods is produced for the completely separable functions of f1 and f9. By identifying the lack of epistatic interactions between the decision variables for these functions, the Predicted calibration method with set parameter values produced a much smaller variance in the best solutions found from run to run compared to the other methods. However, the drift calibration method produced the smallest variance for f1, of $\sigma^2 = 3.4 \times 10^{-9}$. The Predicted method also produced a much smaller variance for f6, which was incorrectly classified as a separable function, in comparison to the other GA calibration methods considered. Interestingly, by allowing the GA to self-adapt the GA parameter values for the predicted method for these functions, the variance in the best solutions found from run to run increased by many orders of magnitude. Again, this may be due to the increase in the number of decision variables, and therefore the GA is spending extra time determining the GA parameter values, as well as locating better solutions.

From Table 6.11, it can be seen that for the composite functions of f15-f23, generally the Predicted GA calibration method produced a variance at least an order of magnitude smaller than the other methods tested. For f21, the GA converged to the exact same solution for each of the 25 runs, producing a variance of $\sigma^2 = 0$, as seen in Appendix B. While this is a local optimum for f21, no other calibration method found a better solution, or found this solution as consistently. Therefore, while the proposed methodology only ranked as the second best calibration method for most of the composite functions, there was much less variance in the solution found from run to run. This result can be explained by the larger population size implemented for the Predicted GA calibration method, and therefore random effects of the initial population have a much smaller effect on the final solution found.

Table 6.11 Variance of best solutions found for $FE = 5 \times 10^5$ and $l = 50$.

Parameter Setting	1	2	3	5	6
Predicted - Set Values	4.7123e-07	3.1201e+06	1.6278e+13	7.2486e+04	2.1560e+04
Predicted - Self Adaptive	3.7330e+01	5.6365e+06	2.0568e+14	1.7595e+06	3.5156e+10
Drift - Set Values	3.4236e-09	7.7687e+02	4.0287e+12	2.7646e+05	4.0151e+06
Drift - Self Adaptive	4.0947e+02	5.4461e+07	1.4291e+15	9.0650e+06	8.2356e+12
Parameterless - Set Values	9.1283e+02	1.3640e+07	9.8510e+14	4.2687e+06	1.6390e+10
Parameterless - Self Adaptive	9.2710e+00	1.0693e+08	1.2347e+15	2.7513e+07	7.6852e+08
Typical - Set Values	8.2768e+04	2.0133e+07	6.8394e+14	1.9764e+06	6.7357e+13
Typical - Self Adaptive	1.3772e+04	8.6824e+07	1.1051e+15	6.2082e+06	2.6242e+08
Parameter Setting	8	9	10	11	12
Predicted - Set Values	1.9499e-03	4.0097e-06	1.3622e+02	4.3862e+00	6.9659e+08
Predicted - Self Adaptive	1.5594e-02	5.1415e+00	4.8478e+02	3.3601e+01	1.8251e+09
Drift - Set Values	2.3761e-03	1.1114e+01	1.4301e+04	7.0055e+00	2.5560e+08
Drift - Self Adaptive	9.2484e-03	3.5953e+01	2.8816e+03	3.5947e+01	4.3946e+09
Parameterless - Set Values	2.0872e-03	1.0523e+01	1.5950e+03	1.1275e+01	3.2356e+09
Parameterless - Self Adaptive	7.7551e-03	1.0265e+01	1.2666e+04	2.8046e+01	4.7468e+09
Typical - Set Values	8.6751e-04	2.7712e+02	7.0030e+02	1.6683e+01	3.8882e+09
Typical - Self Adaptive	1.3497e-02	2.7781e+01	8.5541e+03	3.0506e+01	4.6491e+09
Parameter Setting	13	14	15	16	18
Predicted - Set Values	2.1734e+00	2.6718e-01	9.9663e+03	7.3758e+01	3.1552e+01
Predicted - Self Adaptive	1.1709e+00	1.6538e-01	9.1518e+03	2.0924e+02	7.0041e+01
Drift - Set Values	6.3032e-01	3.5230e-02	7.6416e+03	1.4813e+04	2.0814e+02
Drift - Self Adaptive	1.3860e+01	3.1403e-01	8.2943e+03	4.8374e+03	4.4761e+02
Parameterless - Set Values	6.4822e+00	8.6361e-02	1.7790e+04	1.3031e+04	2.8669e+02
Parameterless - Self Adaptive	1.1396e+01	1.7516e-01	3.4919e+04	1.0807e+04	2.4778e+03
Typical - Set Values	1.2956e+02	5.1124e-02	1.4621e+03	1.5866e+03	2.0592e+02
Typical - Self Adaptive	1.0911e+01	3.2034e-01	7.2611e+03	1.1255e+04	8.9379e+02
Parameter Setting	19	20	21	22	23
Predicted - Set Values	2.8633e+01	2.0747e+01	0.0000e+00	7.5073e+01	4.5299e-04
Predicted - Self Adaptive	6.0706e+01	4.8811e+01	7.1185e-01	1.8900e+02	2.5669e-01
Drift - Set Values	9.8019e+02	9.8507e+02	1.7601e+04	2.2085e+02	6.5716e-04
Drift - Self Adaptive	3.1710e+02	3.2217e+02	1.8484e+04	1.2296e+03	1.6283e+04
Parameterless - Set Values	1.8026e+02	2.4262e+02	4.2799e+04	2.8759e+02	1.6766e+04
Parameterless - Self Adaptive	2.3726e+03	2.6877e+03	1.1955e+05	1.2544e+03	8.0068e+04
Typical - Set Values	5.3872e+01	3.7060e+01	5.0876e+02	1.8242e+02	4.7905e+03
Typical - Self Adaptive	1.2393e+03	1.0699e+03	1.2430e+05	2.1819e+03	9.7277e+04

6.5 DISCUSSION

Firstly, it should be noted that the No Free Lunch Theorem (Wolpert and Macready, 1997) does not apply to the comparison presented in the previous section. If the objective was to find the best set of GA parameter values for any function, for example if eight different sets of GA parameter values were compared, rather than eight different methods to determine the GA parameter values, then it would be expected from the No Free Lunch Theorem that when the performance of each parameter set was averaged over every function, and every combination of l and FE considered for each function, that the sets of parameter values would perform similarly. However, in this study, the proposed GA calibration methodology identifies information about the fitness function, and makes use of it to give the selected algorithm the best chance of solving the problem. Therefore, the No Free Lunch Theorem does not apply, and meaningful results are obtained.

From Table 6.9, it can be seen that generally the 'typical' GA parameter values performed the worst. The typical GA parameter values used were the same for each function, where all other methods considered had a mechanism to change the GA parameter values with the function characteristics; either by increasing the population size with FE or l (Drift method), increasing the population size only after a smaller population size had converged (Parameterless method), or by relating the population size directly to the characteristics of the function (Predicted). Therefore, it is not surprising that this method performed worst overall, as the GA parameter values were not tuned to each fitness function, providing different GA behaviour for the different functions.

Equation 6.1 provided the largest population size that can be expected to converge due to genetic drift for a given FE and l. However, generally the population will converge due to the selection pressure before it will randomly converge due to genetic drift. Convergence due to selection pressure is much more difficult to predict, as the convergence rate is dependent on the fitness function characteristics. The GA calibration methodology proposed in this thesis has quantified the convergence due to the selection pressure, based on the characteristics of the fitness function.

The population sizes produced by the Predicted calibration methodology are much larger than the population sizes determined from the genetic drift model in Equation 6.1. The differences in the population sizes predicted from these two methods can be explained by the different approaches. By considering these population sizes in terms of the number of generations before convergence, the results indicate that the GA will converge due to the selection pressure before it will converge due to genetic drift. This is an expected result, as for a flat fitness function, the convergence will be due to genetic drift, as the

fitness function value has no influence on the solutions chosen during selection. However, for a function with any structure, it would be expected that the fitness function values would cause the GA population to converge quicker than what will occur randomly due to genetic drift. Therefore, as fewer generations are available before the GA population will converge, the Predicted method can use a larger population size, and thus generally find better solutions.

From Table 6.9, it can be seen that generally the static population sizes determined from both the Predicted and Drift methods outperformed the Parameterless GA calibration method. Eiben et al. (2004) also found that the Parameterless GA was much slower than a traditional GA with a constant population size on a number of multi-modal problems. From Figure 6.22, it can be seen that Parameterless method causes the GA to converge quickly, both with and without self-adaptation, but then the rate in the improvement in fitness function value is much slower after the first few thousand function evaluations. This may be due to starting the GA with a small population size and quickly converging to a local optimum, and then by injecting this local optimum into the population for larger population sizes, the GA is lead back toward this local optimum in the first few generations, and therefore potentially away from better regions in the search space.

This result suggests that in general, one GA run with a large population size can identify better solutions than multiple restarts of smaller population sizes for the same FE. Other studies (Cantú-Paz and Goldberg, 2003, for example) have also come to this conclusion. The work presented in this thesis has put a value on how large is 'large', by making use of the fitness function characteristics to identify the population size that can be expected to come close to converging in the FE available. Therefore, fitness function evaluations are not wasted by the GA population prematurely converging, but at the same time the population will converge close to one solution, allowing the best possible solution to be found in the time available.

This work has not looked in depth at the possibility of adaptive population sizes. Methodologies to self-adapt the population size are available (Arabas et al., 1994; Bäck et al., 2000; Eiben et al., 2004), however these approaches replace the population size with one or more parameters to control the change in population size, and therefore contribute to the GA calibration problem. The values for the fitness function statistics for f6 over different search space sizes, presented in Table 6.5, indicate that for this function, the interaction between the decision variables is only significant after the GA has converged to a region in the search space close to the optimal solution. Once the GA converged to this region, it may be beneficial to increase the population size, providing the GA with a

greater chance of processing the interactions between decision variables. Therefore, by computing the fitness function statistics as the GA converges to smaller regions of the search space, the population size can be adapted to suit the local fitness function characteristics. A GA calibration approach such as this would produce an adaptive population size, without introducing more parameters to be calibrated.

It may be that the deterministic rule used to increase the population size for the Parameterless GA calibration method is equivalent to the deterministic rules that are available to control the other GA parameters, such as those proposed for p_m outlined in Section 2.4.4.1. In the case of deterministically adapting the value for p_m, the value is changed without any consideration of the actual progress in solving the problem, and determining the best form of the relationship to change the value is just as difficult to determine as the best static value for the parameter. This is not quite the case for the Parameterless GA calibration method, where some feedback from the GA progress is used to control the population size, as the population size is only increased after a smaller population size has converged. However, the change in population size is deterministic, which may not be the most appropriate change for the fitness function being optimised. For example, f15 is non-separable over the whole search space, however, near the global optimum, the function is composed from the Rastrigin Function, and therefore is separable in all decision variables. For this case, once the GA has converged to this region in the search space, it may be beneficial to decrease the population size, not increase it, as there are no longer interactions between the decision variables to be processed by the GA.

Eiben et al. (1999) suggested that self-adaptive parameter control methods were the most promising method to determine the most suitable GA parameter values. However, the results presented in this chapter indicate that the set parameter values for p_m, p_c and c outperformed the self adaptive methods. A similar result was found by Bäck et al. (2000), who concluded that the performance of self-adapting the parameters p_m and p_c was disappointing when used on its own (i.e.; without any control of the population size). The most likely reason for this is that time spent on searching for good parameter values is time taken away from finding the optimum (Bäck et al., 2000). While a self-adaptive approach provides a mechanism for the GA parameters to adjust to more suitable values as the GA converges to smaller regions of the search space, it is unlikely that the GA will find better values for the parameters through mutation alone after they have converged to an initial value.

It can be seen from Table 6.4, Table 6.6, and Table 6.7 that, for most of the fitness functions considered that were Optimal Generation Functions, the predicted value of k

does not change very much from function to function, and therefore neither does the population size used to solve the problem for each FE considered. This result implies that if a function is known to be an Optimal Generation Function, it may be possible to generalise the value of k to determine the best population size. In reality, it may be reasonable to assume that the types of optimisation problems considered will be structured in some way, and will have a number of epistatic interactions between the decision variables, as well as potentially at least one highly salient variable, classifying the function as an Optimal Generation Function. If a generalised function for the value of k can be determined, similar to that for the genetic drift equation presented in Equation 6.1, then the complex function characterisation can be avoided, and a rough estimate for the population size could still be obtained. However, a study such as this is beyond the scope of this thesis.

The population size predicted to converge due to genetic drift is not related to the fitness function, and therefore is easy to determine by solving the implicit equation in Equation 6.1. This approach to determining the GA population size was a close second to the Predicted method, with a ranking of 2.79 compared to 2.43, and in all but three of the cases where the Predicted method did not perform best in Table 6.10, the Drift method was the best. Therefore, as this approach to determining the population size is much easier than the Predicted method based on the statistics of the fitness function, the Drift method may be suitable in many cases were the absolute best solution is not desired.

6.6 SUMMARY

The first section of this chapter has investigated the relationships between the best values for the GA parameters, leading to a full GA calibration methodology. The best performing GA parameter values, including interaction between the parameter values, were identified from the results of the large scale parametric studies undertaken in Chapter 3. Based on observations from these data outlined in Section 6.1, Section 6.2 has proposed a full GA calibration methodology, based on the characteristics of the fitness function. A simple GA calibration method based on convergence due to genetic drift was also proposed in Section 6.3.

Section 6.4 has tested the proposed GA calibration methods against a number of other GA calibration methods that are currently available. The calibration methods have been compared over a wide range of functions, tested for different problem sizes and convergence criteria. The results presented in Table 6.9 indicate that the proposed GA calibration methodology produced the best overall results of the eight different GA calibration methods considered, while the typical GA parameter values produced the worst results. These

findings highlight the importance of calibrating the GA parameter values to the characteristics of the fitness function. The calibration methodology proposed in this chapter directly relates the best value of the GA parameters to the characteristics of the fitness function, as determined by the fitness function statistics developed in Chapter 4.

Both this GA calibration method, and the calibration method based on determining the largest population size that will converge before genetic drift occurs, outperformed the Parameterless GA calibration method. The genetic drift method can be considered to provide a minimum bound on the most suitable population size to use, where even on a flat fitness function, the population can be expected to converge to a solution. This approach proved to provide a simple alternative to determine a suitable population size to use in many of the cases considered.

The results suggests that, in general, one large population size has a greater chance of locating better solutions than a number of restarts of smaller populations for the same number of function evaluations. The proposed GA calibration methodology determines a value for how large is 'large', ensuring that the GA converges to the best possible solutions, but at the same time preventing the GA population from prematurely converging and relying on the inefficient mutation operator to identify better solutions.

The results presented in Table 6.9 indicate that self-adaptive values for p_m, p_c and c were significantly outperformed by the use of static values for these parameters over the entire GA run. It might be expected that a self adaptive approach would produce better final solutions, as it provides a mechanism for the GA parameter values to adjust to more suitable values as the GA converges to smaller regions of the search space. However, the self adaptive approach produced more decision variables to be solved by the GA, and therefore the most likely reason for the poor performance of the self adaptive method is function evaluations that are spent on searching for good parameter values are taken away from finding better solutions, when compared to a GA with the same population size and fixed values for the remaining GA parameters.

Chapter 7
Application to WDS Optimisation

A full GA calibration methodology was developed in Chapter 6, and tested against other methods to determine the GA parameters on a number of complex test functions. The test functions used were designed to present a realistic test for the GA, however the functions consisted of mathematical functions that cannot represent the complexities involved in the optimisation of WDS. In this chapter, a similar comparison of the GA calibration methods considered in Chapter 6 is undertaken, however in this chapter the fitness functions used for the comparison are actual WDS optimisation problems.

Two different WDS problems have been considered in the study. Both studies consider the optimal operation of a WDS, to minimise the costs involved in operating the system. The first is a network that has been optimised by a number of different approaches in the past, the Cherry Hill-Brush Plains network from the USA. The second network considered for the GA calibration comparison is the Woronora WDS, located in Sydney, Australia. In this case, the optimisation problem is to determine the most cost effective operation of the pumps and valves in the network, while also considering the water quality produced by the operation of the WDS.

The eight GA calibration methods have been tested for different stopping criteria for each problem. The only difference to the methodology used in Chapter 6 is that the number of different sequences of random numbers used for each method has been reduced from 25 to 13, due to the computational effort of running the WDS simulation model. For all the analyses undertaken in this work, the simulation model used was EPANET, version 2.00.10 (Rossman, 1994).

In the following section, the comparative study is undertaken on the Cherry Hill-Brushy Plains network. Firstly, the network is presented, then the fitness function to be optimised. This is followed by a results section, consisting of the characterisation of the fitness function, a comparison of the different GA calibration methods for different convergence criteria, and a description of the best solution found by the optimisation.

The second section of this chapter presents a similar analysis for the Woronora WDS, before a discussion of the results and concluding remarks are made.

7.1 CHERRY HILL-BRUSHY PLAINS NETWORK

The Cherry Hill-Brushy Plains portion of the South Central Connecticut Regional Water Authority network, USA, is the first network used to test the GA calibration methods on WDS optimisation problems. This network has been used to validate and test different water quality models, and has been optimised by a number of methods, including linear programming (Boccelli et al., 1998), a linear least-squares formulation (Propato and Uber, 2004) and a GA (Munavalli and Kumar, 2003). Hence, this network provided a good test case for the comparison of the GA calibration methods, as well as to compare the GA used in this work against other optimisation methods that have been applied in the WDS optimisation field. To assist in the comparison, first-order bulk decay kinetics have been used to model the chlorine decay in the system, as this assumption is required for the linear optimisation methods that have previously been implemented. However, one of the benefits of the GA approach is that the optimisation proceeds with information directly from the simulation model, and therefore more complex kinetics, for both bulk and wall decay, could have been implemented just as easily.

7.1.1 System Description

The data for the system are the same as those used by Boccelli et al. (1998), and the network layout can be seen in Figure 7.1. The link and node data, including demands and pipe lengths can be found in Boccelli et al. (1998). The water source is a pump station, represented by a negative demand at node 1. The pump station is switched on for the first six hours of the simulation. After this time, the pumps switch off, and the demands in the system are met by the water stored in the reservoir at node 26. After the first 12 hours of the simulation, the pump station switches back on for another six hours, allowing the reservoir to be refilled with water from the pumping station. For the final six hours of the day the pump station is off, and again, the demands in the system are met from the storage in the reservoir.

The hydraulics of the system are fixed, hence the optimisation problem is to determine the mass of chlorine to be dosed at each of the six dosing points, for each of the four 6-hour time periods over a day. The sources are represented by nodes A, B, C, D, E, and F in Figure 7.1. Therefore, when the pumping station is on, source node A provides the

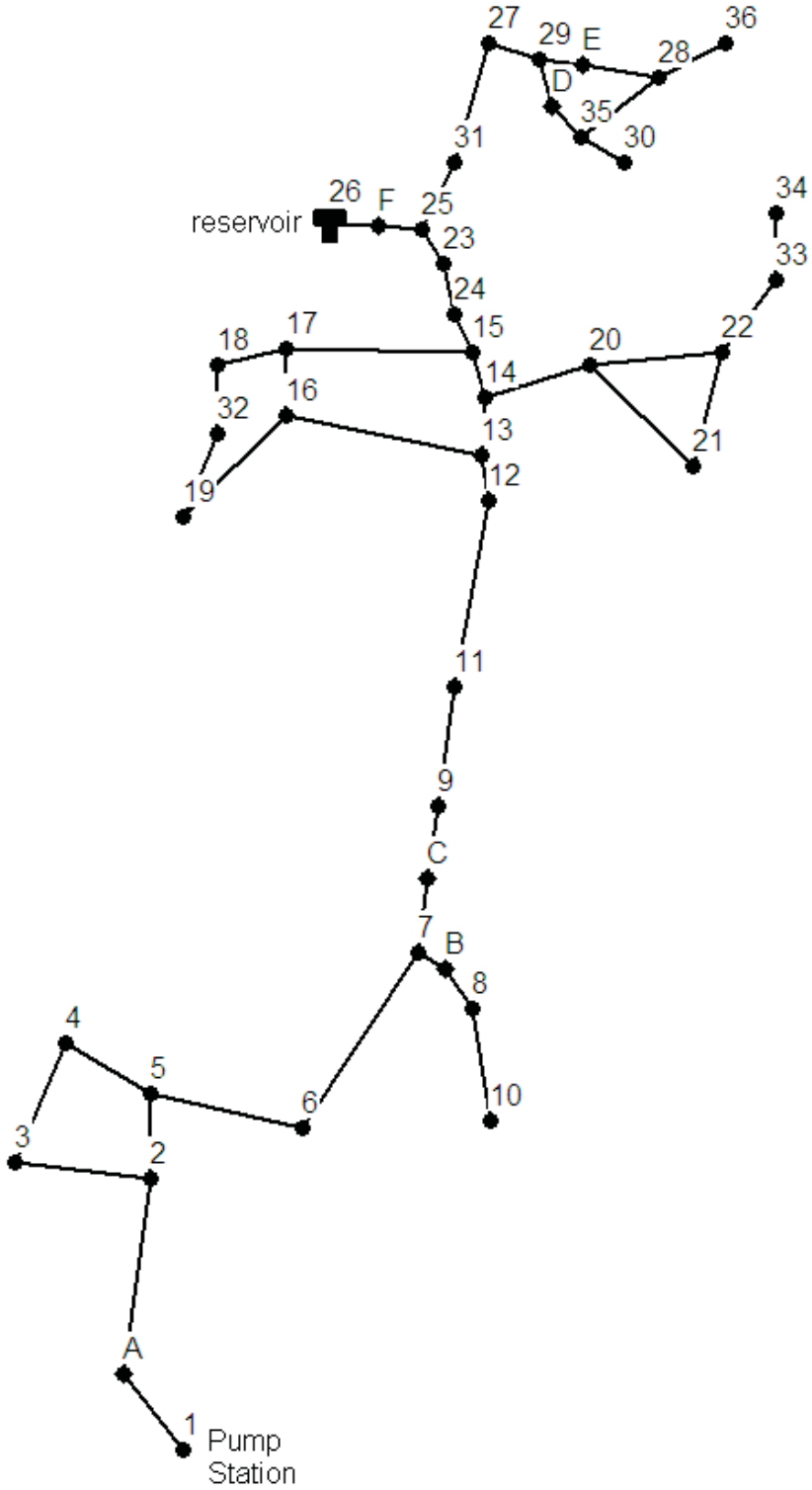


Figure 7.1 Schematic of the Cherry Hill-Brushy Plains Network

initial concentration of chlorine, and the concentration can be boosted at node C before reaching the reservoir. The chlorine concentration of water coming out of the reservoir can be increased at node F, and nodes B, D, and E can be used to boost the concentration of chlorine in the water in their respective parts of the network.

For the time periods when the pump station is off, the flow along the trunk main from node 1 to the reservoir at node 26 will reverse, and the demands in the network will be met from water in the reservoir. Therefore, for this case, the water supplying the demands in the bottom section of the network will have had a much longer detention time, as the water has travelled from the pump station to the reservoir, been stored in the reservoir for a number of hours, and then travelled back to the demand nodes. When the pump station is off, the concentration of chlorine can be increased at nodes F and C, while nodes B, D, and E are used to top up the concentration of chlorine at the relevant nodes.

To remove the influence of the initial chlorine concentrations in the network, the simulation was run until a 24 hour repeating pattern was observed in the chlorine concentrations at the demand nodes. Generally, approximately a two week simulation was required to reach the steady state condition in the chlorine concentrations.

7.1.2 Cherry Hill-Brushy Plains Fitness Function

The objective of the Cherry Hill-Brushy Plains optimisation problem is to minimise the mass of chlorine added to the system over a 24 hour period. The decision variables are the mass of chlorine injected at each dosing point, in mg/min, for each of the four six-hour time periods when the hydraulics of the system change. This produces a total of 24 decision variables to be determined by the GA. The range considered for each decision variable was between 0 and 800 mg/min.

The constraint on the system is to maintain the chlorine concentrations in the network in the acceptable range of 0.2–4.0 mg/L. A penalty of 100 $(\text{mg/L})^{-1}$ multiplied by the difference between the actual and required chlorine concentrations was applied at any violating nodes at each 10 minute time step of the simulation.

7.1.3 Cherry Hill-Brushy Plains Results

The eight GA calibration methods considered in Chapter 6, namely Predicted, Drift, Parameterless, and Typical, with both set and self-adaptive parameter values, have been applied to the fitness function for the Cherry Hill-Brushy Plains Network. Due to the computation requirements of repeatedly running the simulation model, the number of GA runs with different random seeds for each calibration method was 13, as mentioned pre-

viously. The fitness function was optimised for stopping criteria of $FE = 5 \times 10^3$, 10^4, and 10^5. For 1 GA run, these stopping criterion correspond to approximately 15 minutes, 30 minutes, and 5 hours before a solution is required, respectively, on a 2.2GHz workstation with 1GB RAM running the Linux operating system. As each of the 8 calibration methods have been run 13 times with different random numbers, this equates to a total of 24 days of CPU time required to perform the comparison of GA calibration methods on the Cherry Hill-Brushy Plains fitness function.

7.1.3.1 Fitness Function Characterisation

The three fitness function statistics developed in Chapter 4 have been applied to the Cherry Hill-Brushy Plains fitness function over the defined search space of 0–800 mg/min for each decision variable. A sample size of $n = 5000$ has been used to compute each statistic. For the separability measure computed over the whole range of the search space, the results suggested that there were no epistatic interactions between the decision variables. This was a somewhat surprising result, as it would be expected that for most combinations of dosing points, the chlorine added at at one location would have an influence on the best chlorine to be dosed at another location. However, as the fitness function consisted of a simple summation of the chlorine added at each dosing point at each time period to compute the total mass of chlorine injected into the system over a day, the fitness function is completely separable in the decision variables for all feasible solutions. Therefore, there will only be an interaction between the decision variables if two solutions are compared that violate the constraints by different amounts, therefore producing an extra penalty term in the fitness function value.

If a solution is randomly sampled over the whole range of dosing rates considered in this study, up to 800 mg/min, it is unlikely that the concentrations in the network will be below the minimum constraint of 0.2 mg/L, based on the flows observed in the system. Also, when all the dosing points in the network are set to the maximum of 800 mg/min, the only nodes that violate the upper constraint are downstream of node B and downstream of nodes D and E, as well as 5 nodes between nodes F and C for one 10 minute time step over a 24 period. Therefore, for most of the randomly sampled solutions between 0 and 800 mg/min, it is also likely that the upper constraint will not be violated, and hence generally only feasible solutions will be randomly generated, which are completely separable in all decision variables.

In an attempt to identify the expected interactions between infeasible solutions in the search space, the separability measure has been computed over a smaller range of the

search space. It would be expected that the interaction between decision variables would be most prevalent around the optimal solution, as generally the optimal solution occurs on the boundary between the feasible and infeasible solutions. A near optimal solution for the Cherry Hill-Brushy Plain network could be obtained from other optimisation studies that have considered this problem, such as Boccelli et al. (1998), Propato and Uber (2004), or Munavalli and Kumar (2003). However, in order to represent a more realistic situation where the optimal solution is unknown, the search space has been sampled over the range 0–50 mg/min for each dosing point to compute the separability measure.

Each pair of chlorine dosing points in the system have been considered together to compute the separability measure. The sampling for each interaction between the dosing points has been undertaken randomly over all of the time steps, producing an overall interaction between the two dosing points. This approach has the advantage of decreasing the number of pairs of decision variables to be sampled, and therefore drastically reduces the computation effort required to determine the separability measure. However, the disadvantage is that the interactions between the variables in time are lost. For example, if the chlorine dosed at node A in the first six hours produced an epistatic interaction with the chlorine dosed at node F in the second six hour period. As the number of pairs of interactions to compute is given by $l(l-1)/2$, the computation requirements to compute the separability measure increase very quickly with problem size. This is a major disadvantage of the separability measure, especially for computationally intensive fitness functions typical of WDS optimisation problems.

The interactions between each pair of decision variables computed by the separability measure over the smaller region of the search space are given in Table 7.1. The range of the separability measure is $[0 \leq \lambda \leq 1]$, where $\lambda = 1$ indicates a very strong interaction between the decision variables, and $\lambda = 0$ indicates the decision variables are completely separable. The results presented in Table 7.1 indicate that the interaction between the decision variables was still not strong, with the strongest interaction being between nodes C and F with $\lambda_{C,F} = 0.177$. This result can be explained by the layout of the network, given in Figure 7.1. When the pumping station is on, the chlorine dosed at node C will be topped up by the chlorine dosed at node F as the water drains out of the reservoir, while when the pumping station is off, the water draining down to the bottom of the network will first be dosed by node F, and then topped up by node C. Chlorine dosed by these two nodes will have an influence on the chlorine concentrations over a wide range of the network, and therefore will be most likely to produce a change in the feasibility of the solution, and thus an interaction between the decision variables.

Table 7.1 Interactions

	B	C	D	E	F
A	0.053	0.147	0.064	0.065	0.071
B		0.146	0.042	0.000	0.100
C			0.153	0.088	0.177
D				0.029	0.066
E					0.132

The significant interactions in the network seen in Table 7.1 were between nodes A and C, B and C, C and D, C and F and E and F. All of these interactions have very similar values of the separability measure of $\lambda \approx 0.15$. The results indicate that the chlorine dosed at node C is very important, as it is involved in all but one of the interactions detected between the decision variables by the separability measure. Based on these results, it was decided to arrange the solution string in the order ABCDEF, to allow the most effective processing of the interactions. With the solution string ordered in this way, all the interactions are between decision variables one or two positions apart in the solution string, with the only exception being nodes C and F. However, by moving these variables closer together in the solution string, other interactions would have to be separated.

The spatial correlation measure and the dominance measure have also been applied to the Cherry Hill-Brushy Plains fitness function. The values determined by the statistics were:

- $R_{\text{av}} = 0.2434$,
- $R_{\text{l}} = 0.45$,
- $R_{\text{T}} = 0.2434$,
- $D = 0.0407$.

Therefore, the fitness function is an Optimal Generation Function, as the search space is structured with a high R_{av} and R_{l} and there were a number of interactions between the decision variables over the smaller search space. The dominance measure suggested that there were no salient decision variables, as $D < 0.5$. This result can also be explained by the construction of the fitness function, where as the fitness function is computed as a sum of each of the dosing rate decision variables, each decision variable had a similar contribution to the fitness function value.

From these values of the spatial correlation and dominance measures, the value of k computed from Equation 5.9 was $k = 0.9626$. Based on a problem size of $l = 24$, the

number of generations before the population will converge on this function was predicted to be $g_{\text{conv}} = 190$ by Equation 5.1. Therefore, for the Predicted calibration methodology, the population sizes used for the different convergence criteria were $N = 30$ ($FE = 5 \times 10^3$), $N = 50$ ($FE = 10^4$), and $N = 530$ ($FE = 10^5$).

For the Drift GA calibration method, the population size was determined by Equation 6.1, based on l and the different cases of FE considered. Therefore, for the Cherry Hill-Brush Plains Network, the population sizes used by the Drift calibration method were $N = 30$ ($FE = 5 \times 10^3$), $N = 40$ ($FE = 10^4$), and $N = 120$ ($FE = 10^5$). For the shorter convergence times of $FE = 5 \times 10^3$ and 10^4, it can be seen that both the Predicted and Drift methods adopt very similar population sizes. However, for $FE = 10^5$, the population size predicted by the Drift method is much smaller than that computed by the Predicted method.

7.1.3.2 Comparison of GA Calibration Methods

A similar comparison of the GA calibration methods provided in Chapter 6 has also been applied to the Cherry Hill-Brushy Plains fitness function. The only difference between the methodology used in the previous chapter and the methodology used for the WDS fitness functions was that the number of times each calibration method was repeated with different random seeds was reduced from 25 to 13 to reduce the computational requirements, as mentioned previously.

The solutions found by the different GA calibration methods for the three different stopping criteria considered are given in Table 7.2 for the calibration methods with set values for p_m, p_c, and c, and in Table 7.3 for the calibration methods with self adaptive p_m, p_c, and c parameter values. It can be seen that, overall, the solution quality found by the different methods does not vary greatly, with the solutions generally being of the same order of magnitude for the same convergence criteria across all methods.

The rankings of each of the calibration methods for the different stopping criteria can be seen in Table 7.4, and the different approaches have been sorted by by their mean ranking over the different stopping criteria. Overall, it can be seen that the Predicted and Drift methods with set parameter values performed the best, with no significant difference between the two methods. As was the case in Chapter 6, the typical GA parameters with set values performed worst.

The population size for the Predicted and Drift methods are the same for $FE = 5 \times 10^3$ ($N = 30$) and very similar for $FE = 10^4$ ($N = 50$ and 40, respectively). Hence, the results produced from these two methods for these stopping criteria are very similar. How-

Table 7.2 Results for different GA calibration methods with set parameters

FE		Predicted	Drift	Parameterless	Typical
5e3	1st(Best)	1.6127e+03	1.6127e+03	1.4576e+03	1.1834e+04
	4th	2.0348e+03	2.0348e+03	2.0123e+03	1.7449e+04
	7th(Median)	2.6318e+03	2.6318e+03	5.9917e+03	3.4988e+04
	10th	3.3910e+03	3.3910e+03	1.0991e+04	4.2695e+04
	13th(Worst)	7.5492e+03	7.5492e+03	6.1467e+04	6.0100e+04
	Mean	3.4469e+03	3.4469e+03	1.4315e+04	3.3091e+04
	Stdev	1.9433e+03	1.9433e+03	1.9026e+04	1.5169e+04
1e4	1st(Best)	1.3083e+03	1.2819e+03	1.4027e+03	3.4600e+03
	4th	1.4264e+03	1.3925e+03	1.4751e+03	5.2657e+03
	7th(Median)	1.5347e+03	1.4694e+03	1.5520e+03	8.9233e+03
	10th	1.8437e+03	2.9314e+03	1.5758e+03	1.1731e+04
	13th(Worst)	9.2051e+03	1.1573e+04	1.6621e+03	1.7421e+04
	Mean	2.2938e+03	3.1520e+03	1.5311e+03	9.1438e+03
	Stdev	2.0499e+03	3.1674e+03	7.1782e+01	4.2800e+03
1e5	1st(Best)	1.1871e+03	1.1686e+03	1.2513e+03	1.5061e+03
	4th	1.1901e+03	1.1900e+03	1.2918e+03	1.5484e+03
	7th(Median)	1.1934e+03	1.1963e+03	1.3114e+03	1.6093e+03
	10th	1.1978e+03	1.2028e+03	1.3371e+03	1.6421e+03
	13th(Worst)	1.2330e+03	1.2247e+03	1.3770e+03	1.7527e+03
	Mean	1.1965e+03	1.1957e+03	1.3184e+03	1.6159e+03
	Stdev	1.1223e+01	8.1207e+00	3.6939e+01	8.0747e+01

ever, by increasing the population size to $N = 100$, used by the Typical GA calibration method, the mean value found was much higher, being 10 times higher for the stopping criteria of $FE = 5 \times 10^3$, and three times higher for the stopping criteria of $FE = 10^4$.

Very similar solutions are found by all GA calibration methods for the longest stopping criterion of $FE = 10^5$. This is somewhat surprising, as the Drift and Predicted methods used very different population sizes of $N = 120$ and $N = 530$, respectively. The Typical and Drift methods were used with very similar population sizes for $FE = 10^5$, however the poorer performance of the Typical method can be attributed to the higher mutation rate adopted.

The Parameterless method performed similarly to the Predicted and Drift methods for $FE = 10^4$ and 10^5. The Parameterless method was slightly worse for $FE = 10^5$, and from Table 7.4, it can be seen that the difference was significant at a 95% confidence level.

Table 7.3 Results for different GA calibration methods with self adaptive parameters

FE		Predicted	Drift	Parameterless	Typical
5e3	1st(Best)	1.4122e+03	1.4122e+03	1.4451e+03	3.4175e+03
	4th	2.5579e+03	2.5579e+03	2.1777e+03	8.5085e+03
	7th(Median)	1.5385e+04	1.5385e+04	6.0830e+03	2.9381e+04
	10th	2.3368e+04	2.3368e+04	1.8467e+04	4.7575e+04
	13th(Worst)	3.3924e+05	3.3924e+05	4.5227e+04	6.6770e+04
	Mean	5.9970e+04	5.9970e+04	1.3643e+04	2.9792e+04
	Stdev	1.0362e+05	1.0362e+05	1.4309e+04	2.1112e+04
1e4	1st(Best)	1.3853e+03	1.3504e+03	1.3098e+03	1.4260e+03
	4th	3.1609e+03	1.4318e+03	2.1107e+03	1.7058e+03
	7th(Median)	5.3863e+03	1.4669e+03	5.5237e+03	4.1160e+03
	10th	8.4290e+03	6.9937e+03	1.3373e+04	9.7805e+03
	13th(Worst)	5.5312e+04	1.2019e+05	2.4972e+05	6.3273e+04
	Mean	1.2232e+04	1.2746e+04	2.7956e+04	1.0417e+04
	Stdev	1.5556e+04	3.1280e+04	6.5104e+04	1.5889e+04
1e5	1st(Best)	1.1961e+03	1.2075e+03	1.2232e+03	1.2063e+03
	4th	1.2205e+03	1.2228e+03	1.2849e+03	1.2463e+03
	7th(Median)	1.2426e+03	1.3064e+03	1.3128e+03	1.3603e+03
	10th	1.2512e+03	1.3871e+03	1.3395e+03	1.5442e+03
	13th(Worst)	1.3069e+03	6.3453e+03	1.3802e+03	2.7365e+03
	Mean	1.2450e+03	1.7633e+03	1.3073e+03	1.4992e+03
	Stdev	3.2677e+01	1.3598e+03	4.0671e+01	3.9324e+02

Table 7.4 Average Rankings

Parameter Setting	Mean	5×10^3	10^4	10^5
Predicted - Set Values	1.67	1.50	2.00	1.50
Drift - Set Values	1.67	1.50	2.00	1.50
Parameterless - Set Values	3.50	3.50	2.00	5.00
Parameterless - Self Adaptive	4.83	3.50	6.00	5.00
Predicted - Self Adaptive	5.17	6.50	6.00	3.00
Typical - Self Adaptive	5.83	6.50	6.00	5.00
Typical - Set Values	6.67	6.50	6.00	7.50
Drift - Self Adaptive	6.67	6.50	6.00	7.50

Table 7.5 Comparison of the solutions found by different optimisation methods

Booster Period	Optimisation Algorithm	Dosing at Each Booster Location (mg/min) A	B	C	D	E	F
1	LP	589.0	7.9	419.0	0.1	0.1	0.0
	LLS	1442.0	10.2	0.0	3.8	2.9	1211.0
	GA1	599.3	9.8	473.6	0.7	0.4	0.0
	GA2	671.2	6.7	347.5	0.2	0.3	0.2
2	LP	0.0	0.0	0.0	0.2	0.6	727.0
	LLS	0.0	4.7	6.7	2.2	2.1	204.0
	GA1	0.0	0.7	0.0	0.3	0.4	713.5
	GA2	0.0	0.0	0.2	0.4	1.3	717.2
3	LP	636.0	4.9	454.0	0.0	0.1	8.0
	LLS	971.0	11.0	136.0	2.8	1.9	150.0
	GA1	680.9	4.3	413.0	0.0	0.3	47.1
	GA2	659.1	3.7	415.3	0.1	0.3	26.0
4	LP	0.0	0.4	0.0	0.8	1.4	409.0
	LLS	0.0	4.0	0.0	1.6	1.4	223.0
	GA1	0.0	0.0	0.3	0.7	1.0	400.7
	GA2	0.0	0.0	0.0	0.7	1.5	394.3

The Parameterless method did produce the best results averaged over the 13 independent GA runs for the stopping criterion of $FE = 10^4$. In this case, the difference was not statistically better than that produced by the Typical and Drift methods, however this result can be largely attributed to the higher variances for these methods.

As found in Chapter 6, the rankings of the methods presented in Table 7.4 suggest that the self adaptive calibration methods produced worse performance. Generally, the solutions found a similar solution quality as the set parameter values after $FE = 10^5$, however, for the shorter convergence times, the average solutions found by most of the self adaptive methods were much higher when compared to the average fitness function values found by the same calibration method with set values for p_m, p_c, and c.

7.1.3.3 Comparison of Best Solution Found

From Table 7.2, it can be seen that the best solution to the Cherry Hill-Brushy Plains optimisation problem after $FE = 10^5$ using the GA was found using the Drift calibration method, with a fitness function value of 1169 g/day. This solution, together with previously published solutions for this problem are presented in Table 7.5, where the linear

Table 7.6 Total mass of chlorine for different methods

Optimisation Algorithm	Total Mass (g/day)
LP	1176
LLS	1587
GA1	1205
GA2	1169

programming (LP), linear-least squares (LLS) and previous Genetic Algorithm (GA1) solutions are given. The LP and LLS solutions are taken from Propato and Uber (2004), as the Boccelli et al. (1998) solution was found using a previous version of EPANET, and is infeasible if evaluated with the most recent version (2.00.10). The objective of Propato and Uber (2004) was slightly different, which was to minimise the deviation of the chlorine concentration at each node from the minimum of 0.2 mg/L, compared to minimising the total mass of chlorine used in this work, although it is expected that both approaches will produce the same goal of minimising the total mass of chlorine added to the system. It should also be noted that the GA1 results of Munavalli and Kumar (2003) were obtained from $FE = 1.5 \times 10^5$ compared to $FE = 10^5$ used in this work, and if more evaluations were available, it would be expected that better solutions would be found by the GA presented in this study.

Generally, three of the four optimisation methods that have been applied to the Cherry hill-Brushy Plains network found very similar solutions. The LLS method found a slightly different solution, possibly due to a slightly different formulation of the fitness function. The results indicate that the best approach to operate the system was to have a large dose at node A, with some booster dosing at node C when the pump station is on, and some very small additions of chlorine at some or all of the other source nodes in the network. When the pumping station is off, the solutions given in Table 7.5 indicate that it is best to have the majority of chlorine added to the system at node F, with some minor topping up of the chlorine concentrations at source nodes D and E, as well as potentially B and C. None of the solutions found by the different optimisation methods included chlorine dosing at node A when the pumping station was off.

It can be seen from Table 7.6 that the GA solution presented in this thesis found the lowest mass of chlorine to be added to the system, found by the Drift calibration method with set parameter values. The linear programming method of Propato and Uber (2004) located a very similar solution. The major advantage of the GA optimisation

approach compared to the linear programming method is that specific formulations for the chlorine decay kinetics (i.e.; first order decay) or network behaviour (such as fixed pump operations) are not required to apply the GA method, which are required to apply the linear optimisation method. It would be much more difficult, if not impossible, to apply the linear programming optimisation method to the optimisation of a more complex WDS optimisation problem, such as that considered for the Woronora WDS in the following section.

7.2 WORONORA WDS

The Cherry Hill-Brushy Plain network optimisation problem is a rather simple WDS fitness function that has been optimised by a number of methods previously, and therefore allowed for a comparison of the performance of the GA used in this work, as well as for the comparison of the GA calibration methods considered. The second WDS optimisation problem considered in this chapter is an actual 'real world' WDS optimisation problem. The system used in this study is the Woronora WDS located in Sydney, Australia.

The same methodology that was applied in Chapter 6 and applied to the Cherry Hill-Brushy Plain fitness function has also been applied to the fitness function for the Woronora WDS. The following section provides a description of the Woronora system, before the steps taken to calibrate and validate the model are described. The Woronora fitness function is then outlined, before the results of the study are presented, including the fitness function characterisation, a comparison of the GA calibration methods on the Woronora fitness function, and an outline of the best solution found for a number of different scenarios.

7.2.1 System Description

The Woronora WDS supplies over 100,000 customers in south-western Sydney, Australia. The Woronora Dam has a catchment area of 75 km^2 and an operating capacity of 71,800 ML. The Woronora Water Filtration Plant (WFP) was first commissioned in July 1996, and has a capacity is 160 ML/D. Treatment methods at the WFP include chlorination, UV, both conventional and micro-filtration, and chloramination. Chloramination is used as the disinfection process for the total system. To minimise the likelihood of nitrification in the system, the chlorine set point at the WFP is held constant at 1.75 mg/L, and the Cl_2:NH_3 ratio is targeted to be maintained as close as possible to a 4:1 ratio throughout the system. Calcium hypochlorite tablet dosing is also carried out at all service reservoirs

Figure 7.2 Schematic of the Woronora WDS EPANET model

in a further attempt to prevent nitrification occurring in the WDS.

The layout of the Woronora WDS EPANET model can be seen in Figure 7.2. The Woronora dam supplies the Woronora WDS via the Woronora WFP and Clear Water Storage (CWS). The model represents the distribution system that was supplied by the Woronora CWS at the time of the work. However, it is possible to supply other zones in the area that are also connected to the network, but are not shown in the model. A Pressure Reducing Valve (PRV) is used to provide a head from the clear water tanks that is equivalent to the surface level of the dam, which has an elevation of 169 m.

The Woronora trunk main is approximately 1 m in diameter, and supplies water from the Woronora WFP into the distribution network. 10 km along the trunk main the Engadine pumping station supplies water to the Engadine reservoir. The Engadine reservoir then gravity feeds the Loftus and Maianbar reservoirs, as well supplying the Heathcote reservoir via a pumping station. The off-take to the Menai subsystem is 3 km further along the Woronora trunk main from the Engadine pumping station. From the Menai subsystem off-take, the Lucas Heights reservoir is gravity fed from the Woronora WFP. In this area of the network the size of the trunk main is 375 mm in the Lucas Heights subsystem, and 600 mm in the Menai subsystem. The Illawong reservoir is also shown

Table 7.7 Summary of reservoirs in Woronora EPANET model.

Name	Bottom Elevation (m)	Height (m)	Diameter (m)
Menai	124.8	12.2	51.08
Engadine	194.3	10.7	46.66
Loftus	156.4	9.6	35.87
Lucas Heights	145.5	9.5	30.41
Heathcote	219.5	12.5	37.76
Helensburgh	301.5	7.5	41.20
Maianbar	102.4	5.6	22.87

in Figure 7.2, which can be gravity fed from the Menai reservoir. However, at the time of this study, the Illawong reservoir was off-line to reduce detention times in this area of the network, and therefore has not been considered in the remainder of this work. The Helensburgh pumping station supplies water up to the Helensburgh reservoir from the Woronora CWS. The bottom elevations, heights and diameters of the reservoirs used in the EPANET model of the Woronora WDS are given in Table 7.7. As the elevation of the Woronora dam is 169 m, it is capable of supplying the Lucas Heights and Menai systems of the network by gravity, however booster pumps are required to supply the Helensburgh and Engadine systems.

The location of assets that can be used to control the operation of the Woronora WDS are given in Figure 7.2. There are pumping stations located on each trunk main that lead to the Helensburgh, Engadine and Heathcote reserviors. Automatic Inlet Control Valves (AICVs) are located on the inlet to the Lucas Heights, Loftus and Maianbar reservoirs. The Lucas Heights AICV is operated based on the water level in the Lucas Heights reservoir, and the water level to open and close the valve can be controlled remotely. The operations of the AICV on the inlet of the Loftus and Maianbar reservoirs are determined from set rules, as there is no remote control for the operation of these two valves. There is a Throttle Control Valve (TCV) on the inlet to the Menai reservoir, which can also be remotely controlled based on the water level in the Menai reservoir.

7.2.2 Model Calibration

A WATSYS model of the Woronora WDS was provided by the utility responsible for the system, Sydney Water Corporation. The WATSYS model was converted to an EPANET model, to allow the GA to simulate the system via the EPANET toolkit. Sydney Water

Corporation have a sophisticated Supervisory Control And Data Acquisition (SCADA) system in place for the Woronora WDS, and online monitors provide data about the system, including flows, reservoir levels and total chlorine concentrations, at regular intervals, in most cases every 15 minutes. Therefore, there is a large amount of data available for calibration of the model of the system.

The Woronora WFP is also capable of supplying the Sutherland Zone, which is fed off the Woronora trunk main between the Engadine pumping station and the Menai zone, and is located north-east of the Loftus reservoir in Figure 7.2. However, at the time of this study, the Sutherland zone was fed by the the Allawah reservoir, as the volume available from the Woronora dam was very limited. The Sutherland zone accounts for approximately 65% of the demand in the system under normal operating conditions, and therefore the demand on the system has decreased significantly, from approximately 80 ML/D to 25 ML/D. Due to the capacity of the WFP, the minimum operating flow from the WFP is 35 ML/D. Therefore, it is not possible to continually operate the WFP, as the minimum supply is greater than the demand in the system. In the model, the daily demand has been forcast assuming that it will be the same as the previous daily demand, recorded by the SCADA system. The plant has been modelled to shut down at midnight, and then starts up when it can supply the minimum flow for the rest of the day. For example, if the daily demand is forcast to be 28 ML/D, the plant will shut down at midnight, and then restart at 4:48 am to supply a constant 35 ML/D for the remainder of the day.

It is desirable to operate the WFP to produce a constant flow of treated water. This allows for ease of operation of the filtration plant, and also minimises abrupt changes in the water quality provided from the WFP. The current operation of the WDS to achieve a constant flow from the WFP involves sequentially filling each of the reservoirs in the system. For example, once the Engadine reservoir has been filled, the pumps are switched off, and then the TCV in the Menai zone is opened to fill the Menai reservoir, to maintain a similar demand from the WFP. In order to model this requirement, the WFP is represented by a node in the model with a negative demand, equal to the daily demand in the system if it is greater than the minimum operating flow of the WFP, or the minimum operational flow of 35 ML/D otherwise. Therefore, the operations of the network must maintain a constant flow from the WFP, otherwise hydraulic errors are produced by empty or overfull reservoirs in the model.

One of the most important properties of a WDS model is an accurate representation of the demands in the system, as the demand will control the flows and pressures in the system, and therefore also the water quality. Detailed information about the smaller

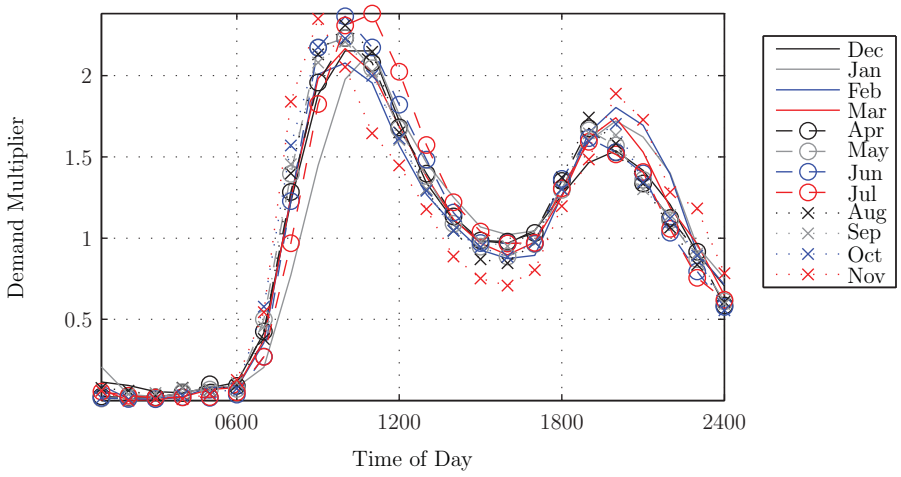


Figure 7.3 Monthly demand patterns for the demand node at Maianbar.

reticulation of the Woronora system was not available, hence each customer zone in the system has been represented by a single demand node on the outlet of each reservoir in the model. The flow and reservoir level data records available from the SCADA were used to determine the distribution of the demands in the system. However, flow data were not available to determine the true demand at all demand nodes in the system. A water balance was undertaken at each time step in the data in order to determine an approximation for the total demand at the different demand nodes in the network, as well as the daily pattern of the demand at each node. As the flow upstream and downstream of a desired node is known, as well as the change in reservoir level, which can be used to compute the change in storage, a water balance was carried out over the desired node to determine the demand for a given time. To undertake this analysis, it was assumed that the travel times were zero, therefore the difference between the flows recorded at different locations in the network at the same time represented the water that was consumed.

The water balance analysis was conducted for each month to determine if there were any seasonal changes in the demands in the system. The demand patterns for the Maianbar demand node for each month of the year can be seen in Figure 7.3. The patterns are multiplied by the base demand at each node, and therefore they are normalised to produce an average demand of 1. From Figure 7.3, it can be seen that the pattern of the demand consumption at the Maianbar demand node did not vary significantly from month to month, and this was a typical result for all the demand nodes that were investigated. Therefore, the average demand pattern at each node over all months have been used in the model.

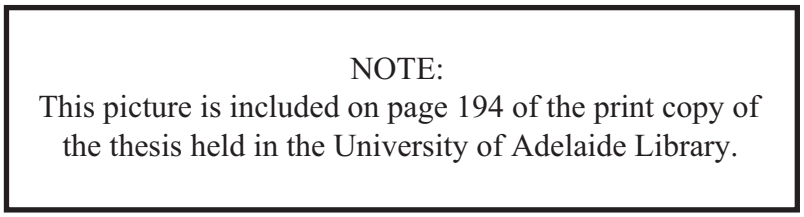


Figure 7.4 Spatial distribution of the daily demand around the Woronora WDS.

The demand patterns are used to represent how the demand at each location varies with the time over a day, however, the spatial distribution of the demand over the Woronora WDS is just as important to accurately model the demands in the system. Based on the same water balance approach used to determine the demand patterns, the fraction of the total daily demand consumed at the different demand nodes was also determined. In this way, if the total demand for a day is predicted to be 28 ML/D, then this total demand can be multiplied by the demand fraction for a particular node to determine the average daily demand, and the average daily demand is multiplied by the demand pattern for that node to determine the hourly demand for that node at each time in the simulation. The location and spatial distribution of demands in the system determined from the water balance study are shown in Figure 7.4. The distribution of demands throughout the system were found not to vary significantly throughout the year.

As the objective of the Woronora optimisation is to minimise the costs involved in operating the network, it is important to have an accurate representation of the cost of the electricity used by the pump stations. To compute total energy used over the day, accurate pump curves are required for each pumping station. The Helensburgh and Heathcote pumping stations are modelled as having two fixed speed pumps. However, the two active

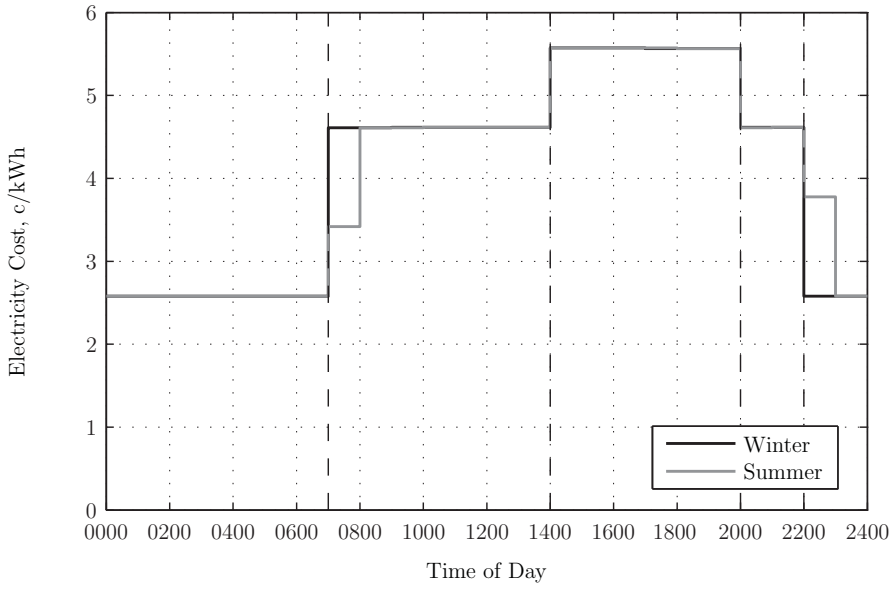


Figure 7.5 Electricity costs over a 24 hour period.

pumps at the Engadine pumping station are variable speed pumps, which are not straight forward to simulate with the EPANET hydraulic solver. In practice, the most efficient pump speed to pump to the mid-point of the upper and lower controls is determined, and this pump speed is maintained until the pumps are switched off, which occurs when the water level in the Engadine Reservoir reaches its upper limit. In order to simulate this operation in EPANET, if the pumps are switched on in one time step, the pump speed to provide the necessary pressure head for the flow through the pump is determined to be used in the following time step, and this pump speed is held constant until the pumps are switched off.

Once the energy used in the operation of the pumps in the system has been computed, the corresponding cost of the electricity used can be determined. The cost of electricity is variable over a day, where the cost of electricity in cents per kilowatt-hour for different time periods is given in Figure 7.5. The difference between the costs at 7am and 11pm are produced by one electricity biller adjusting their times to daylight saving time in summer, while the other electricity biller does not. Therefore, the electricity cost of running the pumps in the system can be computed by multiplying the number of kilowatts used each hour by the electricity cost for that hour.

Sydney Water Corporation is also billed for the maximum electricity demand. Therefore, if the pump stations are pumping more water and more current is being drawn, cost for electricity demand will increase. The peak electricity demand used to compute the

electricity bills persists for two years, thus if the peak electricity demand is increased on a given day, then that demand is used for the following two years to compute the electricity demand cost paid by the utility. It is highly undesirable to increase the peak electricity demand, and hence the peak electricity demand has been included in the fitness function as a constraint, where a penalty is applied if the peak electricity demand is increased, as the increase in demand will produce an increase in the electricity costs for the next two years. To compute the peak electricity demand, the power used by the pumps, in kW, is divided by a power factor to produce the peak electricity demand, in kVA. Based on the maximum power used by the pumps in a simulation of the current operations of the Woronora WDS, as well as billing records of the electricity demand that has historically been paid, a value of 0.85 has been used for the power factor.

The objective of the optimisation (i.e.; minimising cost) is constrained by the water quality produced by the operation of the WDS, and therefore a water quality model has also been developed. Total chlorine was used as a surrogate for the chloramine in the system, as only total chlorine measurements were available to calibrate the model, and it is assumed at the utility that if the total chlorine concentration is above 1 mg/L, then all the ammonia in the system will be bound with chlorine, and nitrification will not occur. A first order decay model has been used to represent the change in chlorine concentration. There are regular total chlorine measurements at the WFP and at the valve before the Engadine pumping station, along with flow measurements along the trunk main in this section of the system. Therefore, as the initial and final chlorine concentrations were known for this section along the pipe, as well as the travel time between the two points, given by the flow and pipe diameter, the decay rate of chlorine in this section of the pipe can be determined. Data from different times of the year were considered to investigate if the chlorine decay rate changed with water temperature. However, the analysis did not provide a reliable result, with the chlorine decay rate varying between $k_{\mathrm{Cl}} = -0.05$ to $-0.4\ d^{-1}$, with no relationship with water temperature or time of year. While this was a relatively closed section of the system, the true chlorine decay coefficient could not be determined from the available data. This is due to the influence of other effects in the system, such as the opening or shutting of valves, which produce sudden changes in the chlorine concentrations, and these could not be taken into account in the chlorine calibration model.

A separate study was undertaken at the Australia Water Quality Centre that performed a number jar tests of chlorine decay on a number of water sources from around Australia at different times of the year. One of the water sources tested in the study was from the

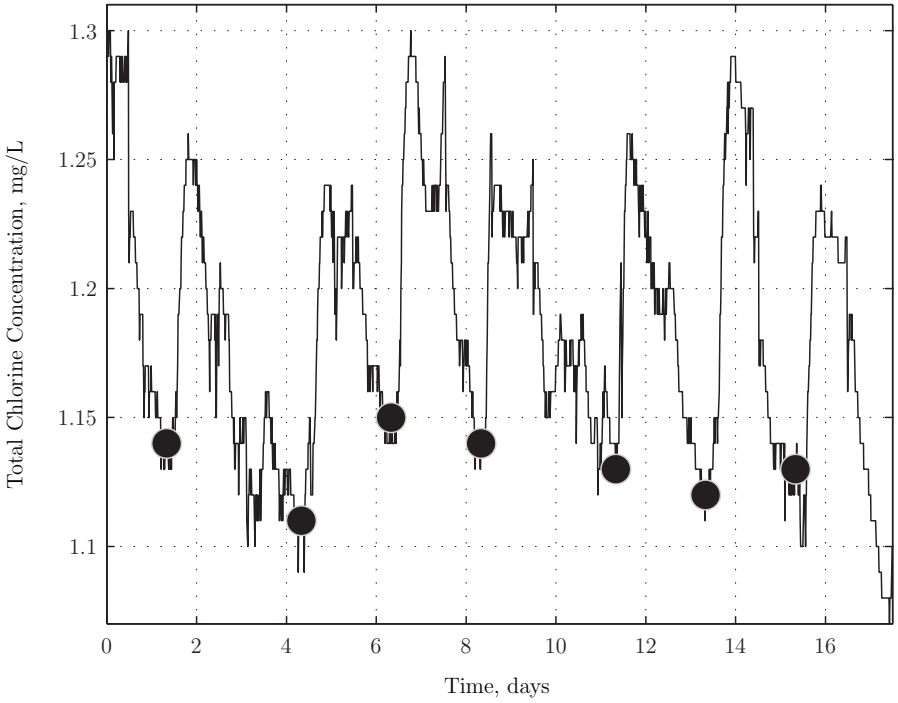


Figure 7.6 Total chlorine trend at the outlet of Menai reservoir for December 2005.

Woronora WFP. Based on the jar tests undertaken in that study, the total chlorine decay rate used for the water quality model of the Woronora WDS was $k_{\text{Cl}} = -0.24\ d^{-1}$, which falls between range of computed decay rates of $k_{\text{Cl}} = -0.05$ to $-0.4\ d^{-1}$.

The final part of the model of the Woronora WDS to be developed was to represent the calcium hypochlorite tablet dosing that was undertaken to increase the chlorine concentrations in the network. To determine the effect of the tablet dosing, the total chlorine concentration at the inlet and outlet of the Menai reservoir was used, along with records of when tablets were dosed into the reservoir. A plot of the total chlorine concentration at the outlet of the Menai reservoir for December 2005 can be seen in Figure 7.6, where the large circles are used to represent when tablets were dosed in the reservoir. It can be seen from Figure 7.6 that after tablets are dosed, the chlorine concentration at the outlet of the reservoir increases, and then slowly decreases over the next two days.

The influence of dosing the calcium hypochlorite tablets can be determined by considering the difference between the chlorine concentrations measured at the inlet and outlet of the Menai reservoir, if the decay of chlorine in the reservoir is assumed to relatively slow. The result from this analysis can be seen in Figure 7.7, where the difference between the outlet and the inlet concentrations is plotted for the time after each of the seven sets of tablets were dosed in December 2005. The thick dashed line is the average influence

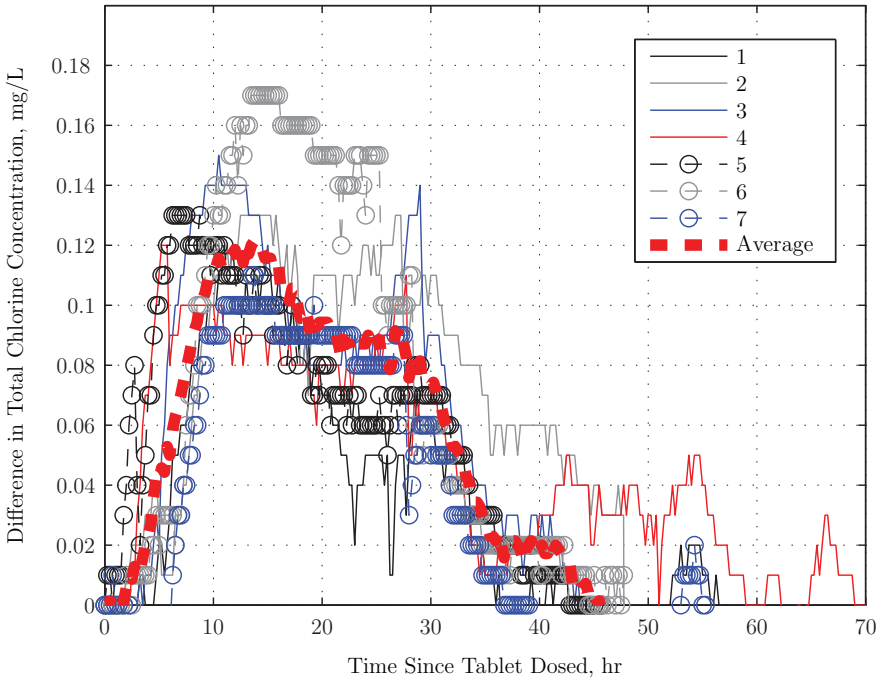


Figure 7.7 Influence of tablet dosing for December 2005.

of the tablets dosed for this period.

A similar analysis was undertaken on data collected over the month of July 2006. The increase in chlorine due to the five sets of tablets dosed in July can be seen in Figure 7.8, and again the thick dashed line provides the average influence of the tablet dosing for this month. These two cases of December 2005 and July 2006 provide an indication of the influence of the tablet dosing under both summer and winter conditions. The chlorine data at the Menai reservoir were erroneous between January and April 2006, therefore December 2005 was the latest time in the summer that the analysis could be undertaken.

The average influence of the calcium hypochlorite tablet dosing at the Menai reservoir for December 2005 and July 2006 are given in Figure 7.9. The peak for the two different plots can be seen to occur at a very similar time, approximately 14 hours after the tablet was dosed. The size of the peak is also very similar, at approximately 0.13 mg/L, and the difference between the two peaks is not significant, given the accuracy of the chlorine measurements. The average influence of the tablet dosing is also given in Figure 7.9. Based on this average influence of the tablet dosing, the model used to represent the effect of tablet dosing for the Woronora WDS is given by the triangular distribution in Figure 7.9. This provides an approximation to the observed influence of the tablet dosing, and was deemed appropriate for this study.

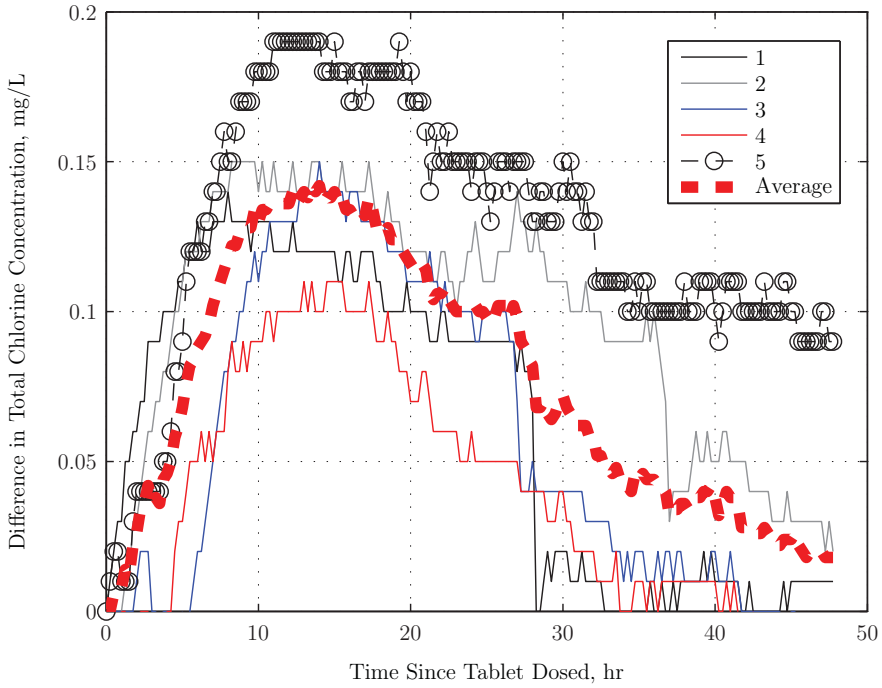


Figure 7.8 Influence of tablet dosing for July 2006.

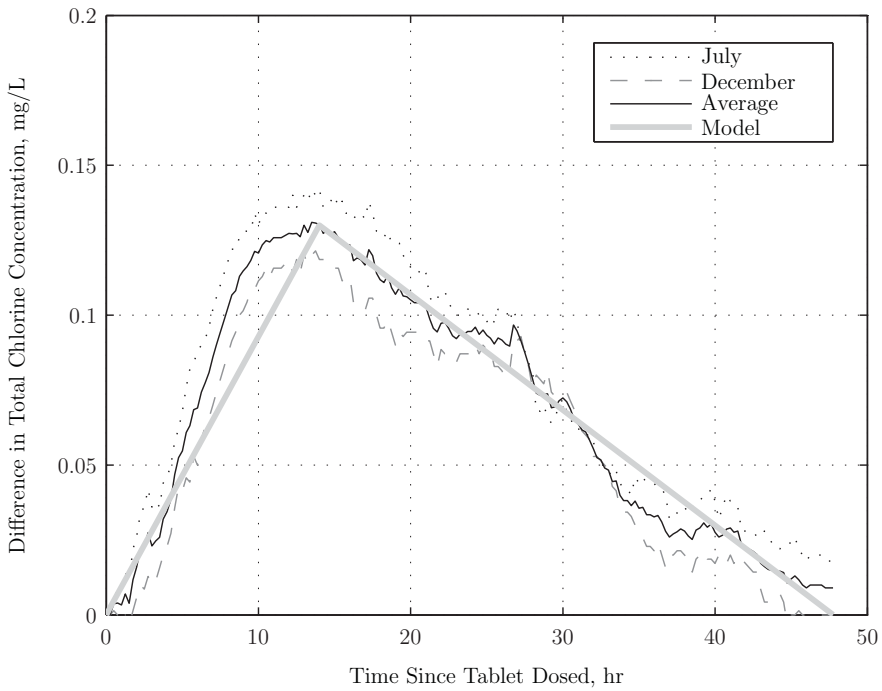


Figure 7.9 Calcium hypochlorite tablet dosing model.

7.2.3 Model Validation

The previous section outlined the calibration of the Woronora WDS EPANET model, in order to simulate the behaviour of the network as well as possible. In this section, the results from the simulation model are compared with measured data from the WDS, to determine the accuracy of the model. The operation of the WDS for a given day, including pump operations, valve operations and tablet dosing, have been simulated in the WDS model, so that the reservoir levels and total chlorine concentrations computed by the model can be compared against what was actually measured in the WDS.

The day used for the comparison was 4/8/2006, as after this date maintenance was undertaken on the Helensburgh pumping station, and the actual operations of the system were unknown. The reservoir levels and total chlorine concentrations in the model were initialised to the values recorded at midnight on this day. For nodes in the model where the chlorine concentrations are not measured, the initial values have been determined by a linear interpolation between data at known locations.

The operations of the AICV on the inlet of the Loftus and Maianbar reservoirs are determined from set rules, as there is no remote control for the operation of these two valves. Based on observations from the SCADA data available, the Loftus reservoir is maintained between 70 and 95% full. Therefore, if the reservoir level approaches 70% full, the valve will open, and then will shut once the water level in the reservoir reaches 95% full. Similarly, the AICV on the inlet of the Maianbar reservoir is operated to maintain the water level in the reservoir between 85 and 87%. These operations have been simulated in the model of the Woronora WDS, and therefore the operation of these AICVs are not decisions to be made.

The simulated and recorded water level in the reservoirs around the network that are controlled by pumping stations or valve operations have been compared. The results can be seen in Figures 7.10 to 7.14. From Figure 7.10, the simulated and observed reservoir profile for the Helensburgh reservoir can be seen, and the results indicate that the simulated result agrees very well with the true reservoir profile. Therefore, it can be concluded that the pump curve used for the Helensburgh pump station is accurate, and reasonable demands were used at the outlet of the reservoir.

The simulated and observed water profiles for the Engadine reservoir can be seen in Figure 7.11. The results suggest that the approximation to the variable speed pumping at the Engadine pumping station is not a perfect representation, as at times the pumps are pumping harder in the simulated model compared to those in the real system, indicated by the levels recorded by the SCADA system. This can be seen in Figure 7.11 by the

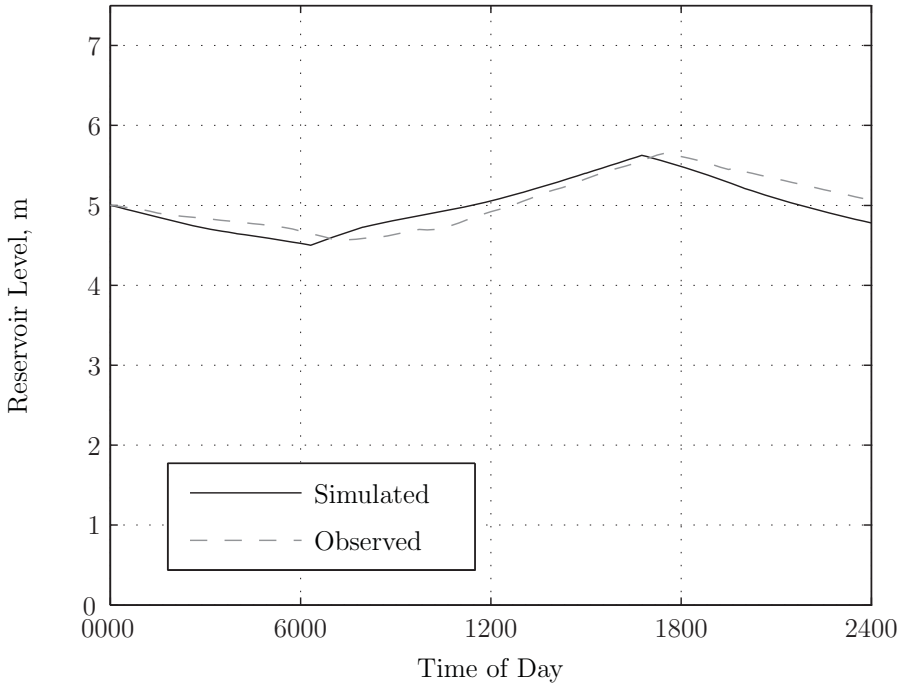


Figure 7.10 Simulated and observed Helensburgh reservoir profile.

faster increase in the water level for the EPANET simulation at around 0600 hours. However, for the remainder of the day, the reservoir profiles are in agreement, and when the pumps switch back on at 1 pm, the pumps can be seen to behave similarly. Therefore, the approximation to the variable speed pumping at Engadine is deemed to be reasonable for the Woronora WDS model.

The water level in the Heathcote, Lucas Heights and Menai reservoirs can be seen in Figure 7.12, Figure 7.13, and Figure 7.14, respectively. For all three of these plots, the simulation of the reservoir profile can be seen to agree with the true profile for this day for the first part of the simulation. However, generally after the respective pump or valve shuts off for the first time, the water level in the observed profile indicates that the reservoir is draining quicker than was simulated in the EPANET model. This will result in the pump or valve filling the reservoir earlier in the day than in the simulation model. The most likely reason for this difference is the demands that have been used in the simulation model have been produced by the demand patterns and the distribution of the daily demand determined in the calibration of the model. On a day-to-day basis, the actual demands in the system may be quite different from the average demands that were used in developing the model. Therefore, for the case presented here, it is likely that the predicted demands in the model are slightly lower than the true demands, and therefore

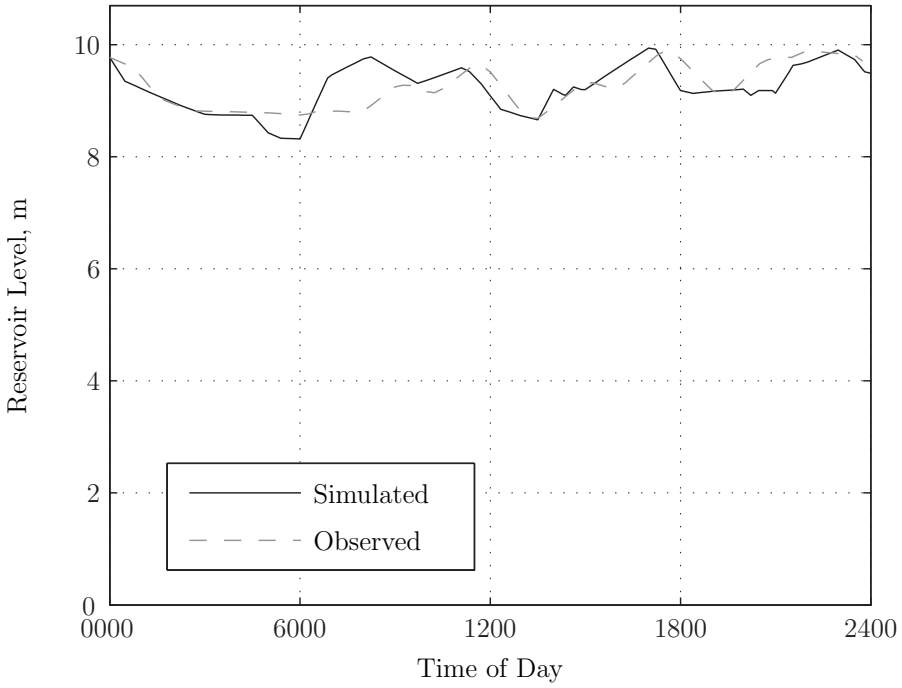


Figure 7.11 Simulated and observed Engadine reservoir profile.

the reservoirs take more time to drain to the trigger level where the pump or valve is opened again in the EPANET model.

Generally, the water profiles presented in Figures 7.12, 7.13, and 7.14 are in agreement with the true water profile recorded by the SCADA system. The most important characteristic to note is that the reservoir at Heathcote fills at the same rate for both the simulated and observed cases, indicating that the pumps at Heathcote are simulated accurately. Therefore, if the pumps are simulated accurately, it may be expected that the simulated electricity cost of pumping is also an accurate resemblance of the true electricity cost.

The pumping costs computed by the simulation of the Woronora WDS system for this day of operation was $176. The electricity bill for the month of August 2006 was $5086, which gives an average daily cost of $164. The simulated pumping cost does not include other costs that are included in the total electricity bill paid by the utility, such as capacity costs and other fees, however the majority of the electricity bill consists of the costs of the electricity used. Therefore, from the analysis of the reservoir profiles and electricity costs, it can be concluded that the simulated hydraulic behaviour of the Woronora WDS is a reasonable representation of the true hydraulic behaviour of the system.

The simulation of the total chlorine concentrations in the network have also been com-

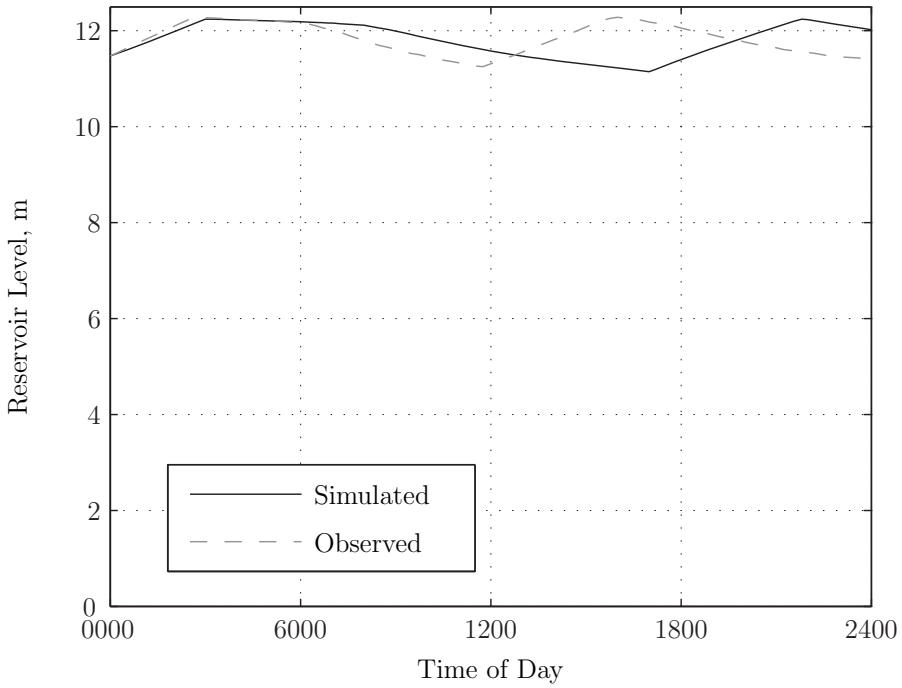


Figure 7.12 Simulated and observed Heathcote reservoir profile.

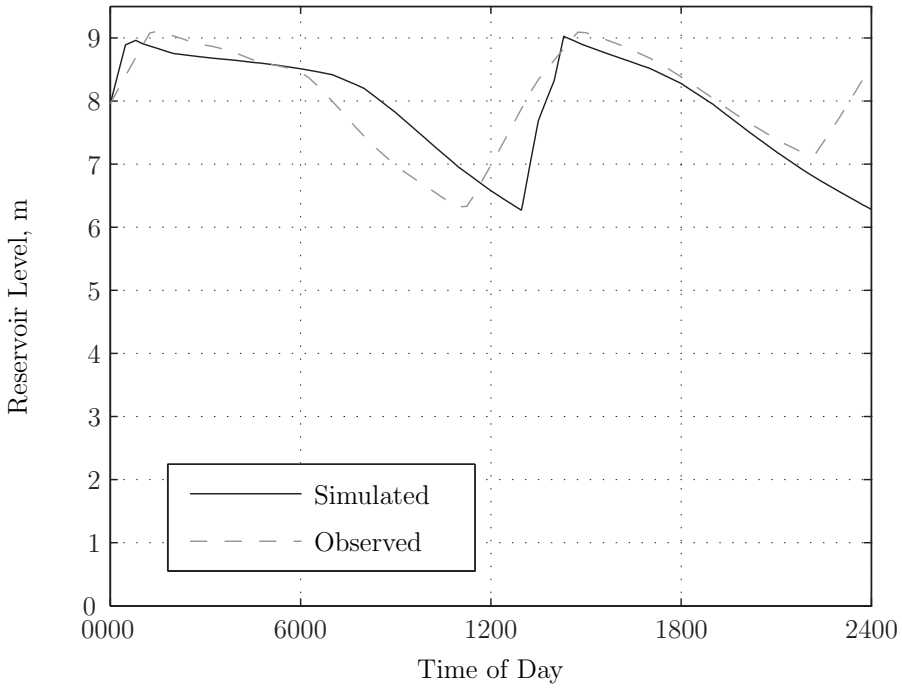


Figure 7.13 Simulated and observed Lucas Heights reservoir profile.

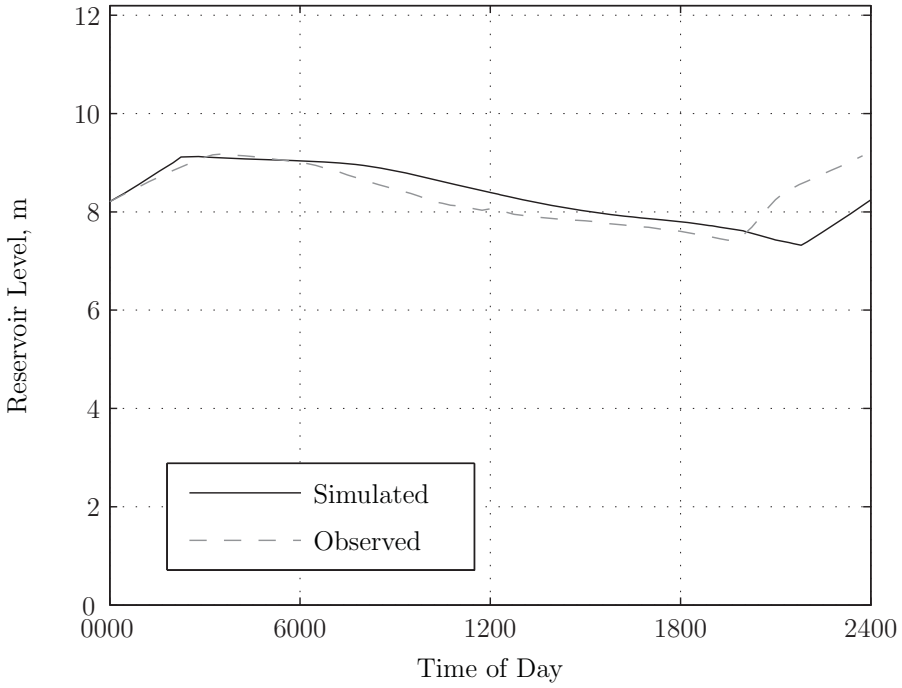


Figure 7.14 Simulated and observed Menai reservoir profile.

pared against those recorded by the SCADA system for the same day of operation. Two locations in the network where total chlorine concentrations are measured have been compared; the Engadine pumping station, and the inlet to the Menai reservoir. The simulated and observed total chlorine concentrations over a 24 hour period at the Engadine pumping station can be seen in Figure 7.15. It can be seen that the concentration of chlorine at this location does not change significantly over the day. This observation is typical of the chlorine concentrations at most locations in the WDS, and is part of the reason for the large range of chlorine decay constants observed from the data in Section 7.2.2. The water quality model can be seen to predict the slight changes in the chlorine concentration at the Engadine pumping station, however the small short term variations in the chlorine concentration are not simulated. These short term variations are a similar magnitude to the accuracy of the equipment used to measure the chlorine concentrations (approximately ± 0.1 mg/L). Therefore, these small changes in the concentration are most likely insignificant, and hence the simulation of the chlorine concentration at the Engadine pumping station is a reasonable representation of the true chlorine concentration at this location.

The other location where suitable data were available to compare the chlorine concentrations was at the inlet to the Menai reservoir. The simulated and observed chlorine concentrations over a 24 hour period at this location can be seen in Figure 7.16. Again,

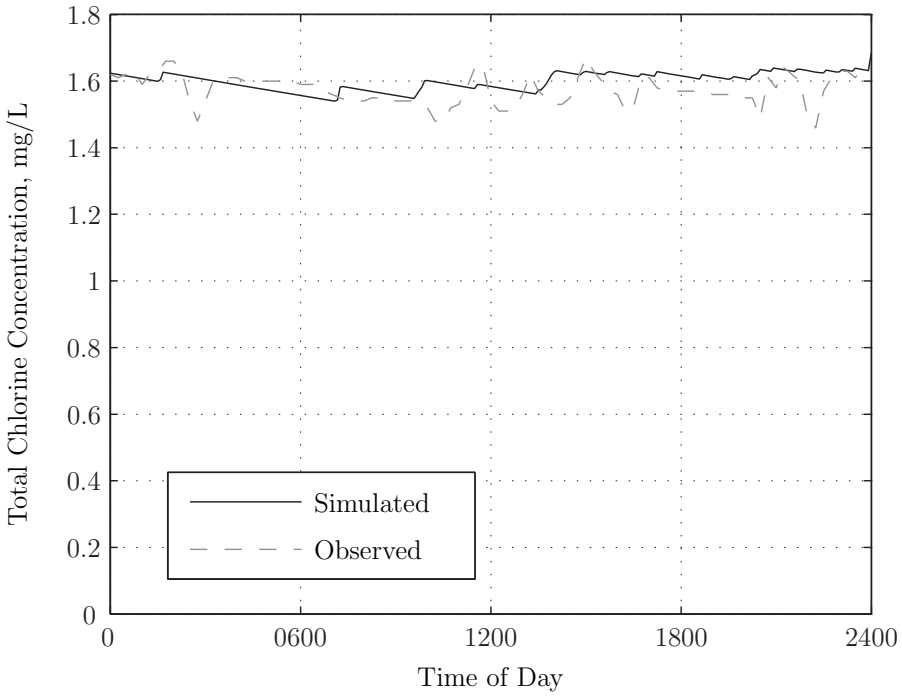


Figure 7.15 Simulated and observed total chlorine concentrations at the Engadine pumping station.

the simulation of the chlorine concentration can be seen to be a reasonable representation to the overall change in the true chlorine concentration, however, some of the small fluctuations in the chlorine concentration were not simulated. As outlined above, these fluctuations are of the same order as the error of the measurement equipment, and are most likely not significant. There is large step change in the chlorine concentration simulated with EPANET, which is not present in the true chlorine concentrations for this day. This result is most likely due to the simulation of the TCV at the inlet to the Menai reservoir, which has been opened relatively quickly in the model, and water of higher chlorine concentration from the WFP is released to the Menai reservoir. In reality, the TCV is opened more slowly, and therefore chlorine concentrations are observed to increase more slowly. The simulated result may also be due to effects such as dispersion being ignored in the simulation model, as in reality, longitudinal dispersion tends to reduce sharp fronts in chlorine concentrations in the system. While on first inspection, the step change in the simulated chlorine concentration in Figure 7.16 appears to be very different from the true chlorine concentration, the two profiles are rarely more than 0.1 mg/L apart, and therefore it has been concluded that, for the requirements of this study, the water quality model provides a suitable representation of the total chlorine concentrations in the Woronora

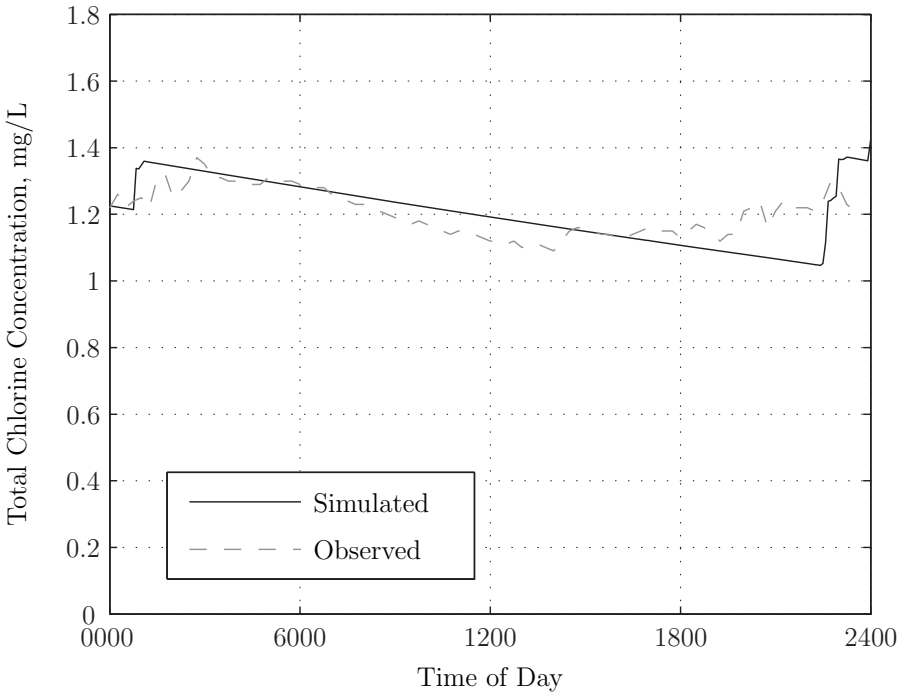


Figure 7.16 Simulated and observed total chlorine concentrations at the Menai reservoir inlet.

WDS.

The results presented in this section indicate that the simulated reservoir levels, chlorine concentrations and calculated pumping cost were all in reasonable agreement with that recorded from the actual Woronora WDS. Therefore, the Woronora WDS model calibrated in Section 7.2.2 has been determined to provide a reasonable representation of the true behaviour of the Woronora WDS, and thus can be used for the optimisation of the system. In the following section, the fitness function to be optimised for the Woronora WDS is outlined.

7.2.4 Woronora WDS Fitness Function

The objective of the Woronora optimisation problem is to minimise the total cost of operating the system. The total cost consists of the electricity cost due to running the pumps in the system, and the cost of dosing calcium hypochlorite tablets in the reservoirs. The cost of the tablets themselves is relatively cheap, however in order to dose the tablets, two employees are required to drive to each reservoir that requires the tablet dosing. Therefore, the cost of dosing a reservoir with calcium hypochlorite tablets has been used as $240, to include the costs of employing two staff to dose the tablets and the vehicle costs involved

in driving to each reservoir (Corinna Doolan, *pers. comm.*, October 30, 2006).

The decision variables that can be changed to meet the objective of the optimisation are given in Table 7.8. Each of the pumping stations consist of two active pumps, which are controlled by the water level in the reservoir downstream. Therefore, the decision variables for each pump are the water level in the reservoir to trigger the pump to turn on, and the water level in the reservoir to trigger the pump to turn off. In order for the pump operations to be accepted by the software that implements the trigger levels, the decisions must be structured in order from lowest to highest of: first pump on, second pump on, first pump off, and second pump off. The other hydraulic decisions to be made are the reservoir trigger levels for the AICV at Lucas Heights to open and close, and the trigger levels in the Menai reservoir to control the TCV upstream. Therefore there are a total of 16 different hydraulic decision variables.

As seen in Figure 7.5, the cost of electricity varies a great deal over a 24 hour period. Therefore, it might be expected that different operational regimes may be beneficial at different times of the day. For example, to minimise the cost of running the pumps, it may be beneficial to undertake as much of the required pumping as possible at night, when the electricity is much cheaper. Five different control periods over the day have been implemented to allow solutions such as this to be identified by the GA. The five periods have been derived from the electricity costs given in Figure 7.5. Consequently, the 16 hydraulic decision variables must be determined for the periods: 0000–0700, 0700–1400, 1400–2000, 2000–2200, and 2200–0000 hours. For the final period, the operations revert back to the operations that were used for the first time period of the day, as the electricity costs are the same, and in order to produce consistent pumping operations overnight. Therefore, there a total of four sets of the 16 hydraulic decision variables must be determined, producing 64 hydraulic decision variables.

The final decisions that must be made are whether or not to dose calcium hypochlorite tablets in each of the reservoirs. As the GA is real coded, for these decisions the tablet dosing has been represented by a value over the range $[0, 2)$, and to determine if the tables are dosed the value is truncated. Therefore, if the truncated value is 1, tablets are dosed, otherwise if the value is 0, no tablets are dosed in the reservoir. As there are seven reservoirs in the Woronora WDS, there are seven water quality decision variables, producing a total of 71 decision variables for the Woronora WDS fitness function when the water quality constraint is considered.

The objective of minimising the costs of operating the Woronora WDS is subject to a number of constraints on the system that must be satisfied. The constraints on the system

Table 7.8 Woronora Decision Variables

No.	Location	Asset	Decision
1	Helensburgh	Pump 1	Low TL
2	Helensburgh	Pump 1	High TL
3	Helensburgh	Pump 2	Low TL
4	Helensburgh	Pump 2	High TL
5	Engadine	Pump 1	Low TL
6	Engadine	Pump 1	High TL
7	Engadine	Pump 2	Low TL
8	Engadine	Pump 2	High TL
9	Heathcote	Pump 1	Low TL
10	Heathcote	Pump 1	High TL
11	Heathcote	Pump 2	Low TL
12	Heathcote	Pump 2	High TL
13	Lucas Heights	AICV	Low TL
14	Lucas Heights	AICV	High TL
15	Menai	TCV	Low TL
16	Menai	TCV	High TL
17	Helensburgh	Reservoir	Tablet Dose
18	Engadine	Reservoir	Tablet Dose
19	Loftus	Reservoir	Tablet Dose
20	Maianbar	Reservoir	Tablet Dose
21	Heathcote	Reservoir	Tablet Dose
22	Lucas Heights	Reservoir	Tablet Dose
23	Menai	Reservoir	Tablet Dose

are:

- Minimum water levels in each reservoir to maintain pressures in the reticulation system. As the reticulation system is not modelled further downstream than the reservoirs, the water pressure delivered at the customer level cannot be constrained directly. Therefore, the pressure constraint is implemented by requiring minimum water levels in the reservoirs that are known to maintain suitable pressures in the system.
- The water level in a reservoir must return to at least the level it was at the start of the day. This constraint is used to ensure that pumping costs are not reduced artificially, as without this constraint, the minimum amount of pumping, and therefore

the minimum electricity cost, is achieved by allowing all reservoirs to drain down to their lowest allowable level. Obviously, this is not a desirable situation for the operation of the system for the following day, where all the reservoirs will need to be refilled at the same time at the beginning of the day.

- Minimum flow from the CWS at the WFP. Due to the size of the plant equipment, the achievable turn-down ratio restricts operation to a minimum flow of 10 ML/D. Hence, flows from the CWS of less than 10 ML/D cannot be maintained in practice, and either the valve on the outlet of the CWS is open with a flow greater than 10 ML/D, or the valve must be shut.
- Minimum total chlorine concentrations of 1 mg/L to prevent nitrification occurring in the system.
- Maximum electricity capacity of 428 kVA to ensure the electricity capacity cost is not increased for the following two years.
- A maximum of 13 pump switches per day for each pump to reduce the risk of pump burn out produced by constantly switching the pumps on and off.
- The correct order of pumping trigger levels, so that the levels are ordered: first pump on, second pump on, first pump off, and second pump off. Each solution can be analysed to determine if it satisfies this constraint without running the hydraulic simulation model. Therefore, if necessary, each solution has been repaired before running the EPANET model to ensure the solution does not violate this constraint. As outlined in Section 2.1.1, a decision must then be made about whether or not to replace the infeasible solution with the repaired solution in the GA population. For this work, the rule suggested by Michalewicz (1996) has been adopted, which states that a 15% replacement of infeasible solutions is the best choice for numerical optimisation problems with nonlinear constraints.

In order for the selection operator to compare two infeasible solutions, penalty factors have been applied to any constraints that have been violated. A penalty of 100 multiplied by the maximum constraint violation over the simulation is included in the fitness function value for any infeasible solutions. The exception to this is the constraint on the minimum total chlorine concentration, which had the penalty of 5 $(\text{mg/L})^{-1}$ multiplied by the constraint violation at each time step that the chlorine concentration was below 1 mg/L at a demand node. These penalty values were found to provide a reasonable trade off between penalising infeasible solutions while still making use of slightly infeasible solutions in the optimisation process. However, it is clear that future research should be directed towards determining suitable methods for handling feasible and infeasible solutions.

Four different cases of the Woronora WDS system have been considered to be optimised. The first scenario initialised the model with reservoir levels to 70% full, and the chlorine concentrations initialised with the concentrations measured at midnight 4/8/2006. The second scenario to be optimised initialised the reservoir levels to 70% full, however, the constraint on the chlorine concentrations was ignored. This scenario was considered to investigate the effect of including the water quality constraint on the best hydraulic operation of the system. The third scenario was initialised with the actual reservoir levels and chlorine concentrations measured at midnight 4/8/2006. Many of the reservoirs in the system are much higher than the 70% full considered for scenario 1, and it is expected that by reducing the overnight reservoir levels in scenario 1, compared to scenario 3, that more pumping will be able to be shifted into the off-peak electricity period, therefore producing lower pumping costs. The final case of the Woronora WDS fitness function optimised considered only the hydralic constraints with the reservoirs in the model initialised to the actual levels measured at midnight, 4/8/2006, to provide a comparison with scenario 3 to investigate the effect of removing the water quality constraint on the best hydraulic operation of the system.

In order to suitably simulate each scenario, different simulation times were used for the scenarios with and without water quality constraints. For the two scenarios without the water quality constraint, the simulation time was a standard 24 hour period. For scenarios 1 and 3, where the water quality produced by the operation of the WDS was taken into consideration, the simulation was run until the total effect of the tablet dosing was observed. If 24 hour simulations were also used for the water quality solutions, it would be unlikely that any tablets would be dosed, as the increased chlorine concentration produced by the tablets would not be fully observed until the day after the tablets had been dosed. Usually, the tablets are dosed at 9 am in the Woronora WDS, and the tablet dosing model, given in Figure 7.9, indicates that the effect of the tablet dosing persists for 48 hours. Therefore, a 57 hour simulation has been selected for scenarios 1 and 3, to allow enough simulation time for the influence of the tablet dosing to be observed, without unnecessarily increasing the computational requirements. For these scenarios, where the simulation time was longer than 24 hours, the demand, demand patterns, and operation of the system were repeated for each 24 hour period.

7.2.5 Woronora Results

The same comparison of the different GA calibration methods considered in Chapter 6 has been applied to the Woronora WDS fitness function. Each GA calibration method

has been tested with 13 different sequences of random numbers. The same stopping criteria have been used for the Woronora fitness function that were used for the Cherry Hill-Brushy Plain fitness function, $FE = 5 \times 10^3$, 10^4, and 10^5. For 1 GA run, these stopping criteria correspond to approximately 2.5 minutes, 5 minutes and 50 minutes of CPU time, respectively, for the version of the fitness function with only the hydraulic constraints. With the water quality constraints also considered in the fitness function, both the hydraulic and water quality models must be run, and the simulation time was increased from 24 to 57 hours. Therefore, for scenarios 1 and 3 these stopping criteria correspond to approximately 9 minutes, 18 minutes, and 3 hours of CPU time, respectively. The benchmark machine was a 2.2 GHz workstation with 1 GB RAM running the Linux operating system. As each of the eight calibration methods has been run 13 times on the four different variations of the fitness function, this equates to a total of 38 days of CPU time required to perform the comparison of GA calibration methods on the Woronora fitness function.

In the following section, the results from applying the fitness function statistics to the Woronora fitness function are presented, before the performance of the GA calibration methods are compared for the different versions of the fitness function for different stopping criteria. This is followed by a description of the best solutions found by the GA to operate the Woronora WDS for the different scenarios considered.

7.2.5.1 Fitness Function Characterisation

The results from applying the separability measure to the Woronora WDS fitness function are given in Table 7.9. The interactions are presented for case 2 of the fitness function, where the water quality constraints are not considered and the model was initialised with the lowered initial reservoir levels. Similarly to the approach used for the Cherry Hill-Brushy Plain fitness function, each pair of decision variables has been considered independent of the time of day, to dramatically decrease the number of pairs of decision variables to be processed. Each decision variable is represented by the corresponding number given in Table 7.8.

The results presented in Table 7.9 suggest that every decision variable in the Woronora fitness function interacts with every other decision variable. As the fitness function is highly constrained, it is unlikely that a feasible solution that satisfies every constraint will be found by the GA, let alone by randomly sampling the search space to compute the separability measure. Also, as the constraint violation is computed using a highly non-linear simulation model, the fitness function value might be expected to vary greatly for

Table 7.9 Separability measure results for scenario 2 of the Woronora fitness function

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0.685	0.482	0.661	0.640	0.665	0.609	0.646	0.693	0.553	0.582	0.550	0.655	0.574	0.673	0.583
2		0.700	0.715	0.671	0.577	0.657	0.722	0.553	0.701	0.715	0.640	0.996	0.641	0.901	0.616
3			0.661	0.682	0.622	0.548	0.638	0.729	0.709	0.522	0.674	0.694	0.657	0.712	0.657
4				0.724	0.662	0.668	0.627	0.667	0.697	0.662	0.711	0.679	0.991	0.721	0.688
5					0.676	0.754	0.741	0.696	0.718	0.726	0.663	0.713	0.673	0.588	0.682
6						0.811	0.796	0.627	0.675	0.509	0.650	0.789	0.667	0.673	0.699
7							0.767	0.721	0.774	0.802	0.663	0.878	0.701	0.659	0.618
8								0.694	0.718	0.732	0.726	0.912	0.955	0.992	0.714
9									0.713	0.650	0.634	0.944	0.597	0.741	0.702
10										0.748	0.708	0.594	0.660	0.689	0.675
11											0.673	0.738	0.715	0.630	0.622
12												0.996	0.676	0.643	0.933
13													0.694	0.610	0.811
14														0.579	0.653
15															0.487

Table 7.10 Statistic values for the Woronora fitness function

Scenario	l	R_{av}	R_{l}	R_{T}	D	k	g_{conv}	N		
								5×10^3	10^4	10^5
1	71	0.134	0.45	0.134	0.011	0.988	655	10	20	150
2	64	0.122	0.45	0.122	0.019	0.989	692	10	10	140
3	71	0.129	0.45	0.129	0.020	0.989	693	10	10	140
4	64	0.117	0.3	0.117	0.014	0.993	1159	10	10	90

changes in the value of one or two decision variables. If there is no systematic change in the fitness function values produced by different values for one or two of the decision variables, the mutual information used to compute the separability measure will detect a relationship, implying a highly epistatic interaction and the value for the separability measure will be close to 1. This can be seen for a number of pairs of decision variables in Table 7.9, for example, between decision variables 8 and 15, and 12 and 16. Due to the computation requirements involved in computing the separability measure, it has not been applied to the three other scenarios of the fitness function considered, as it is expected that a similar result, of all decision variables interacting with each other, would be obtained.

The spatial correlation measure and the dominance measure have also been applied to the Woronora WDS fitness function. The values determined for the four different scenarios considered are given in Table 7.10. It can be seen from Table 7.10 that the different variations of the fitness function considered have little affect on the fitness function statistics computed, with the only noticeable difference being a decrease in the correlation length for scenario 4. The results for the dominance statistic indicate that there was not one decision variable that dominated the fitness function value. The results for the spatial correlation statistics indicated that the fitness function is a plane, as $R_{\text{av}} = R_{\text{T}}$. Otherwise, if $R_{\text{av}} < R_{\text{T}}$ the results would suggest the fitness function has a 'big bowl' structure, as the fitness function value for solutions far apart in the search space are correlated on average.

The values of k and g_{conv} computed from the spatial correlation and dominance measure results are given in Table 7.10. The larger value predicted for g_{conv} for Scenario 4 is produced by the higher value of $k = 0.993$, which is due to the shorter correlation length for this version of the fitness function. The corresponding population sizes used by the Predicted method for each of the stopping criterion considered are also given in Table 7.10.

For the Drift GA calibration method, the population size was determined by Equation 6.1, based on l and the different cases of FE considered. Therefore, for the Woronora

Table 7.11 Ranking of GA calibration methods for Woronora fitness function for $FE = 5 \times 10^3$

Parameter Setting	Mean	Scenario 1	2	3	4
Predicted - Set Values	1.6	1.0	2.0	1.5	2.0
Parameterless - Set Values	2.8	2.5	5.0	1.5	2.0
Drift - Set Values	3.6	2.5	2.0	4.0	6.0
Typical - Self Adaptive	4.4	6.5	2.0	7.0	2.0
Typical - Set Values	4.8	4.0	5.0	4.0	6.0
Drift - Self Adaptive	5.4	6.5	5.0	4.0	6.0
Predicted - Self Adaptive	6.8	6.5	7.5	7.0	6.0
Parameterless - Self Adaptive	6.8	6.5	7.5	7.0	6.0

WDS fitness function, the population sizes used by the Drift calibration method were $N = 30$ ($FE = 5 \times 10^3$), $N = 40$ ($FE = 10^4$), $N = 110$ ($FE = 10^5$). The different problem sizes of $l = 64$ and 71 did not influence the population size used, as the population size to be used by the GA has been rounded to the nearest 10 solutions.

7.2.5.2 Comparison of GA Calibration Methods

The results from the optimisation of the Woronora WDS by the different GA calibration methods can be found in Appendix D, where the fitness function values for the 1st (Best), 4th, 7th (Median), 10th, and 13th (Worst) solutions are presented, along with the mean and standard deviation of the fitness function value for all 13 solutions found for each calibration method, and each convergence criterion. The ranking of the different GA calibration methods for the four scenarios of the Woronora WDS fitness function are given in Table 7.11, Table 7.12 and Table 7.13 for the stopping criteria of $FE = 5 \times 10^3$, $FE = 10^4$, and $FE = 10^5$, respectively.

In general, the calibration of the GA did not have as great an impact on the performance of the GA on the Woronora WDS fitness function as the other fitness functions considered in this thesis. This can be seen in the rankings of the different methods for each case in Table 7.11, Table 7.12, and Table 7.13, where generally there were only two or three different ranks shared between the eight GA calibration methods. This may be due to the lower average correlation computed for this fitness function of $R_{\text{av}} \approx 0.125$, where for the Cherry Hill-Brushy Plain fitness function, and many of the fitness functions considered in Chapter 6, the average correlation of the fitness function was $R_{\text{av}} \approx 0.25$.

Table 7.12 Ranking of GA calibration methods for Woronora fitness function for $FE = 10^4$

Parameter Setting	Mean	Scenario 1	2	3	4
Predicted - Set Values	2.5	2.0	3.0	3.0	2.0
Parameterless - Set Values	2.5	2.0	3.0	3.0	2.0
Drift - Set Values	3.5	2.0	3.0	3.0	6.0
Typical - Set Values	4.2	5.0	3.0	3.0	6.0
Typical - Self Adaptive	4.2	5.0	3.0	7.0	2.0
Drift - Self Adaptive	5.9	7.5	7.0	3.0	6.0
Parameterless - Self Adaptive	6.2	5.0	7.0	7.0	6.0
Predicted - Self Adaptive	6.9	7.5	7.0	7.0	6.0

Table 7.13 Ranking of GA calibration methods for Woronora fitness function for $FE = 10^5$

Parameter Setting	Mean	Scenario 1	2	3	4
Predicted - Set Values	3.8	2.5	3.5	4.0	5.0
Drift - Set Values	3.8	2.5	3.5	4.0	5.0
Parameterless - Set Values	3.8	6.5	3.5	4.0	1.0
Typical - Self Adaptive	4.6	2.5	7.0	4.0	5.0
Predicted - Self Adaptive	4.8	6.5	3.5	4.0	5.0
Typical - Set Values	4.8	6.5	3.5	4.0	5.0
Drift - Self Adaptive	4.8	6.5	3.5	4.0	5.0
Parameterless - Self Adaptive	5.9	2.5	8.0	8.0	5.0

Therefore, as there is less structure in the fitness function, the GA must rely more on randomly searching the search space, and the parameter values are less influential on the final solution found by the GA.

The rank of each calibration method for each stopping criterion, and the average overall ranking for each calibration method, is given in Table 7.14. The order of the different calibration methods is very similar to that obtained in Chapter 6 and Section 7.1, where again, the Predicted method with set values produced the best overall results. The Parameterless method with set parameter values also produced very good results, as the only statistical difference between the two methods was that the Parameterless method had a slightly lower ranking for the $FE = 5 \times 10^3$ stopping criterion.

The Drift GA calibration method with set values for p_m, p_c, and c also performed well

Table 7.14 Overall ranking of GA calibration methods for Woronora fitness function

Parameter Setting	Mean	5×10^3	10^4	10^5
Predicted - Set Values	2.62	1.62	2.50	3.75
Parameterless - Set Values	3.00	2.75	2.50	3.75
Drift - Set Values	3.62	3.62	3.50	3.75
Typical - Self Adaptive	4.42	4.38	4.25	4.62
Typical - Set Values	4.58	4.75	4.25	4.75
Drift - Self Adaptive	5.33	5.38	5.88	4.75
Predicted - Self Adaptive	6.12	6.75	6.88	4.75
Parameterless - Self Adaptive	6.29	6.75	6.25	5.88

on the Woronora WDS fitness function, with a ranking of 3.62. The Drift method also had the same overall ranking as the Predicted and Parameterless calibration methods for the $FE = 10^5$ stopping criterion. The Drift and Predicted calibration methods used very similar population sizes for $FE = 10^4$ and 10^5, and therefore the ranks for these methods for these two stopping criteria are also very similar, with the only difference being that the Predicted method located a better solution on average for scenario 4 after $FE = 10^4$. However, for $FE = 5 \times 10^3$ the Predicted method with set parameter values consistently outperformed the Drift method, where in this case the Predicted method used $N = 10$, compared to $N = 30$ for the Drift method.

The Typical GA parameter values produced statistically poorer results for all stopping criteria than the three other GA calibration methods with set parameter values. This result further reinforces the need to determine the most suitable GA parameter values on a case by case basis. For the Woronora case study, the Typical method was again the only calibration method that produced better results with self-adaptive parameter values than with set parameter values. Otherwise, for the Drift, Predicted and Parameterless methods, the self-adaptive parameter values produced significantly worse results than when static values were used for the whole GA run. As mentioned previously, this result is most likely due to the increase in the problem size, and time spent on searching for good parameter values, which is time taken away from finding better solutions to the fitness function.

7.2.5.3 Comparison of Best Solutions

The best solutions found by the GA for each of the four scenarios of the Woronora WDS fitness function was identified, and the costs involved in operating the pumps in the system for each case can be seen in Table 7.15. The results indicate that the inclusion of the

Table 7.15 Pumping cost of best solution found for different scenarios.

Scenario	1	2	3	4
Tank Levels	Low	Low	High	High
WQ Constraint	Yes	No	Yes	No
Total Pumping Cost ($)	295.80	120.65	333.82	140.68
Simulation Time (hr)	57	24	57	24
Pumping Cost/Day ($)	124.55	120.65	140.56	140.68
Improvement Over Cost of Current Operations (%)	29.23	31.45	20.14	20.07

constraint on total chlorine did not have a significant influence on the cheapest pumping cost found. The cheapest cost found by the GA with the original reservoir levels was actually slightly cheaper with the water quality constraint. However the difference is only $0.12/day, and the cheaper solution is most likely also suitable for scenario 4 without the water quality constraint, and was probably not found for that case due to the stochastic nature of the GA.

From Table 7.15, it can be seen that decreasing the initial reservoir levels produced a significant decrease in the cost of pumping required. With the original initial reservoir levels (scenarios 3 and 4), a saving of 20% was found when compared to the pumping cost determined for the current operation of the system. By reducing the initial reservoir levels to 70% of the height of the reservoir (scenarios 1 and 2), the solutions found produced a 30% saving in the pumping costs compared to the current operations. By reducing the reservoir levels, more pumping can be undertaken before 0700 hours, when the electricity cost is cheapest. It should be noted that the decrease in the initial starting level of the reservoirs did not produce a lower operating level for the reservoirs, as in Figures 7.17–7.21 it can be seen that the reservoirs are still filled to close to 100% full at some time during the day.

For scenarios 1 and 3, the best solutions found suggest that no calcium hypocholorite tablets should be dosed for the simulations of the Woronora WDS considered. The increase in the total chlorine concentration provided by the tablets peaks at 0.13 mg/L, 9 hours after the tablets are dosed, and the cost of dosing the tablets to provide this increase in chlorine concentration at a reservoir was $240. Hence, it is very expensive to dose the tablets, considering the small increase in chlorine concentration that is gained, and it is therefore not surprising that the best solutions found by the GA did not include tablet dosing in the system.

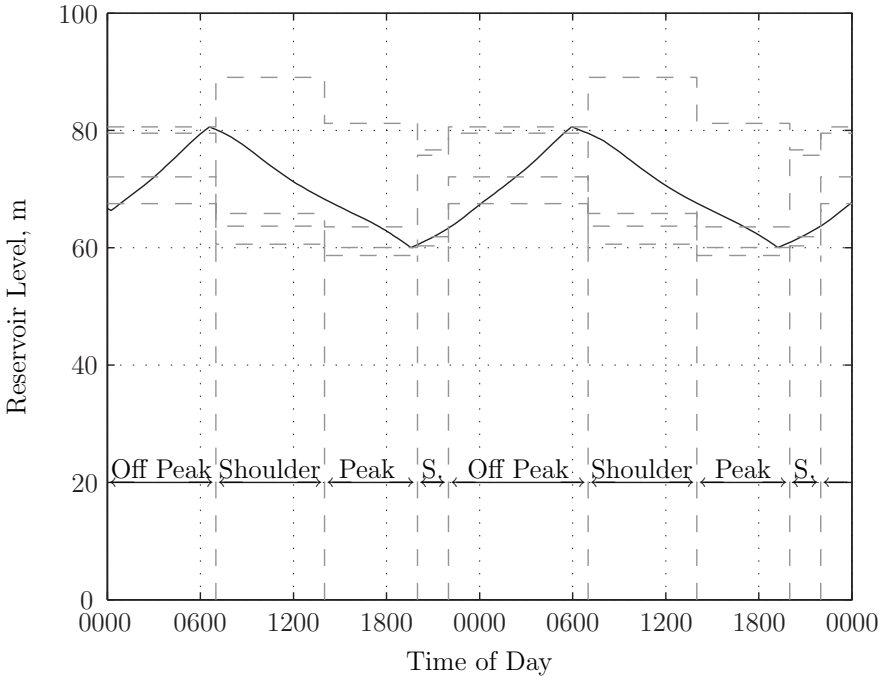


Figure 7.17 Helensburgh reservoir profile with original initial levels

Original Initial Reservoir Levels The reservoir levels over a 48 hour period for scenario 3 with original initial reservoir levels and the water quality constraint can be seen in Figures 7.17–7.21. The grey dashed lines on the figures represent the reservoir trigger levels determined by the GA to control the asset (pump or valve) that fills each reservoir. From Figure 7.17, it can be seen that the Helensburgh reservoir is filling between 2000 and 600 hours, therefore the pumping required is undertaken in the off peak period when electricity is cheapest, thus minimising the electricity cost involved.

From Figure 7.18, it can be seen that there is some pumping to the Engadine reservoir during the cheapest part of the day between 2200 and 2400 hours. However, Figure 7.18 suggests that some pumping must be undertaken during the day, as the reservoir is filling between 0900 and 1400 hours, when electricity is more expensive. The reservoir starts to drain after 1400 hours, and therefore for the best solution found suggests that no pumping should be undertaken in the period of the day when electricity is most expensive.

A similar profile can be seen for the Heathcote reservoir, however, there is only one period during the day when the reservoir is filled, as opposed to two for the Engadine reservoir. The reservoir profile produced by the best GA solution found for scenario 3 can be seen in Figure 7.19. The results suggest that it is not possible to fill the Heathcote reservoir during a period of the day when electricity is cheapest, as the reservoir is filling

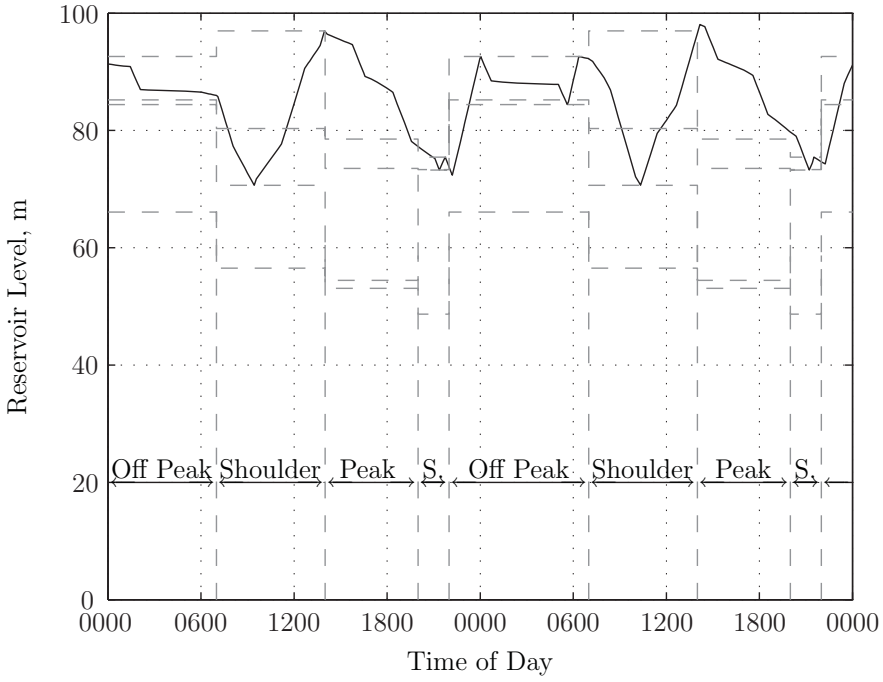


Figure 7.18 Engadine reservoir profile with original initial levels

between 0700 and 1400 hours, in the period of the day when electricity is second cheapest. As was the case for the pumping at the Engadine pumping station, the pumps are off during the most expensive time of the day, between 1400 and 2000 hours.

The Lucas Heights reservoir is filled by opening an AICV upstream, and therefore there is no electricity cost involved in filling this reservoir. The Lucas Heights reservoir water profile for the best solution found can be seen in Figure 7.20. The results suggest that it is best to fill the Lucas Heights reservoir after 1400 hours, when it is most expensive to operate the pumps that fill the Engadine, Heathcote and Helensburgh reservoirs. The reservoir is kept full until the electricity becomes cheaper at 2000 hours, when the Helensburgh pumping station switches on, and soon after, the Engadine pumping station switches on at 2200 hours. Filling the Lucas Heights reservoir at 1400 hours will assist in maintaining the required constant flow from the WFP, as generally one of the reservoirs in the WDS should be filling at all times. The results suggest that in the period of the day when electricity costs are highest, it is possible to direct the flow to the Lucas Heights reservoir, and keep the pump stations in the system off to reduce pumping costs.

As was the case for the Lucas Heights reservoir, the Menai reservoir is gravity fed from the WFP, and hence there is no cost involved in opening the TCV upstream to fill the reservoir. Therefore, the best operation of the Menai reservoir is based on balancing

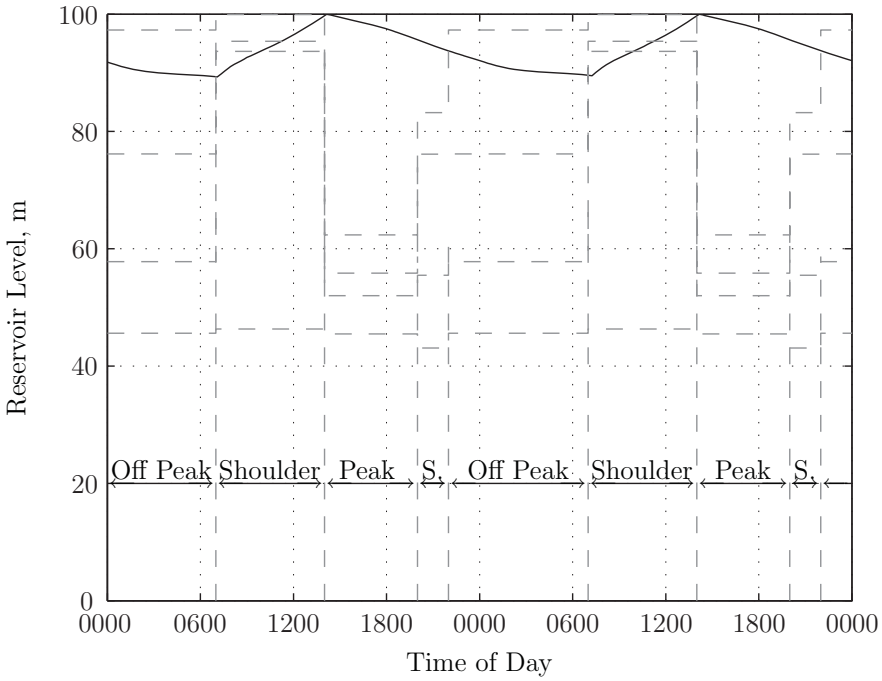


Figure 7.19 Heathcote reservoir profile with original initial levels

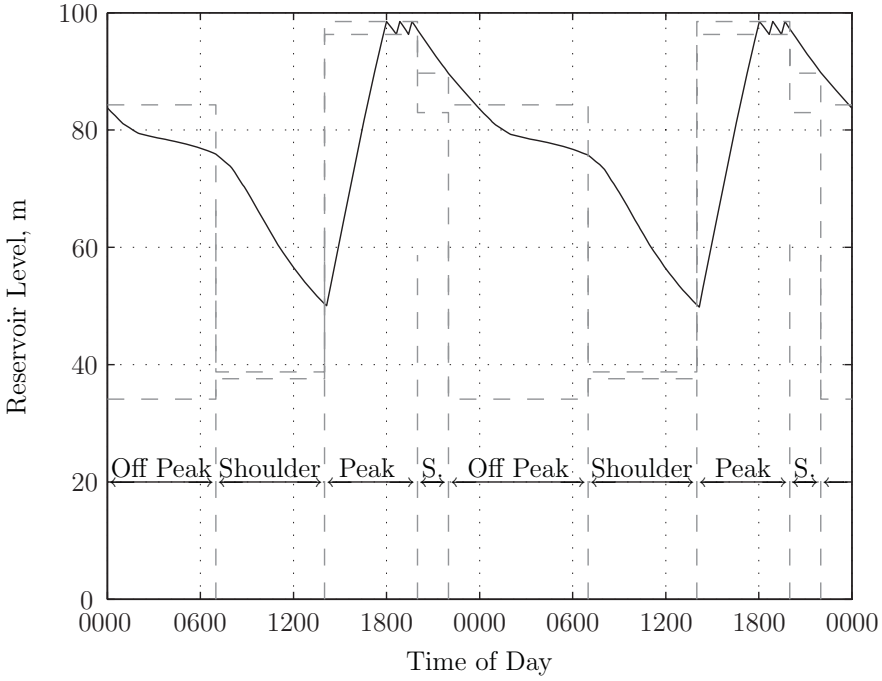


Figure 7.20 Lucas Heights reservoir profile with original initial levels

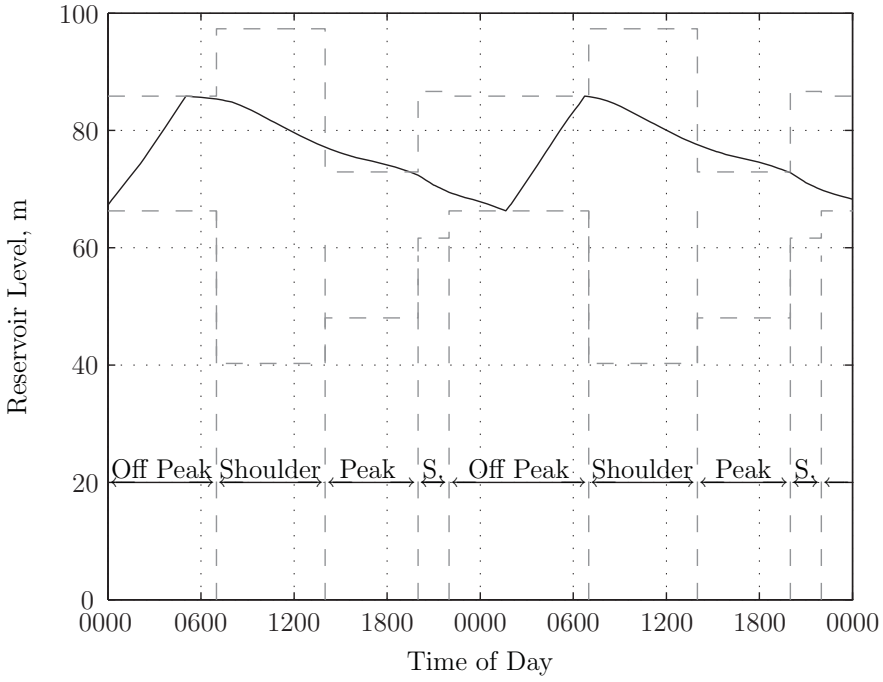


Figure 7.21 Menai reservoir profile with original initial levels

the flows in the system, as opposed to minimised the cost of filling the reservoir. From Figure 7.21, it can be seen that the results suggest that the best time to fill the Menai reservoir is between 0000 and 0600 hours. For this period of the day, the demands on the system are very low, as most people are asleep and not consuming water, and therefore most of the water produced must go into storage. The only other reservoir filling at this time is Helensburgh, and therefore to balance the flow from the WFP, the best solution found by the GA suggests that it is best to fill the Menai reservoir at this time to balance the flows in the system.

The total chlorine concentration at the demand nodes in the Woronora WDS for the best solution found for scenario 3 in terms of the fitness function can be seen in Figure 7.22. It can be seen that the chlorine concentration at most nodes was well above 1 mg/L, and therefore the minimum chlorine constraint does not affect the operation of the WDS. This result can be used to explain the very similar pumping costs determined for scenarios 3 and 4, with and without the water quality constraint, as the operations do not need to change to satisfy the chlorine constraint that is imposed in scenario 3.

While the chlorine concentration at most nodes was above the 1 mg/L minimum constraint, the concentration at the Maianbar demand node started below 1 mg/L, and stayed below 1 mg/L for the entire simulation. While it may have been possible to find a solution

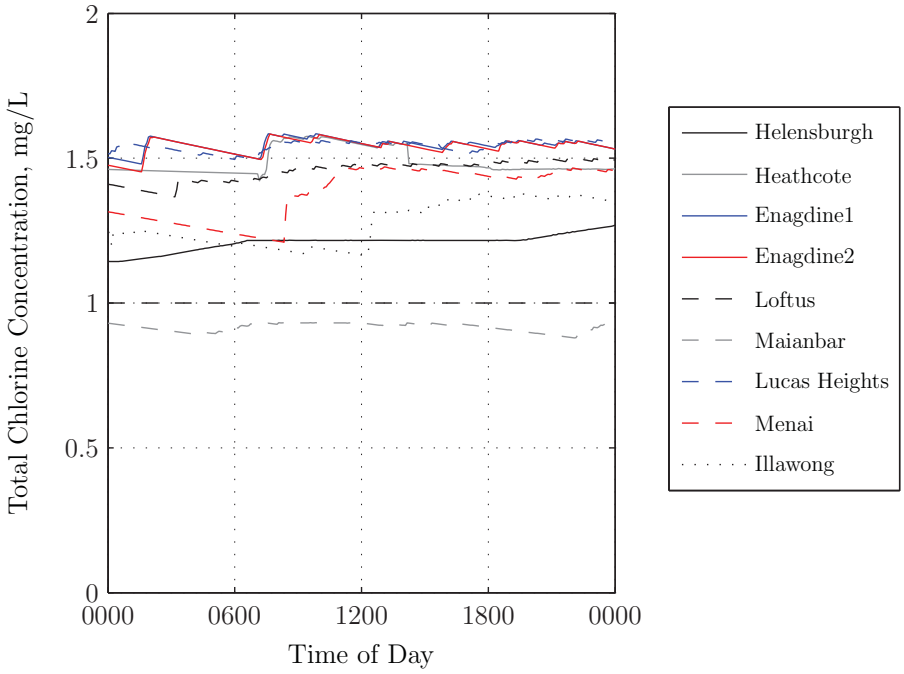


Figure 7.22 Total chlorine concentration at demand nodes with original initial reservoir levels

that satisfied this constraint for at least some of the simulation by increasing the penalty factor on this constraint, the initial value for the Maianbar demand node in Figure 7.22 is measured from the WDS, and therefore has been deemed to be appropriate. From Figure 7.22, it can be seen that the chlorine concentration at Maianbar remains around this initial concentration. Therefore, this solution has been determined to be suitable, as the alternative is to increase the penalty on the chlorine constraint, which would result in higher pumping costs to feed water from the WFP to the Maianbar reservoir to increase the total chlorine concentration.

Lower Initial Reservoir Levels The reservoir profiles for scenario 1 with the initial reservoir levels reduced to 70% full and the water quality constraint can be seen in Figures 7.23–7.27. From Figure 7.23, it can be seen that the reservoir profile determined for scenario 1 is almost identical to that found for scenario 3, seen in Figure 7.17, as the original water level in scenario 3 for the Helensburgh reservoir was already very close to 70% full. However, the reservoir trigger levels to produce this reservoir profile can be seen to be very different for these two methods, as indicated by the grey dashed lines in Figure 7.23 and Figure 7.17. For most time steps, it can be seen from Figure 7.23 that only one of the trigger levels is active, for example before 0700 hours the only active control

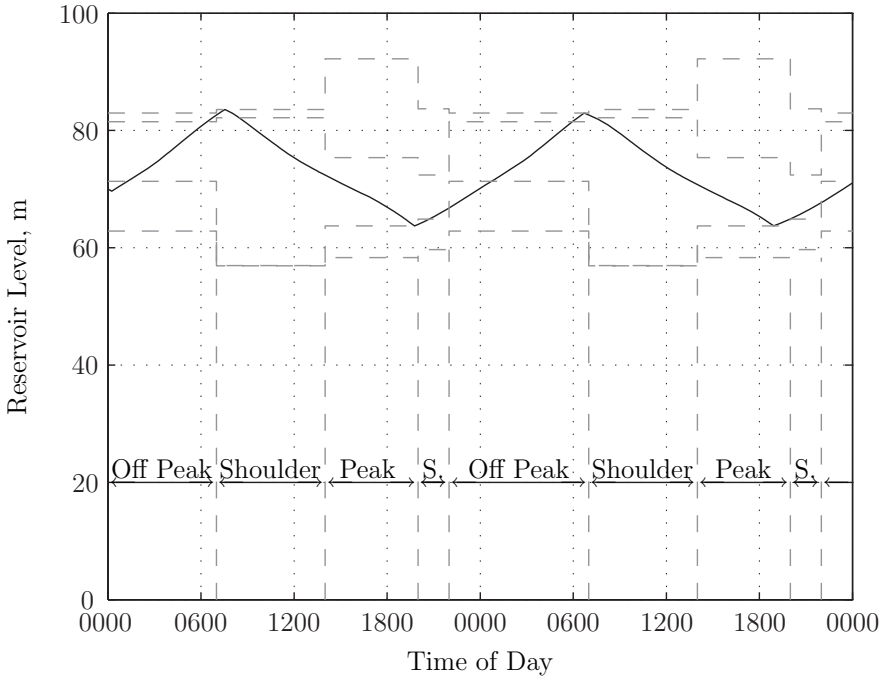


Figure 7.23 Helensburgh reservoir profile with low initial levels

turned on pump 2 at the Helensburgh pumping station, and between 0700 and 1400 hours the only trigger level that influenced the reservoir profile was the upper trigger level, to turn pump 2 off again. Therefore, as long as the other trigger levels do not interfere with the reservoir profile, they can take any value over the allowable range.

The reservoir profile for the Engadine reservoir with an initial reservoir level of 70% can be seen in Figure 7.24. While some pumping is still required between 1000 and 1200 hours, it can be seen from Figure 7.24 that by lowering the tank level overnight, the majority of the pumping required to fill the Engadine reservoir has been shifted into the period between 2200 and 0600 hours, when the cost of electricity is at its cheapest. The lower overnight reservoir level did not imply that the Engadine reservoir operated at a lower level, as it can be seen from Figure 7.24 that the reservoir was close to 100% full at approximately 0600 hours each morning.

From Figure 7.25, it can be seen that with the lower reservoir level at midnight, the Heathcote reservoir can be filled between 2200 and 0300 hours, compared to between 0700 and 1400 hours for the higher initial reservoir levels solution, seen in Figure 7.19. Therefore, the pumping required to fill the Heathcote reservoir has been shifted from the morning to overnight, when the cost of electricity is almost half, producing a lower cost to operate the Woronora WDS.

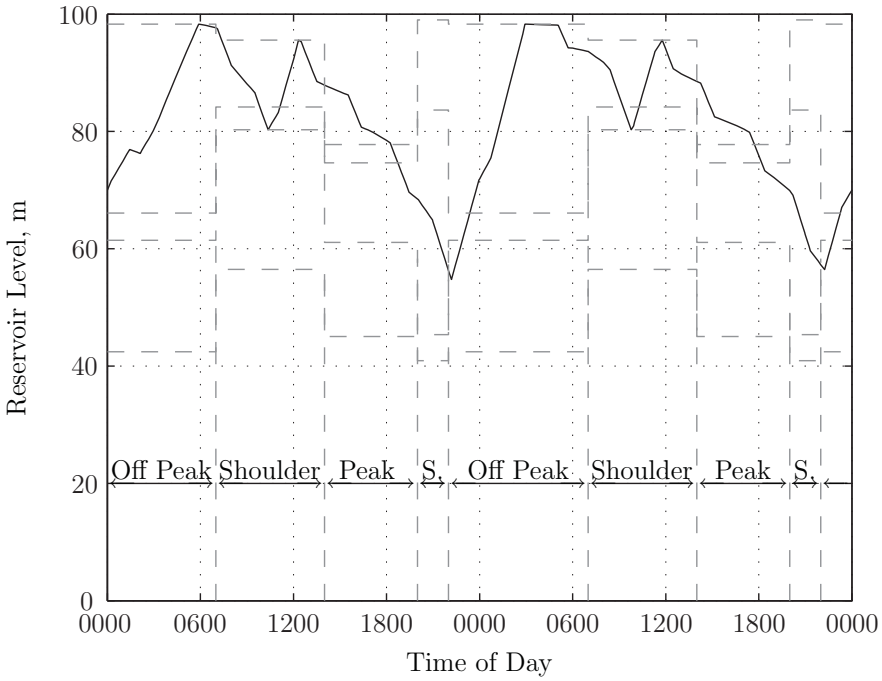


Figure 7.24 Engadine reservoir profile with low initial levels

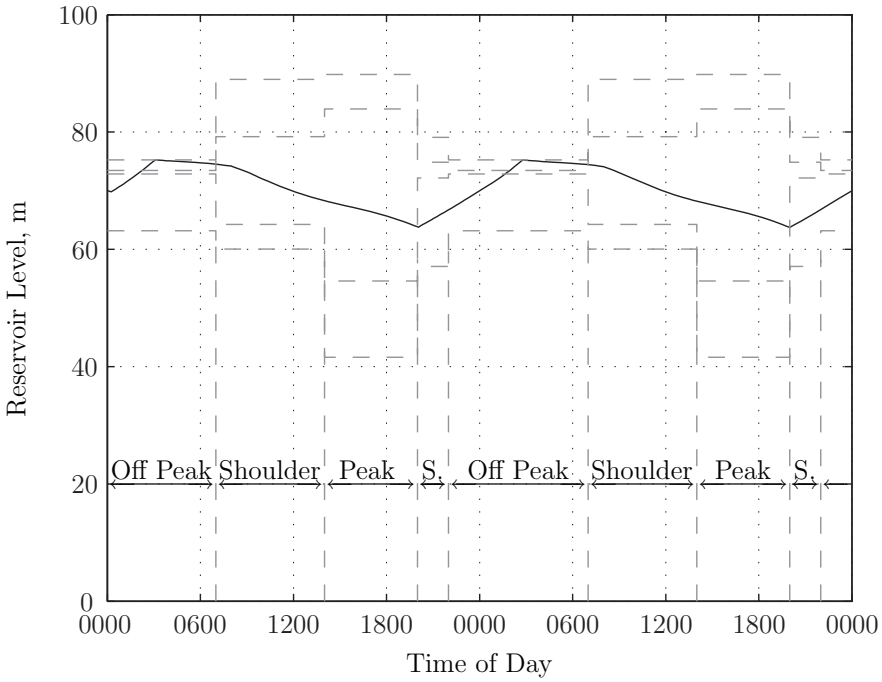


Figure 7.25 Heathcote reservoir profile with low initial levels

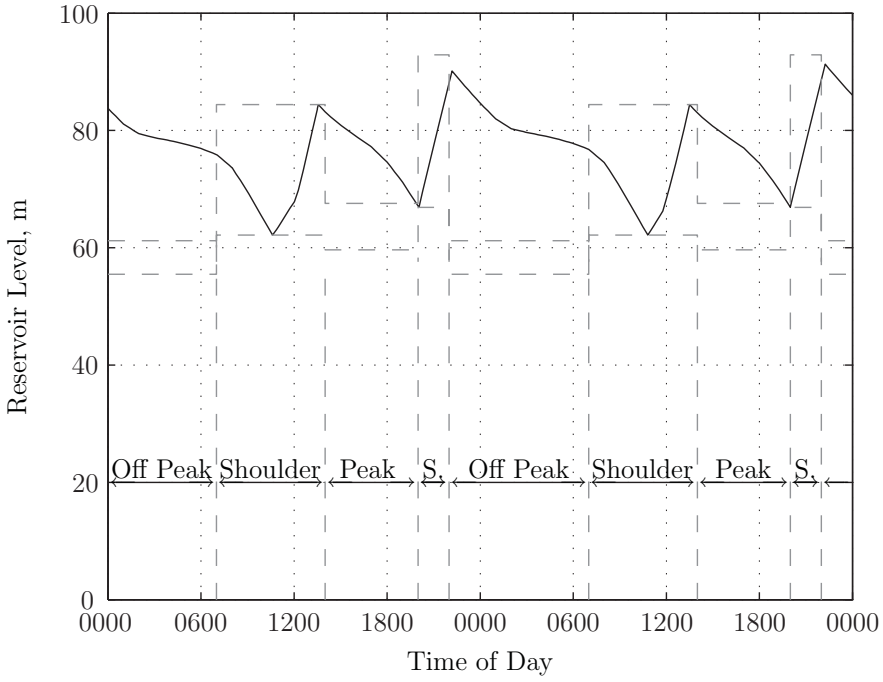


Figure 7.26 Lucas Heights reservoir profile with low initial levels

The two reservoirs that are gravity fed from the WFP were located at Lucas Heights and Menai. The reservoir profile resulting from the best solution found for scenario 1 of the Woronora fitness function can be seen in Figure 7.26 for the Lucas Heights reservoir and in Figure 7.27 for the Menai reservoir. The results suggest that for this scenario, it is best to fill the Lucas Heights reservoir twice a day, once at approximately 1200 hours and again at 2000 hours. These times correspond to the times directly after the Engadine pumping station switches off at 0000 hours, and directly before the Engadine pumping station comes back on at 2200 hours. The best time to fill the Menai reservoir was determined to be 0700 hours, six hours later than it was filled for scenario 3 with higher initial reservoir levels. This result is due to most of the pumping required in the system occurring overnight before 0700 hours for the case of lower initial reservoir levels, and therefore the Menai reservoir is filled later in the day, once the pump stations have turned off.

The total chlorine concentration produced by the best solution found to scenario 1 of the Woronora fitness function can be seen in Figure 7.28. The concentrations are very similar to those produced for the higher initial reservoir levels seen in Figure 7.28, where at most of the demand nodes, the chlorine concentrations are much higher than the minimum concentration of 1 mg/L. The initial chlorine concentration at Maianbar is again

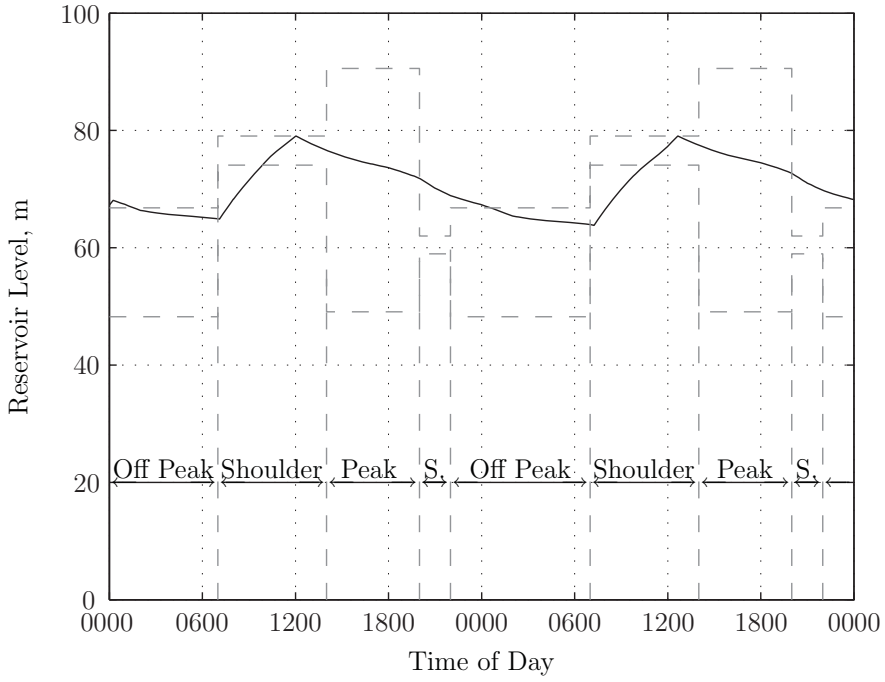


Figure 7.27 Menai reservoir profile with low initial levels

below 1 mg/L, however, by the second day of the simulation, the chlorine concentration at the Maianbar demand node can be seen to increase to the minimum constraint level of 1 mg/L. This can be explained by the pumping at the Engadine pumping station occurring earlier in the day, and therefore water from the WFP reaches the Maianbar reservoir quicker than was occurring for the higher initial reservoir levels seen for scenario 3. Therefore, as the water reaches Maianbar quicker, there is less decay of chlorine in the system, and a higher chlorine concentration results.

7.3 Discussion

This work adds to the significant body of literature that has applied GAs to the optimisation of WDS. The best solution found for the Cherry Hill-Brushy Plain system by the GA was slightly better than the solution found by the Linear Programming method used by Propato and Uber (2004). While the two solutions are very similar, the main advantage of the GA over traditional mathematical optimisation methods is that the methods can be applied to much more complex WDSs and fitness functions, such as the fitness function for the Woronora WDS considered in the second section of this chapter, without any change to the algorithm.

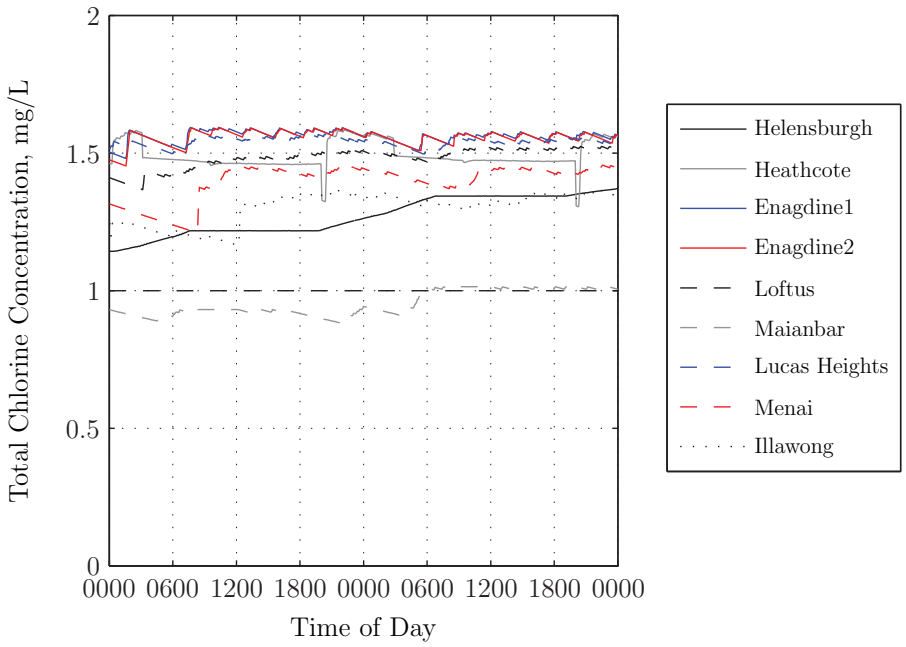


Figure 7.28 Total chlorine concentration at demand nodes with low initial reservoir levels

By applying the GA to optimise the operation of the Woronora WDS, solutions were found to save between 20-30% of the daily operating costs for the utility responsible for the system. This result highlights the fact that GA optimisation of WDSs can identify much more effective solutions than can be determined by trial and error, either in a modelling environment or in the actual WDS. The solutions found suggest that by allowing the reservoir levels to be lower overnight, more pumping can be shifted to the off-peak periods, resulting in a greater reduction in the pumping costs involved in operating the system.

In Chapter 6, the proposed GA calibration method based on the characteristics of the fitness function was found to produced the best overall results for a number of test functions and stopping criteria. While the mathematical functions considered in that chapter were not simple, the results presented in this chapter indicate that the Predicted GA calibration method can also be extended to the optimisation of more irregular WDS optimisation functions, as the best overall results were again obtained using the Predicted GA calibration method.

The ranking of the performance of the different GA calibration methods tested for the two WDS optimisation problems considered in this chapter was very similar to that found in Chapter 6, where the Predicted method with set parameter values located the best solutions, and the Parameterless and Drift methods with set parameter values also

provided good parameter values for the GA. Again, the typical GA parameters produced the worst results of the four methods of determining the GA population size considered. Another similar result to that observed in Chapter 6 was that for the WDS optimisation problems, the calibration methods that adopted self-adaptive values for the p_m, p_c, and c parameters produced the worst overall results of all the methods considered.

The main disadvantage to the Predicted GA calibration method proposed in this work is that the fitness function statistics must be computed prior to the optimisation of the problem to determine the population size. However, it is not possible to quantify the convergence of the GA without some information about the fitness function. The Parameterless calibration method is much easier to implement than the Predicted method, as in this case, the information about the fitness function is obtained as the GA is solving the problem, as the population size is increased only after a smaller population size has converged. However, the Parameterless method is not straight forward to apply either, as a number of modifications must be made to the GA to allow the population size to be doubled automatically, and to inject the best solution from the previous GA run into the initial population. The main disadvantages of the Parameterless GA calibration method are that: 1) multiple restarts of smaller population sizes are used, as opposed to one run with a larger population size, and 2) a deterministic rule is used to alter the population size, as in some cases the GA may benefit from a decrease to the population size, as opposed to an increase, as discussed in Section 6.5.

The Drift GA calibration method performed as well as the Predicted method for the Cherry Hill-Brushy Plains fitness function, and this approach was only slightly outperformed by both the Predicted and Parameterless methods on the Woronora fitness function. The Drift method has the advantage that the population size is can be to determined very easily, and therefore may provide a simple alternative to the Predicted calibration method to determine a suitable population size for applications where it is not feasible to compute the fitness function statistics.

A large scale GA parametric study, similar to that undertaken in Chapter 3, was applied to the Cherry Hill-Brushy Plain fitness function by Gibbs et al. (2005). The results from that study are not directly comparable to results presented in this section, as the GA implemented is slightly different, and only five different random number seeds were considered in that study, compared to 13 used in this work. A total of 378 different combinations of GA parameter values were considered in the parametric study, where the best solution found added a total of 1163 g/day of chlorine to the system, and the average fitness function value over the five different GA runs was 1184 g/day. By making use of

the Drift and Predicted calibration methods developed in this thesis, only one combination of the GA parameter values was tested for each method, as opposed to the 378 used by Gibbs et al. (2005). While the chlorine dosing rates found in this work were slightly higher, with a best fitness function value of 1168 g/day, and average fitness function values of 1196 g/day for the Drift method and 1197 g/day for the Predicted method, the solutions found are very similar to those found by the full GA parametric study, with a much smaller computation requirement to identify a similar solution quality.

Application of the fitness function statistics to the WDS fitness functions considered in this chapter provided a number of insights into the characteristics of these functions that are generally largely unknown. For the Woronora WDS fitness function, the results from the spatial correlation measure suggested that the search space was much less correlated than many of the other Optimal Generation Functions that have been considered in this work, with $R_{\text{av}} \approx 0.125$ compared to $R_{\text{av}} \approx 0.25$. The correlation in the fitness function can be used to determine the expected influence of the GA parameters, where for the more correlated functions, the calibration of the GA parameters had a large effect on the quality of the final solutions found. However for the Woronora fitness function with a lower value of R_{av}, the GA parameter values had a much smaller influence on the solutions found.

This result also agrees with the flowchart used to determine if a function is an Optimal Generation Function or a Maximal Generation Function based on the structure in the fitness function, given in Figure 3.13 in Section 3.1.3. The results suggested that for a function with no correlation in the fitness function, the GA parameter values had very little influence on the performance of the GA, however generally slightly better solutions were found with a small population size. For functions that were highly correlated, the calibration of the GA parameters was much more important to the final solution found by the GA. Therefore, if a function is highly correlated, it may be beneficial to determine the best GA parameter values possible from the Predicted method, otherwise if the function is less correlated, the simpler Drift method may provide a suitable alternative to determining the GA parameter values.

For the Cherry Hill-Brushy Plains fitness function, the separability measure indicated that for feasible solutions, the fitness function was completely separable. Intuitively, this is not an obvious result, as it would be expected that the best mass of chlorine to be dosed at one location would be strongly influenced the best mass of chlorine dosed at an upstream location. These interactions between the decision variables are detected for infeasible solutions, and once the chlorine concentrations became important to the constraints on the fitness function, a number of interactions were identified between the decision

variables for the Cherry Hill-Brushy Plains system.

The separability measure was extremely computationally intensive to compute, and in practice, the resources spent on determining the interaction between the decision variables may be better spent actually optimising the fitness function. The information provided by the measure could be used to encourage better solutions to be found, as if interacting decision variables are located near each other in the solution string, the GA will have a greater chance of locating good combinations of values. However, it may be that online linkage learning methods, such as those reviewed in Section 2.2.5, provide more efficient learning of the interactions between the decision variables. For most realistic problems, such as the WDS optimisation problems considered in this chapter, it may be a reasonable assumption that there are epistatic interactions between the decision variables, and therefore it is only the correlation in the search space that must be estimated to determine if a problem is an Optimal Generation Function or a Maximal Generation Function.

Both of the WDS optimisation problems considered in this work have constraints on the objective of the fitness function, which is the case for almost all WDS optimisation problems. It is these constraints that make the optimisation process difficult, otherwise the solution would generally be very simple to determine, either to dose the smallest amount of chlorine, undertake all of the pumping required overnight when electricity is the cheapest, or in the case of a design problem, install all of the smallest pipe sizes. The penalty factors used in this work are the most common approach used to take the constraints on a problem into account, however, there is very little guidance available to determine suitable values. Many methods have been proposed to take the constraints on an objective into account in the optimisation process, a number of which are outlined in Section 2.1.1. However, at the present time, there is not an accepted best approach to consider all the constraints on the objective of an optimisation problem, and this area should be the focus of future research.

It is likely that better solutions could be easily obtained by fine tuning the dosing rates found from the GA optimisation of the Cherry Hill-Brushy plain system. As the fitness function is separable in all decision variables for feasible solutions for this problem, a simple local search algorithm could be applied after the GA to improve the best solution found and determine if even smaller amounts of chlorine can be added to the system. The mutation operator used to determine better solutions for each decision variable is relatively inefficient, and a local search operator applied during or after the GA optimisation may also be beneficial for more complex problems, such as the fitness function for the Woronora WDS. As provided by the No Free Lunch theorem (Wolpert and Macready,

1997), problem specific GA operators may also lead to better solutions in a number of cases. An example would be the use of creeping mutation, which tends to mutate smaller doses of chlorine for the Cherry Hill-Brushy plain network.

Many of the values determined for the decision variables for Woronora fitness function are not binding. For example, if a pump does not come on for a given time period, then the trigger level to turn the pump off can take a value anywhere over the range of the decision variables, provided that the order of trigger levels required is satisfied. If the demands used for the optimisation of the system are significantly different from the true demands in the system, the optimised reservoir profile will change, and the trigger levels that were not important in the simulation may become more significant in reality. One method to ensure reasonable solutions are implemented would be for an experienced operator of the system to determine reasonable trigger levels in the event of unforeseen demands. Another approach to identify more reliable solutions would be to incorporate the uncertainty into the optimisation, and simulate each proposed solution over a range of demands, to consider all of the potential scenarios that could occur. However, this approach would require multiple runs of the simulation model for the evaluation of each solution, adding to the already computationally expensive process of running hydraulic and water quality models.

7.4 SUMMARY

The GA calibration method based on the characteristics of the fitness function proposed and tested in Chapter 6 has been applied to the optimisation of WDSs in this chapter. The results presented indicate that the proposed GA calibration method can be extended to provide very good GA parameter values for the optimisation of more irregular WDS optimisation functions, as the best results of the eight calibration methods considered were obtained using the proposed GA calibration method.

The ranking of the performance of the different GA calibration methods tested for the two WDS optimisation problems considered in this chapter was very similar to that found in Chapter 6, where the Predicted method with set parameter values located the best solutions, and the Parameterless and Drift methods with set parameter values also provided good parameter values for the GA. As previously found in this work, the calibration methods that adopted self-adaptive values for the p_m, p_c, and c parameters produced the worst overall results of all the methods considered. The ranking of the performance of the different calibration methods also indicate that the typical GA parameter values lead to poor GA performance, reinforcing the importance of determining the most suitable GA

parameter values for each fitness function optimised.

For the Cherry Hill-Brushy Plain fitness function, the solutions found by the GA calibrated by the Drift and Predicted methods were very similar to the solutions found in the large scale GA parametric study undertaken by Gibbs et al. (2005). While the mass of chlorine added to the system was slightly higher, the computational requirements to obtain these solutions are much lower when suitable GA parameter values are determined from one of the calibration methods presented in this work, compared to the full scale GA parametric study required to determine the best GA parameter values in that study. This result highlights the usefulness of the proposed calibration methods, as near optimal GA parameter values can be determined directly, and therefore only one GA run is required, as opposed to the 378 different combinations of parameter values tested by Gibbs et al. (2005).

By applying the GA to optimise the operation of the Woronora WDS, solutions were found to save up to 30% of the daily operating costs involved in running the system. The solutions found suggest that by allowing the reservoir levels to be lower overnight, more pumping can be shifted to the off-peak periods, and therefore a greater reduction in the pumping costs is achieved. This result reinforces the understanding that GA optimisation can provide much more effective solutions to WDS optimisation problems than can generally be determined by trial and error, either in a modelling environment or in the actual WDS.

Chapter 8
Conclusions and Further Work

8.1 Contributions of this Work

The following contributions have been made to the field of Real-Coded Genetic Algorithms in this thesis:

1. The empirical validation that the number of generations before a GA population will converge is constant. It was found in Section 3.1 that for fitness functions with certain characteristics, the number of generations before the GA population converged was constant with respect to the stopping criterion. Therefore, if the number of function evaluations (FE) available to solve the problem was increased, the population size was observed to increase accordingly. Fitness functions that possessed this characteristic were called Optimal Generation Functions. If a function was not solved with a constant number of generations, it was called a Maximal Generation Function, as the best solutions for these functions were found with a small population size, run for as many generations as possible. The function characteristics that were identified to determine which type of function a given fitness function belonged to were:

    - Structure in the fitness function. If a function was essentially flat with no structure to lead the GA toward better solutions, it is a Maximal Generation Function.
    - Epistatic interactions between some or all of the decision variables. If a fitness function was structured and there was at least one interaction between the decision variables, the GA benefited from the implicit parallelism in a larger population size, and was therefore an Optimal Generation Function.
    - Highly salient decision variables. If at least one decision variable had a much larger influence on the fitness function value, and the fitness function was

somewhat structured, then the GA benefited from the diversity in a larger population, and was therefore an Optimal Generation Function.

- If a function did have structure in the fitness function but did not have epistatic interactions or highly salient variables, then it was found to be a Maximal Generation Function. In this case, the function is separable, and as each decision variable has a similar contribution to the fitness function value, each variable can be optimised independently of the others, and function evaluations are wasted in a larger population.

2. The observation that the number of generations before convergence was related to the characteristics of the fitness function. In Section 3.2, controllable changes were made to the characteristics of the Optimal Generation Functions identified in Section 3.1, to investigate the effect on the number of generations before the GA population converged. It was determined that the number of generations before convergence observed was related to the characteristics of the fitness function, where:

   - In general, it was observed that an increase in the roughness of a function produced an increase in the number of generations before the GA converged, g_{conv}. The implication of this result is that a smaller population size is better for functions with greater roughness. For rougher functions, the GA is relying on mutating the best solutions, as there is no structure in the fitness function to benefit the search, and a larger population size will only waste function evaluations.
   - An increase in the multimodality of the function produced a slight decrease in the value of g_{conv}. For a given FE, a decrease in g_{conv} indicates that a larger population size performed better for functions with more local optima. This result can be explained by the fact that a larger population size will contain more solutions distributed over the search space, and therefore will be less likely to become trapped in the local optima that are present.
   - An increase in the salience of the decision variables produced a increase in g_{conv}. The consequence of this result is that for an available number of FE, a smaller population size is better for functions that have a higher salience of at least one decision variable. It might be assumed that a larger population size would be more beneficial for a more salient problem, so that there is more diversity available in the population once the more salient decision variables

have converged, to allow the remaining variables to be optimised and avoid the occurance of genetic drift. However, mutation can be used to inject diversity into the population, and when only a small percentage of the decision variables have a significant effect on the fitness function value, function evaluations are wasted when a large population size is used.

- A change in the degree of interaction between variables did not have a significant effect on g_{conv} for the cases considered (Section 3.2.2). While the results suggest that the degree of interaction between the decision variables had little effect on g_{conv}, this characteristic did have a large effect on the solution quality found. The number of interactions was observed to have a large impact on GA performance, where for problems with a large number of interactions, the GA found it much more difficult to find better solutions. The distance between the interacting variables in the solution string also had a slight impact on GA performance, where the GA was able to find slightly better solutions when the distance between the interacting variables was shorter. However, the difference was negligible when compared to the impact of an increase in the number of interactions, m_{BB}.
- Of all the problem characteristics considered, problem size had by far the largest effect on the value of g_{conv} (Section 3.2.2).

3. The development of a number of statistics to provide accurate information about the characteristics of a fitness function. Statistics were developed to provide information about each of the characteristics identified in Contribution 2, namely:

   - A spatial correlation measure was developed to characterise the roughness and multimodality of a fitness function. A correlation measure based on the distance between solutions in the search space was proposed, as opposed to the more common approach of using a series of fitness function values to represent the distance between solutions. The two approaches to computing the correlation of a fitness function were applied to functions with a known correlation function, and therefore the results from the different sampling methods could be compared to the true correlation. From these analyses, it was concluded that the traditional temporal correlation statistic is unable to represent a multi–dimensional space as a one dimensional series of function values, as the approach lead to an over estimation of the true correlation of a function. The proposed spatial correlation statistic provided a more accurate represen-

tation of the search space, and therefore a more accurate estimation of the true correlation of a function.

- A separability measure to identify epistatic interactions between decision variables. Mutual Information was identified as a useful statistic to determine interactions between decision variable by Seo et al. (2003). However, the experiments conducted in this thesis suggest that there are some serious shortcomings when applying the gene epistasis measure in practice. The separability measure was proposed to determine epistatic interactions between pairs of decision variables, and as the proposed statistic is based on the distribution of values produced by the difference from changes to pairs of decision variables, it can be accurately applied to any fitness function. Higher order interaction can also be identified.
- A dominance measure to determine the presence of any highly salient decision variables in the fitness function. The results presented in Section 4.3 indicate that the proposed statistic is a suitable method for comparing the effect of each decision variable on the fitness function value, as the results agreed with the known characteristics for the functions considered.

4. A relationship was developed to predict the number of generations before convergence based on the characteristics of the fitness function. This is the most important contribution of this work for a number of reasons:

    - The results suggest that the fitness function statistics developed are useful for the calibration of GA parameter values. Many statistics to provide information about different fitness function characteristics have been developed previously, however, for each example that indicates that any of these measures provide information about the difficulty of an optimisation problem, there appear to be as many counter examples displaying that statistic's unreliability. Generally, other studies have concluded that fitness function statistics are a poor indicator of GA performance. However, these studies did not consider the calibration of the algorithm, and hence the convergence observed was rather arbitrary. The relationship developed has demonstrated that the number of generations before a GA will converge is a function of the characteristics of the problem, which fitness function statistics provide information about. This approach is different to that generally adopted when making use of these statistics, which typically attempts to relate fitness function statistic values to how closely a

GA will converge to the optimum solution. However, this is not only a function of the fitness function characteristics, but also the GA parameters and the convergence criterion.

- The relationship characterises the selection pressure that can be expected to decreases the variance in the population by producing multiple copies of fitter solutions. As the selection operator is based on the fitness function values, this selection pressure is directly related to the characteristics of the fitness function. The analytic change in population variance due to genetic drift was identified by Rogers and Pruegel-Bennett (1999), however, an analytic expression for the change in population variance can not be developed, as it is dependent on the fitness function. The relationship developed in this thesis has provided an empirical relationship to determine the decay in population variance due to the selection pressure.
- Most importantly, in terms of the application of GAs, the relationship developed allows the most important GA parameter value, the population size, to be estimated. This result is based on the assumption that there is a fixed FE available to solve the problem. From the initial and final population variance, the number of generations before convergence can be determined from the change in population variance predicted from the fitness function statistics. Hence, from the number of function evaluations that can be made before a solution is required, the population size can be calculated by $N = FE/g_{\text{conv}}$.

5. A complete GA calibration method based on the characteristics of the fitness function has been developed. The previous contribution provided the population size based on the fitness function characteristics. However, this is only one of the GA parameter values that must be set before the GA can be applied. In Chapter 6, the relationships between the GA parameter values were identified and taken into consideration to propose a full GA calibration method in Section 6.2. The proposed calibration method has been compared with a number of other GA calibration methods that completely remove the calibration of any parameters from the user, including another method to determine the population size proposed in this thesis, based on the time to convergence due to genetic drift. A total of 20 difficult mathematical fitness functions, as well as two different WDS optimisation problems, have been used as a basis for the comparison of the GA calibration methods. In almost all individual cases considered, the proposed GA calibration method leads the GA to find the best solutions, and hence was the best overall GA calibration method

considered. This result reinforces the understanding that the GA parameter values must be calibrated to each individual fitness function, and that the GA calibration method proposed in this work provides an effective technique to determine the most appropriate parameter values from the fitness function characteristics.

8.2 Conclusions

1. It has been well established that the values used for the GA parameters are critical to the quality of the final solutions found by the GA. This fact has again been highlighted in this work, as the typical GA parameter values tested generally located solutions of significantly poorer quality when compared to the other calibration methods that made use of some information about the fitness function. The GA calibration method developed in this thesis is explicitly based on the characteristics of the fitness function, and therefore it is not surprising that it produced the best results of the different methods tested. Therefore, based on the results presented in this thesis, if the best solutions to a given fitness function are desired, it is recommended that the proposed fitness function statistics be applied to characterise the fitness function, allowing the most suitable GA parameter values to be determined. This approach to determining the most suitable GA parameter values effectively removes the need to undertake repeated runs of the GA in an attempt to determine reasonable GA parameter values.

2. One of the calibration methods proposed in this work was based on the modelling of Rogers and Pruegel-Bennett (1999) of when a GA population will converge due to genetic drift. This result was extended in this work to provide a very simple method for determining a suitable population size, as the only requirement is to solve Equation 6.1. The only function characteristic that is taken into account in this equation is the problem size, which was found to have the greatest effect on g_{conv}. This approach can be considered to provide a lower limit to the population size, as the population size that is expected to converge due to genetic drift is determined. However, if the selection pressure is higher, a larger population size will also converge in the time available, and potentially find better solutions. However, in many cases, the difference in the final solution found by the GA with a larger population size may not be significant, as generally the Drift method produced similar results to the Predicted method in Chapter 6. Therefore, the Drift calibration method developed in this thesis can be used to provide a simple estimation of a

suitable population size for situations where it is not feasible to compute the fitness function statistics.

3. Another general conclusion made in this work is that, provided the fitness function has certain characteristics, one GA run with a larger population size generally produced better results than a number of GA runs with a smaller population size for the same number of fitness function evaluations. The results presented in Chapter 6 demonstrated that the population sizes determined from both the Drift and Predicted methods generally produced better results than the Parameterless method. Other studies, such as Cantú-Paz and Goldberg (2003), have come to the same conclusion. Generally, it is understood that a larger population size will locate better solutions, provided it has time to converge to a solution. Therefore, the question that arises is how large a population can be used to locate the best solutions possible in the time available? The Drift and Predicted calibration methods presented in this work have provided methods to answer this question. If a simple approximation to the population size is required, the Drift method is recommended. However, if the best possible solutions are desired, then the increased effort required to determine the population size from the characteristics of the fitness function may be worthwhile.

4. The optimisation results presented in this thesis indicate that a self-adaptive framework for GA parameters that are applied on a solution level (p_m, p_c, c) is less effective than set parameter values used for the whole GA run, provided reasonable values are used. The results of Bäck et al. (2000) support this finding, where in their empirical study the performance of self-adapting p_c and p_m was found to be disappointing, and adapting the population size was the key to improving the GA performance. The exception to this rule was for the typical GA parameter values, where self-adaptive parameter values performed better than the set values used. This result suggests that the set values used were not appropriate, and allowing the values to self-adapt provides a mechanism to determine better solutions. In Chapter 6, a method has been developed to determine suitable values for GA parameters other than population size. For the calibration methods that implemented this method to determine parameter values, the GA consistently outperformed the GA with self-adaptive parameter values.

5. The fitness function statistics developed in this thesis have been shown to provide useful information about the characteristics of a fitness function. Each of the statis-

tics was applied to functions with known characteristics in Chapter 4, and each statistic was confirmed to provide accurate information. Concluding remarks based on the results presented in this thesis regarding each of the statistics developed are provided below.

- The spatial correlation statistic was found to compare very favourably to the true autocorrelation for functions with known autocorrelation. Not only was the spatial correlation statistic accurate, but it was also able to identify changes in the roughness, and to a lesser extent, the multimodality of a fitness function. The correlation statistic provided information about two important aspects regarding GA parameter values:
  - The degree of correlation provided an indication of the expected convergence of the algorithm. An uncorrelated fitness function will eventually converge due to genetic drift, where a highly correlated function will converge to one solution much quicker, and therefore a larger population size can be used to identify better solutions over the same number of function evaluations.
  - The degree of correlation provided an indication of how influential the GA parameter values will be on the solution quality found. For an uncorrelated fitness function, the GA parameter values will have very little influence on the final solution found by the GA, whereas for a highly correlated fitness function, the GA parameter values are much more important to the final solution quality found.
- The proposed dominance statistic was found to accurately characterise the contribution of each decision variable to the fitness function value. However, as the statistic is based on the normalised MI between one decision variable and the fitness function value, it is unlikely that highly salient combinations of decision variables will be identified by the proposed statistic. It may be possible to determine the salience of combinations of decision variables, however this would require higher dimensional pdfs to be estimated, and would dramatically increase the number of combinations of decision variables to be computed, increasing computational requirements.
- The separability measure was proposed to identify epistatic interactions between decision variables. By testing the statistic on functions with different numbers of interactions, and distance between the interactions, it was found

that the measure was able to very accurately determine the location of any epistatic interactions between the decision variables for a fitness function. However, as each pair of decision variables must be considered in turn, the computation requirements to calculate the separability measure increases dramatically as problem size increases. The separability measure was not required to determine the population size in the proposed GA calibration method, as the degree of epistatic interactions between the decision variables for a fitness function was not found to influence the most suitable GA parameter values. The main use for the separability measure is then to provide information about how to rearrange the solution string, to allow more efficient processing of the building blocks. However, in Section 3.2.3, the distance between the interactions was not found to have a great influence on the final solution quality found by the GA, and the number of interactions was much more important. There is very little that can be done to reduce the number of epistatic interactions in a given fitness function, therefore the effort required to compute the separability measure is most likely better directed toward the actual optimisation of the fitness function.

8.3 Recommended Future Work

1. It has been assumed in this thesis when computing the spatial correlation that the fitness function is a reasonable approximation to the true fitness landscape. This assumption implies that the distance between two solutions in the search space is a close approximation to the likelihood of one solution being produced from the other solution by the GA operators. Strong relationships were obtained between the spatial correlation of a fitness function and the observed convergence of the GA, indicating that this assumption was valid for cases considered in this work. However, the fitness function is still only an approximation to the fitness landscape, and more accurate results would most likely be obtained by using the true fitness landscape to compute the distance, or likelihood of being produced, between two solutions for the spatial correlation measure. An analysis such as this may also provide an insight into the best GA operators to use, or even the best EA to use, as if a certain operator produced a more correlated fitness landscape, then it may be more likely to find better solutions.

2. The most pressing future work arising from this study is to develop the theoreti-

cal knowledge of the fitness function statistics that have been proposed. The work undertaken in this study has advanced the understanding of the contribution fitness function statistics have in the field of GAs, where prior to this study the general consensus was that statistics such as these do not provide a reliable representation of the expected performance of a GA. This work has related the information provided by fitness function statistics to GA calibration, not GA performance. Now that this relationship has been identified, it has opened the way for more rigorous definitions of the mechanisms of the statistics themselves. Following this, theoretical analyses should consider the relationship between the information provided by the statistics and the change in population variance. The form of the empirical relationship identified in this work was based on the modelling of GAs, however a more theoretical understanding is required to provide a more generalised form for the relationship proposed in Equation 5.9.

3. It was assumed in this work that the only change in population variance was due to the selection operator. While the influence of the other GA operators, such as crossover and mutation, are likely to be much less significant to the change in population variance compared to the selection operator, they will have some influence. Future work should aim to include these operators in the calculation of the change in population variance in each generation, to provide a more accurate estimation of the number of generations before the GA population will converge. Similarly, the only GA coding scheme considered in this work was real coding. Further work should also investigate if similar results are obtained for GAs with other encodings, such as binary and integer.

4. For a number of the fitness functions considered in this thesis, it was observed that the characteristics of the fitness function were different in different regions of the search space. The values computed for the fitness function statistics computed over the complete search space provide a representation of the fitness function the GA is presented with at the start of the optimisation. However, as the GA converges to smaller regions of the search space, the characteristics of the fitness function may change significantly. To address this issue, it is recommended that an adaptive approach to the GA calibration be undertaken, where as the GA convergences to smaller regions, the spatial correlation and dominance measures could be re-computed from solutions that have already been evaluated by the GA, and therefore would not significantly increase the computational requirements for computationally expensive fitness functions. Therefore, the GA parameter values could

be adjusted based on the characteristics of the region of the search space the GA is currently working on. However, more research is required to implement a method such as this.

5. Two aspects related to the optimisation of WDS by GAs were identified in Chapter 7 that should be the focus of future research. The first concern that arose was how to compare feasible and infeasible solutions, or two infeasible solutions, for the selection operator. The approach that is implemented will have a significant influence on the quality of the final solution found by the GA. While there are a number of constraint handling methods available, there is still not an accepted approach for handling constraints on the objective of a fitness function. The second aspect identified to improve the solutions found by the GA was to include the uncertainty in the model into the optimisation process. A number of decision variables for the Woronora case study were found to be not important for the demand case considered. However, if the true demand was slightly different to the simulated demand, the value used for these decision variables could be much more important. By incorporating this uncertainty into the model when a potential solution is evaluated, factors such as this could be taken into account, and much more reliable solutions could be identified by the GA.