

# **PARALLEL NICHING GENETIC ALGORITHMS: A CROWDING PERSPECTIVE**

BY

JASON BROWNLEE

B.A.S., Swinburne University of Technology, 2002

MINOR THESIS

Submitted in partial fulfilment of the requirements for the degree of Master of  
Information Technology

Centre for Intelligent Systems and Complex Processes  
School of Information Technology  
Swinburne University of Technology  
2004

Victoria, Australia

© Copyright by  
Jason Brownlee  
2004

## **Abstract**

Genetic algorithms are a class of global search algorithm inspired by biological evolution that return approximate solutions to difficult search problems. Parallel niching genetic algorithms are a specialised type of genetic algorithm that search for and maintain multiple approximate solutions by grouping subpopulations of candidate solutions around areas of interest in the search space. Crowding is one way of achieving a niching effect based on the concept of localised competition for limited resources where similar candidate solutions are matched together and compete for survival based on solution usefulness. An abstract conceptual model called the generalised crowding model is provided to describe discussed traditional crowding algorithms. It provides a template for existing crowding algorithms and a framework for crowding algorithm based analysis and development. A simple crowding algorithm is proposed that has no forced biases or similarity approximations and is shown to be an embodiment of the generalised crowding model and the redefined crowding principle of localised generational competition. The simple crowding algorithm is analysed using crowding centric measures, revealing insights into unbiased parental pairing, candidate solution matching and actual replacement behaviour. Analysis of common crowding functionalities on the simple crowding algorithm and comparison between the proposed algorithm and other crowding techniques reveals a practical trade-off between decreased time until convergence at the cost of decreased niche stability.

## Table of Contents

<b>ABSTRACT.....</b>	<b>I</b>
<b>TABLE OF CONTENTS .....</b>	<b>II</b>
<b>LIST OF FIGURES .....</b>	<b>IV</b>
<b>LIST OF TABLES .....</b>	<b>V</b>
<b>LIST OF EQUATIONS.....</b>	<b>V</b>
<b>INTRODUCTION.....</b>	<b>1</b>
CANONICAL GENETIC ALGORITHM .....	2
NICHING GENETIC ALGORITHM .....	7
<b>METHODS .....</b>	<b>12</b>
METHODOLOGY .....	12
TEST PROBLEMS.....	14
MEASURES .....	16
<b>PREVIOUS RESEARCH.....</b>	<b>18</b>
CROWDING FACTOR MODEL .....	18
DETERMINISTIC CROWDING .....	20
RESTRICTED TOURNAMENT SELECTION .....	23
MULTI-NICHE CROWDING .....	25
PROBABILISTIC CROWDING.....	27
DISCUSSION .....	29
<b>QUINTESSENTIAL CROWDING.....</b>	<b>31</b>
CROWDING.....	31
GENERALISED CROWDING MODEL.....	32
DISCUSSION .....	36
<b>QUINTESSENTIAL CROWDING INVESTIGATED.....</b>	<b>39</b>
SIMPLE CROWDING ALGORITHM.....	39
NATURE OF REPLACEMENTS .....	44
HYBRIDS AND NORMALS IN SEARCH.....	47
REPLACEMENT WASTE AND REPLACEMENT ERROR.....	49
DISCUSSION .....	53
<b>AUGMENTED SIMPLE CROWDING .....</b>	<b>56</b>
BIAS IN UPFRONT SELECTION .....	56
SAMPLING IN LOCALISED SELECTION .....	60
ALTERNATIVE REPLACEMENT STRATEGIES .....	63
DISCUSSION .....	65
<b>SIMPLE CROWDING COMPARED .....</b>	<b>68</b>
MAINTENANCE OF DIVERSITY .....	68

PROMOTION OF USEFUL DIVERSITY .....	70
TIME UNTIL CONVERGENCE .....	71
LOCALISATION MATCHING BEHAVIOUR .....	72
REPLACEMENTS BEHAVIOUR .....	73
REPLACEMENT ERROR BEHAVIOUR .....	74
DISCUSSION .....	74
<b>CONCLUSION .....</b>	<b>76</b>
SUMMARY .....	76
FUTURE RESEARCH .....	77
CONCLUSIONS .....	83
<b>BIBLIOGRAPHY .....</b>	<b>85</b>
<b>APPENDIX A – TEST FUNCTIONS .....</b>	<b>87</b>
F1 - SINE FUNCTION - 5 PEAKS OF EQUAL HEIGHT .....	87
F2 - SINE FUNCTION - 5 PEAKS OF DIFFERING HEIGHT .....	87
F3 - SCALED SIX-HUMP CAMEL BACK FUNCTION .....	88
F4 - HIMMELBAU’S FUNCTION .....	89
F5 - SCHWEFEL’S FUNCTION .....	90
<b>APPENDIX B – TEST RESULTS .....</b>	<b>91</b>
INVESTIGATION INTO QUINTESSENTIAL CROWDING .....	91
AUGMENTED SIMPLE CROWDING .....	91
SIMPLE CROWDING COMPARED .....	91
<b>APPENDIX C – GENERALISED CROWDING MODEL .....</b>	<b>93</b>

## List of Figures

FIGURE 1 - PSEUDO CODE LISTING OF THE CANONICAL GENETIC ALGORITHM .....	4
FIGURE 2 - PSEUDO CODE LISTING OF THE CROWDING FACTOR MODEL .....	19
FIGURE 3 - PSEUDO CODE LISTING OF THE DETERMINISTIC CROWDING TECHNIQUE.....	21
FIGURE 4 - PSEUDO CODE LISTING OF THE RESTRICTED TOURNAMENT SELECTION TECHNIQUE.....	24
FIGURE 5 - PSEUDO CODE LISTING OF THE MULTI-NICHE CROWDING TECHNIQUE.....	26
FIGURE 6 - PSEUDO CODE LISTING OF THE PROBABILISTIC CROWDING TECHNIQUE .....	28
FIGURE 7 - PSEUDO CODE OF AN ABSTRACTED GENERALISED CROWDING MODEL (GCM)..	33
FIGURE 8 – SHOWS THE SUBPOPULATION SIZES AT EACH OF THE FIVE PEAKS IN THE F2 TEST PROBLEM FOR A1 (MEAN OVER 100 RUNS) .....	41
FIGURE 9 – SHOWS THE SUBPOPULATION SIZES AT EACH OF THE FIVE PEAKS IN THE F2 TEST PROBLEM FOR A2 (MEAN OVER 100 RUNS) .....	42
FIGURE 10 - SHOWS SUBPOPULATION SIZES ON THE F2 TEST PROBLEM FOR A3 (MEAN OVER 100 RUNS) .....	43
FIGURE 11 – SHOWS THE NUMBER OF EACH REPLACEMENT TYPE OVER 500 GENERATIONS FOR SIMPLE CROWDING ON THE F1 TEST PROBLEM (MEAN OVER 100 RUNS).....	46
FIGURE 12 – SHOWS THE NUMBER OF TOTAL SUCCESSFUL REPLACEMENTS OF OFFSPRING GENERATED FROM BOTH SAME-NICHE PAIRING AND DIFFERENT-NICHE PAIRINGS ON THE F1 TEST PROBLEM (MEAN OVER 100 RUNS) .....	49
FIGURE 13 – SHOWS THE NUMBER OF WASTED REPLACEMENTS OF EACH TYPE FOR THE F2 TEST PROBLEM (MEAN OVER 100 RUNS) .....	51
FIGURE 14 – SHOWS THE NUMBER OF MATCHES (NOT NECESSARILY REPLACEMENTS) OF EACH TYPE FOR THE F1 TEST PROBLEM (MEAN OVER 100 RUNS) .....	52
FIGURE 15 – SHOWS THE REPLACEMENT TYPES FOR THE SAME-NICHE BIASED PAIRING SELECTION OPERATOR ON THE F1 TEST PROBLEM (MEAN OVER 100 RUNS) .....	58
FIGURE 16 – SHOWS THE REPLACEMENT TYPES FOR THE DIFFERENT-NICHE BIASED PAIRING SELECTION OPERATOR ON THE F2 TEST PROBLEM (MEAN OVER 100 RUNS) .....	59
FIGURE 17 – SHOWS THE SUBPOPULATION SIZE FOR EACH OF THE FIX PEAKS IN THE F1 TEST PROBLEM WHERE A SMALL SAMPLE SIZE IS USED IN THE LOCALISED SELECTION STRATEGY (MEAN OVER 100 RUNS) .....	61
FIGURE 18 – SHOWS THE EXACT REPLACEMENT ERROR FOR THE F2 TEST PROBLEM WHEN A LARGE SAMPLE SIZE IS USED THE LOCALISED SELECTION STRATEGY (MEAN OVER 100 RUNS) .....	62
FIGURE 19 – SHOWS THE SUBPOPULATION SIZES OF THE FIVE PEAKS IN THE F1 TEST PROBLEM WHEN USING THE DIRECT REPLACEMENT STRATEGY (MEAN OVER 100 RUNS) .....	64
FIGURE 20 - SHOWS THE MEAN SUMMED SQUARED DEVIATION FROM MEAN SUBPOPULATION SIZE FOR TESTED CROWDING ALGORITHMS ON THE SELECTED TEST PROBLEMS .....	69
FIGURE 21 – SHOWS THE MEAN MAXIMUM PEAK RATIO FOR THE TEST ALGORITHMS OVER THE SELECTED PROBLEM DOMAINS.....	70
FIGURE 22 – PROVIDES A SUMMARY OF THE NUMBER OF GENERATIONS BEFORE REPLACEMENTS DROPPED BELOW ONE FOR THE CROWDING ALGORITHMS ON THE SELECTED TEST PROBLEM DOMAINS. ....	72

FIGURE 23 - DEMONSTRATES THE CURRENT AND AN ALTERNATIVE REPLACEMENT MECHANISM FOR CROWDING BASED ALGORITHMS.....	78
FIGURE 24 - SHOWS THE CURRENT AND INVERTED CROWDING ALGORITHMS .....	81
FIGURE 25 – PSEUDO CODE THE CROWDING MAP CONCEPTUAL MODEL.....	82
FIGURE 26 – SHOWS A SUMMARY OF THE GENERALISED CROWDING MODEL IN GENETIC SEARCH .....	93

## List of Tables

TABLE 1 – SUMMARY OF SELECTED CONFIGURATION FOR COMMON ALGORITHM PARAMETERS .....	14
TABLE 2 – EQUATIONS FOR SELECTED TEST PROBLEMS .....	15
TABLE 3 – SUMMARY OF SELECTED TEST PROBLEMS INCLUDING NAME, DIMENSIONALITY, NUMBER OF OPTIMUM AND INPUT RANGE .....	15
TABLE 4 – SHOWS A BREAKDOWN OF THE AVERAGE REPLACEMENT WASTE AND AVERAGE REPLACEMENTS FOR EACH TYPE OF REPLACEMENT FROM THE F1 TEST PROBLEM .....	51
TABLE 5 – SUMMARY OF THE DIRECTORY AND FILENAMES WHERE THE INVESTIGATION INTO QUINTESSENTIAL CROWDING TEST RESULTS ARE LOCATED ON THE CD-ROM .....	91
TABLE 6 – SUMMARY OF THE DIRECTORY AND FILENAMES WHERE THE AUGMENTED SIMPLE CROWDING TEST RESULTS ARE LOCATED ON THE CD-ROM.....	91
TABLE 7 – SUMMARY OF THE DIRECTORY AND FILENAMES WHERE THE SIMPLE CROWDING COMPARED TEST RESULTS ARE LOCATED ON THE CD-ROM.....	92

## List of Equations

EQUATION 1 – DERATION OF FITNESS BASED ON A MEASURE OF LOCAL SAMPLE DENSITY .	10
EQUATION 2 – SHARING FUNCTION THAT ACCEPTS A DISTANCE BETWEEN TWO SAMPLES AND RETURNS A SHARED VALUE.....	10
EQUATION 3 – TOTAL FITNESS EVALUATIONS CALCULATED FOR EACH ALGORITHM TESTED .....	16
EQUATION 4 – MAXIMUM PEAK RATIO (MPR).....	16
EQUATION 5 – FITNESS PROPORTIONATE SELECTION, PROBABILITY FOR THE SELECTION OF $X$ .....	28

## Introduction

Genetic algorithms are a class of search algorithm from the field of evolutionary computation, inspired by the metaphor of biological evolution. They are a global search algorithm intended to provide approximate or “good enough” solutions to “hard” problems where little or nothing is known about the nature of the search space. A specialised type of evolutionary algorithm inspired search algorithm called niching genetic algorithms employ the exploratory and exploitative benefits of abstracted natural selection. Unlike standard genetic algorithms they are capable of locating multiple solutions by forming groups or subpopulations areas of interest in the search space. Crowding refers to a class of niching genetic algorithms that are based on the biological metaphor of localised competition for limited resources. Algorithms that exploit the concept of crowding in genetic search share commonality in design and function. These algorithms are typically referred to as using restrictive replacement where tight controls are employed as to how offspring samples generated by recombination, are inserted into their parental sample population.

This work provides an investigation into the field of crowding based parallel niching genetic algorithms that form and maintain groups of samples within a fixed sized population at multiple “good” or fit areas in the search space. The common implementations of the crowding approach are reviewed with a focus on the manner in which the concept of crowding is employed and the specific functionalities of each algorithm. The underlying crowding principle is exposed as localised competition for limited resources and is implemented in genetic search as localised generational competition. An abstracted conceptual model of crowding based algorithms is proposed called the generalised crowding model (GCM) that provides a template for the known crowding approaches in genetic search and a framework for implementing the crowding principle.

The underlying goals of crowding are identified as the maintenance of general diversity and the promotion of useful diversity, which become the goals of the generalised crowding model and all crowding based approaches that are specialisations of the model. A simple crowding algorithm is predicted by the model and is shown to represent an embodiment of localised generational competition in genetic search. A simple crowding algorithm is proposed and investigated revealing insight into the nature and behaviours of crowding in genetic search. This is shown to provide useful information as to how to best use and tune existing crowding based algorithms in practical application.

A number of useful crowding centric analysis tools are proposed and used to evaluate simple crowding in terms of the three main elements of the generalised crowding model, namely; upfront selection, localised selection and the replacement strategy. Further investigation into the effect of common approximations and variations to the crowding model used in reviewed crowding algorithms validates the effectiveness of the generalised crowding model as a framework for crowding based algorithm development.

Preliminary test results of augmentations to simple crowding provide empirical support in verifying some assumptions regarding replacements in crowding algorithms. Further, preliminary test results provide quantified evidence that dispels previously held assumptions regarding biasing in upfront selection and specialised crossover operators in crowding based niching. The detailed investigatory effort realised in this work provide unique and quantified insight into the application of the crowding principle to genetic search, the results of which are expected to both aid in the design and development of future crowding based niching genetic algorithms, as well as aid in the application and tuning of existing crowding based niching genetic algorithms.

### ***Canonical Genetic Algorithm***

Darwin [1] first proposed the theory of biological evolution indicating that natural selection is always searching for slight variations that give a natural organism an advantage in their environment, and thus more of a chance at survival and procreation. Minor variations that allow organisms to better adapt to their surroundings are likely to be preserved and accumulated in a population over long periods of time. It is this accumulation and preservation of favourable variations (or constant refinement and adaptation) and rejection of unfavourable variations that facilitates the adaptive search technique. It is this important search metaphor that is used as the founding principle for the field evolutionary computation.

Holland and colleagues clearly identified the robustness and power of adaptive natural systems and during the 1960's and 1970's investigated using metaphors from natural systems to drive adaptive artificial systems. Holland [2] devised an optimisation technique that was a shift over more conventional approaches. His idea was to let the algorithm determine "what need to be done" rather than the idea of explicitly specifying "how to do it". The proposed system used experience to adapt behaviour to an environment, where previous sampling of the environment was exploited to improve the system.

Holland's original system involved three elements, specifically:

1. The environment or the selective pressure forcing individuals to compete for limited resources
2. The reproductive plan or the progressive modification of structure through means of suitable genetic operators
3. And finally the fitness measure or means to evaluate individual structures

Experience is the mechanism that allows Holland's reproductive plans to positively vary structures and continue to improve system performance. By using a population of structures or candidate solutions, the algorithm was essentially sampling the problem space. The method used to retain the experience gained by sampling was by having that encoded knowledge represented in the current population. This implies that the population provides a summary of interesting sampled areas discovered thus far that may payoff. A search exploration-exploitation trade-off was identified between the amount of information retained in the population, and the amount of exploration to perform.



Retaining too much information by keeping all good samples prevents progressive modifications and improvements. Too little information regarding previous sampling causes information to be lost, potentially for ever and the inability to exploit knowledge already discovered.

The canonical genetic algorithm is historically the standard genetic search technique in a class of computational search techniques called evolutionary computation that are based on the metaphor of biological evolution. Genetic algorithms attempt to find an adequate or 'good enough' solution for the selected problem domain given the time and resources allocated. They are an algorithm best suited to complex domains where getting the best solution is typically less important than getting an approximate solution.

Parameters of the problem domain that define a candidate solution are encoded. This encoding allows a defined range of possible values for each search parameters to be expressed. Encoded candidate solutions are directly manipulated by the algorithm allowing a search through each encoded parameters value space. A population of encoded candidate solutions or samples is maintained by the algorithm. This gives the algorithm the power of being able to sample many different areas of the encoded problem space at the same time. Using a rich set of sample points in the problem space in parallel reduces the probability of finding and getting stuck on poor samples in the search space.

The genetic algorithm is directed in its search for a single global solution. This direction is provided by the algorithms objective or fitness function. It is the means by which candidate solutions are evaluated and assigned some level of goodness or cost. Being blind in terms of the search for a global optimum, the algorithm relies only upon feedback from the objective function to determine the direction in to which proceed. It does not concern itself with the details or mechanisms by which the evaluation figure was derived. In this way, very little information is required regarding the means or mechanisms used to evaluate a potential solution to the search problem.

## **The Algorithm**

There are only two elements of the algorithm's implementation that are problem specific, they are the details of what is encoded as a candidate solution and the manner in which solutions are evaluated. For this reason genetic algorithms are considered a weak form of artificial intelligence because the technique makes relatively few assumptions about the search domain it is applied to. Though the algorithm provides no guarantee of locating the best possible solution, it is free to effectively explore in unexpected or unintuitive directions in the search space and potentially arrive on surprising solutions.

The following as a pseudo code listing of the canonical genetic algorithm.

N: The fixed number of individuals in the population

1. Initialise population with random bitstrings
2. Evaluate fitness of population
3. Loop until stop condition

- a. Fitness proportionate select reproductive set of size  $N$
- b. Recombine to produce  $N$  offspring
- c. Perform mutation of offspring (optional)
- d. Evaluate fitness of offspring
- e. Replace current population with offspring

Figure 1 - Pseudo code listing of the canonical genetic algorithm

A means for encoding solutions proposed by Holland involves reducing the solution parameters to their binary values and concatenating them together to form a fixed length bitstring. Values are mapped onto value ranges between zero and a power of two, for example  $2^{10}$  gives 1024 distinct values between 0000000000 and 1111111111. The encoded version of a candidate solution is commonly referred to as the genotype or a chromosome and is directly manipulated by the algorithm. The decoded bit string is called the phenotype and is meaningful in terms of the problem domain. Solutions are decoded so that they can be evaluated by the objective function and allocated a fitness costing. In biological terms each bit value is called an allele or gene value, and each location or offset in the bit string is referred to as a locus.

The problem specific objective function allows the genetic algorithm to meaningfully compare individual candidate solutions directly using their evaluation score or fitness. Once each candidate solution has a fitness score it is possible for the algorithm to exert selective pressure on the population giving individual solutions opportunity to survive and procreate. The intent is to select on average those solutions that are more fit in the population. The more accurate and meaningful the objective function's evaluation of a potential solution, the more useful the information is to the algorithm allowing it to be more effective at searching for a global solution. This abstraction and simplification of natural selection affords a higher probability for fit individuals to survive and reproduce, though does not prevent the possibility of any solution being selected. Selection is not picking the top  $n$  fittest solutions in the population, rather it is a probabilistic process that gives every individual the chance of being selected, and a greater or proportional increased chance for those individuals that are fitter than average. This has the effect of maintaining some level of diversity in the population, somewhat preventing the algorithm from stalling or arriving at a poorer quality solution sooner.

The goal of the canonical genetic algorithm is to explore the encoded problem search space sufficiently given the computational space (memory) and time allocated. The end point of the algorithm involves converging the population of samples in the problem space towards a particular area of the encoded problem space. Techniques that maintain solution diversity in the population promote the exploration process and ensure that the population does not prematurely converge. This phenomenon occurs when the algorithm begins overusing sample experience in a locally 'good' area of the search space, before effectively exploring other better (more fit) areas.

Always selecting the fittest solutions to parent the next generation will almost certainly cause rapid convergence and the 'take-over' of genetic material from fit solutions which

will stall the search process. It should be noted that for some problems it is effective to select the best individual each generation and propagating it directly into the next generation. This is called elitism [3] and provides a simple form of generation overlap that is able to speedup the searching process. Another simple amendment to the technique involves sterilising or removing those solutions with the lowest fitness each generation in an effort to provide some focus on the better solutions in the population.

The probabilistic selection of parent individuals is implemented using a stochastic roulette wheel. After the population has been evaluated using the objective function, each solution is allocated a slice on a roulette wheel. The more fit an individual, the larger the allocated slice on the wheel. The wheel is then spun  $N$  times (where  $N$  is the size of the population) to select parents that may reproduce (recombine) and generate offspring that become the next generation.

The selection method defines the solutions that survive out of the current generation and are able to reproduce, thus satisfying stochastic simulation of survival of the fittest or survival to reproduce. Reproduction of encoded candidate solutions is achieved using a crossover of two parents' genetic material (bitstrings). The intention is that children are produced by copying segments of genetic material from both parents. A random location (or locus) on the bit string is selected and used to split both parents. The four bit string segments are then recombined to create two new children that are included in the following generation. This form of crossover is called one-point crossover because one point is selected to split the genetic material of both parents.

Once enough offspring solutions are generated to populate the next generation, the final genetic operator called mutation is applied. Mutation is based on the biological idea of copying errors in genetic material during the reproductive process of creating offspring where parental genetic material is duplicated for the offspring and recombined. This is implemented by flipping a random bit, commonly with a probability in the range of a single bit in one thousand copied bits (0.001). Mutation is needed because as discussed not all sampled information can be retained. Information is lost through the selection and recombination processes as the population converges to an interesting area of the search space. Mutation provides a mechanism to introduce minor changes infrequently that introduce new information back into the gene pool, providing an additional dimension to exploration of the search space. Selective reproduction and recombination provide the real power of the algorithm, and it is believed by Goldberg [3] that mutation is a secondary genetic operator and should have a weak effect.

The size of the problem space for the genetic algorithm to search is determined by the number of bits in an encoded chromosome. The relationships between distinct samples from the encoded problem space and the allocated fitness scoring can be visualised as a topographical landscape, where the peaks of the landscape represent the good solutions. It is common for problems that are less than about 30 bits in length to be exhaustively enumerated [4]. Problems with more components spaces are simply too large to be exhaustively search and may be suited to the genetic algorithm.

## Theory

When a population is reasonably diverse, the genetic algorithm can explore many peaks in the problem space at the same time. Information is learned through testing samples and is stored intrinsically in the population by the algorithm as it searches. The information stored is the similarities in genetic material between fit individuals. A theory to describe how genetic algorithms use and exploit genotype similarity proposed by Holland [2] is called schema theorem.

Schema theorem is centred on small templates or abstractions of genetic material called schemata that match onto chromosomes in the population providing an indication of similarity. It is a fundamental mathematical theory of how and why genetic algorithms are successful search technique. Specifically the theorem provides a limit on the change in sampling rate of a single schema from the current generation to the next. Schemata consist of a 3 character alphabet;  $\{1, 0, *\}$  where '\*' is a wildcard character that can match onto 0 or 1. Similarity between schemata is measured by the number of bit positions that that are different between two bitstrings of equal length (Hamming distance). This idea provides a way of thinking about chromosome similarity allowing the genetic algorithm to explicitly use the representation-to-fitness relation information and implicitly use the 'chromosome similarly'-to-fitness information as well.

The fitness of schema can be calculated by the average fitness of all chromosomes that exhibit the schema. The fitness proportionate selection scheme in genetic algorithms ensures that a larger percentage of fit schemata are propagated to the next generation on average. The longer a schemata structure in the bitstring, the more likely it is to be disrupted or broken up by the crossover operation. This means that the GA is only able to implicitly work with short, fit schemata which Goldberg [3] refers to as building blocks. These small schemata become in effect partial solutions to the problem which are combined to provide larger building blocks; an idea put forward by Goldberg called the building block hypothesis.

Using the schema theorem allows the population to be considered as a collection of instances of schemata. The theorem provides a way of analysing the effects of genetic operators allowing insight into what is actually happening under the covers of the algorithm. Whitley [4] proposes that it is useful to consider a schema as a hyperplane within the hyperspace of the encoded problem area. By operating on schemata it can be thought that the algorithm firstly selects building blocks that represent fit hyperplane's, and then focus in on an area in the selected hyperplane. A candidate solution is nothing more than the algorithm testing various combinations of fit hyperplane's. As the algorithm progresses, the hyperplane's to choose from becomes reduced by selection. If two parents are instances of the same schema, then the child too will have the schema. In the alternate case, recombination of two fit building blocks from different parents is expected to produce a child that is on average more fit.

The ability of the genetic algorithm to sample many small schemata at the same time is something that Holland calls intrinsic or implicit parallelism. This idea implies that many complementing and competing hyperplane's are evolved in parallel and propagated via

recombination. This shows the importance of the crossover recombination operator in both not disrupting building blocks, while at the same time combining building blocks in a way that is meaningful in terms of the search.

### **No Free Lunch**

Wolpert and Macready [5] proposed a framework for comparing search algorithms, whose basis was using average performance over all possible cost functions. The framework provided an effective way of evaluating search algorithms without the details of the problem domain the search was applied to. It was shown that all search algorithms extrapolate information from some subset of points in the problem space and apply costing, and that search performance thus far cannot be used to predict future search performance.

A theory regarding search algorithms was proposed that indicates that there is no silver bullet in terms of search algorithms or ‘no free lunches’. The theorem proposed that there can be no search algorithm that can solve all types of problems better than any other search algorithm. When the performance of a particular algorithm applied to all types of search problems and compared to other search algorithms, it will be shown to perform the same, on average. This means that if search algorithm A outperforms search algorithm B on a set of test problems, then there will exist a set of problems in which B will outperform A on average by the same amount. The theorem predicts that there is a balance between problem domains and search algorithms. Though genetic algorithms can be applied to just about any problem domain where candidate solutions can be encoded and assigned meaningful cost, they cannot be the best solution for all domains.

It is not a total loss, genetic algorithms do perform well for many problem domains, and are generic enough to be applied without any restructuring of the algorithm for the problem domain. “Genetic algorithms are theoretically and empirically proven to provide robust search in complex spaces” Goldberg pp.2 [3]. A good way to judge whether the optimisation technique will perform well is to investigate whether it is able to effectively solve similar problems. To make the best use of the algorithm, application should seek to maximise useful and meaningful domain specific information in both encoding and evaluating candidate solutions. The genetic algorithm is good at sorting interesting areas of the problem and exploiting that information. In fact, in a lot of cases the algorithm is too good, and needs to be slowed down by introducing more diversity. Wolpert and Macready describe the algorithm as having a reordering of priorities compared to conventional optimisation techniques where the goal is continuous improvement rather than focusing on destination of the search. Genetic algorithms provide a way of getting some form of a “good” solution quickly, with no guarantee that it is globally the best solution.

### ***Niching Genetic Algorithm***

A niche is a concept from the field of ecology and population biology that represents the amount of activity of a species in an n-dimensional hyperspace of every variable in the natural environment [6]. The concept is typically used to analyse a species activity along one or two environmental variable dimensions revealing information about a species that

can be compared to other species. Further it can be said that a species occupies an economic role in an ecosystem. Though a population of a species can occupy a niche in an ecosystem, a species is typically distributed over a number of ecosystems, where each population of the species occupies similar or the same economic role in each ecosystem. A species occupies a niche when the extent of the species is limited to a single ecosystem. Organisms living within a niche exist within reproductive communities and are typically distributed discontinuously across a geographical area. A fuller biological review of niching will be found in [7].

The metaphor of natural species living in or occupying economic roles in an environment can be abstracted and used in genetic search to locate and maintain samples at areas of interest in the search space. In the abstraction, the search space is an environment or ecosystem that contains a variable number of economic roles, the location of which is the goal of the search algorithm. Typically a finite set of samples (population size) is distributed uniformly random across the search space, and it is the role of the niching search algorithm to discover the economic roles (fit areas of interest) in the domain, and maintain samples at those areas. In this way it is common to refer to samples at areas of interest as a niche or subpopulation of solutions sitting on an adaptive peak in the search space.

Niching genetic algorithms seek to both explore sufficiently to locate areas of interest in the domain, and exploit the discovered knowledge to maintain solutions at areas of interest in the domain. Niching algorithms are designed for search and thus solution maintenance is required to both provide limited exploration of the area of interest as well as return the solutions to the user at the end of the run. The final population is commonly taken as the result of the search, where the best (fittest) solutions in each subpopulation are taken as the solutions discovered by the search. Given the nature of the base metaphor and the goal of the algorithm to locate multiple solutions, niching genetic algorithms are commonly applied to those search problems that require multiple approximate or “good-enough” solutions, such as engineering and optimisation problems that require a good but not necessarily the best solutions.

Goldberg [3] defines the requirements of a niching algorithm as follows:

1. Subpopulations should be maintained stably. Once an area of interest is discovered, a subpopulation should be maintained there with some stability of population size.
2. Subpopulations should not be competitive. Ensures that niches do not compete for dominance of the entire fixed size population.
3. Subpopulation size should decrease with the fitness of the area of interest. Niche exploration should be relative to the potential quality of the solution returned from the niche.

Point 3 is a requirement of some, but not all niching genetic algorithms as will be discussed in the Parallel Niching Genetic Algorithms section. The founding principle exploited by niching genetic algorithms is broad and can be employed in genetic search

in a number of differing ways. Mahfoud [8] described a simple niching genetic algorithm classification system as follows:

1. Spatial. Search techniques that are capable of forming multiple niches at the same time.
  - a. Population space. Niches are formed in parallel within the same population.
  - b. Geographical space. Niches are formed across a number of island populations.
2. Temporal. Search techniques that form niches sequentially.

The majority of niching genetic algorithm research has been into spatial techniques that form niches in parallel within a single population of solutions called parallel niching genetic algorithms. Little research has been performed into geographical or temporal niching algorithms that meet the definition of niching algorithms provided. Given the bias in previous research, the focus of this work was on the parallel niching class of search algorithms.

## Theory

Given two areas of the search space with unknown local best fitness and one of them biased to better fitness than the other, how is the algorithm able to redirect sampling (subpopulation size) towards the better of the two areas? The question of sharing payoff between exploration and exploitation was related by Holland [2] to a version of the two-armed bandit problem for statistical decision theory. This useful concept was related to the niching theory in genetic algorithms by Goldberg [3].

Goldberg described the two-armed bandit problem as a two-armed slot machine with that returns a payoff for the cost of pulling of each handle, where one of the handles had a bias towards higher payoff. The biased arm is not known a priori, though if it was, it would be the only arm pulled each turn. The solution to the problem is to use sampling of each arm on the slot machine to discover the bias. Given a canonical genetic algorithm, the population is expected to eventually converge to the biased arm in the long run. The goal of the problem is to maximise payoff and limit cost, where each sample performed reveals more information regarding the payoff bias in the problem. The goal of a search algorithm is to effectively use the information revealed to limit cost for each sample taken, and discover the solution as quickly as possible. The problem can be generalised from two-arms to k-arms and was thus referred to by Goldberg as the k-armed bandit problem.

The problem accurately describes the difficulty faced by both the canonical genetic algorithm and niching genetic algorithms. Specifically the necessary trade-off between exploration of the search space and exploitation of knowledge already gained from exploring the search space. Further it accentuates the point that for niching search algorithm to be effective they must maintain some information regarding discovered areas of interest in the search space relative to the payoff of other discovered areas of interest in the search space.

## Parallel Niching Genetic Algorithms

Parallel niching genetic algorithms are the class of niching genetic search techniques that have historically been researched more than other niching techniques. As previously defined, parallel niching requires that niches are formed and maintained in parallel within a single population. Unlike geographically distributed and temporal niching techniques, parallel niching is generational and must achieve niche maintenance across iterations of the algorithm where newly generated samples replace existing samples. There are two main classes of parallel niching genetic algorithm that achieve the goals of a niching search in differing ways, they are sharing and crowding methods.

Sharing or fitness sharing niching genetic algorithms were proposed by Goldberg and Richardson [9], and are based on an extension of the above two-armed bandit problem where payoff is shared between the members of a niche. This is achieved by derating the payoff (fitness) of samples based on the density of samples within a region of the search space (similarity) as follows:

$$F'(i) = \frac{F(i)}{\sum_{j=1}^N sh(d(i, j))}$$

Equation 1 – Deration of fitness based on a measure of local sample density

where  $F'(i)$  is the derated fitness of sample  $i$ ,  $F(i)$  is the raw fitness,  $d(i, j)$  is the distance between population members  $i$  and  $j$ , and  $N$  is the population size. Density is approximated by evaluating the similarity between a sample and all other samples in the population using a sharing function. A user defined radius limits the area in which similarity is calculated and thus the sum of the shared values is representative of the sample density within the user defined radius of each derated sample. The sharing function is implemented as follows:

$$sh(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^\alpha & d < \sigma_{share} \\ 0 & d \geq \sigma_{share} \end{cases}$$

Equation 2 – Sharing function that accepts a distance between two samples and returns a shared value

Where  $sh(d)$  is the sharing function,  $\sigma_{share}$  is the user defined niche radius and  $\alpha$  controls the shape of the sharing function. Fitness sharing is the most practically used and researched of the parallel niching genetic algorithms. This is because the sharing technique is capable of redistributing population members across the search space relative to the payoff of the areas of interest discovered in the search space.

The second main class of parallel niching genetic algorithms is referred to as crowding and is based on the concept of competition for survival between similar members of the population that provides a niching effect. This area of parallel niching genetic algorithms



has had some focus, though significantly less than that of the fitness sharing technique. Crowding techniques are commonly computationally less expensive and simpler to implement and apply than fitness sharing techniques, and are capable of locating and forming stable niches at areas of interest in the search domain. Given the potential power of the class of algorithm, and the relatively less understanding of crowding based implementations of the niching metaphor, the topic of crowding based niching genetic algorithms is the focus of this work.

## Methods

This section provides a review of the component based research methodology adopted in this work. Further a summary and explanation of common algorithm configuring choices is provided, as is a listing of the selected test problem domains used to test the crowding based algorithms. Finally a summary is provided of the properties that are important when evaluating niching genetic algorithms with a focus on the relevant measured used to gauge algorithm performance in this work.

### *Methodology*

A reductionist methodology was adopted for the research in terms of both investigatory and experimental work where algorithms are decomposed into discrete logic units. The philosophy was selected for both its empirical nature and its compositional or subsumption approach to algorithm analysis. The approach partitions components into logical groupings to aid in the identification of a succinct definition of the broader goal of an algorithm. Further it aids in the isolation and facilitates analysis of those components that are required or critical to achieving the identified broader goal.

The methodology selected required that the core principle or desired end point were identified for the algorithms under study. This was achieved by reviewing the algorithm in question and removing those components that were not required though are considered useful for practical application such as stochastic generational gap (crossover rates  $< 1$ ) and mutation operators. Conservative configurations were selected to both provide consistency in algorithm result and attempt to reveal the true nature and behaviour of the algorithm been studied. This was achieved by using conservative configurations based on algorithm usage heuristics and algorithm operators that introduced low amounts of disruption and noise to the underlying search principle.

### Common Algorithm Configurations

A binary encoding was used to represent solutions or members of the population. Binary encoding is both a standard form of encoding in evolutionary computation and has been the encoding of choice in previous research into niching genetic algorithms providing a measure of comparison. Further, test results by Watson [10] shows that performance varies little between binary encoding and gray encoding. A bitstring size of 15 bits was used for each encoded parameter value. For the selected function optimisation problems, bitstrings were decoded by first calculating associated integer value, and dividing that value by the maximum decoded value ( $2^{15}-1$ ). This gave a floating point value  $x \in [0,1]$ , where the length of the bitstring for each parameter was 15 bits providing a resolution of  $2^{15}-1$  unique values. The floating point value was then rescaled to the problem specific parameters range of values.

A distance measure of some kind is required in most niching genetic algorithms to provide a scoring of similarity between two population members. Two common measures are the Hamming distance and Euclidean distance that provide distance between encoded bitstrings and decoded vectors of equal length respectively. Hamming distance was

commonly used in early algorithm development, though in function optimisation domains it is now common to use the Euclidean distance between the decoded function parameters as it provides a more meaningful result given the domain is function optimisation.

Mutation is not used at all during algorithm testing. Mutation is a standard means of introducing random genetic material into the search and can provide local hill climbing behaviour. The operator is useful in practical application, though it was not used during empirical testing given its potential to add noise, disrupt or discuses the underlying behaviour of the holistic algorithms and isolated algorithm components understudy. For similar reasons, the conservative one-point crossover operator was selected. It provides a means of recombining solutions with less disruption than some other recombination techniques. A crossover rate of one was used to provide the opportunity for the majority of the population to participate in recombination, and competition for survival each generation. Again, the reason was to remove the stochastic generational gap that lower crossover rates can provide a conservative baseline in all tested algorithms.

An extended duration of 500 generations was used in all testing to reveal any potential algorithm stability concerns. The number of generations is considered an extended duration given the relative simplicity of the selected test problem domains. All algorithm testing was completed 100 times to provide a statistically improved indication of performance on the selected test problems, and results were taken as the mean over the 100 test repeats. A population size of 100 was used for algorithm testing across all test problems. The size was selected to avoid potential genetic drift concerns known to occur with smaller populations, though was sufficiently large to allow  $\geq 20$  members for each desired optimum in the selected test problems, a value considered to be more than adequate.

### Configuration Summary

The following table provides a summary of the selected configuration and a summarised reason for each of the algorithm parameters common to the majority of niching genetic algorithms tested.

<i><b>Configuration Parameter</b></i>	<i><b>Selected Configuration</b></i>	<i><b>Summarised Reason</b></i>
Parameter Encoding	Binary encoding, 15 bits per parameter	Standard in evolutionary computation, little effect on crowding algorithms
Distance Measure	Euclidean distance between decoded parameter vectors	Consistency, more meaningful measure
Test Pseudo-Random Seed	Test number, same set for each batch of tests	Consistency of pseudo-random number sets between algorithms
Mutation	No Mutation	Local hill climbing effects, may hide true algorithm capability
Crossover	One-point crossover, full crossover (crossover rate of	Simple and low disruption recombination, should be able to

	one)	niche under full crossover
Duration Stop Condition	500 generations	Extended duration to reveal potential algorithm stability concerns
Number of Test Repeats	100 tests	Better than single run evaluation of performance statistically
Population Size	100 members	Suitable size to allow $\geq 20$ members per desired optimum

Table 1 – Summary of selected configuration for common algorithm parameters

### ***Test Problems***

Niching genetic algorithms are suitable for those search domains that are also suitable for standard genetic algorithms, though are algorithms that are suited to those domains that have multiple desired solutions. A type of search domain better suitable for preliminary experimentation is function optimisation. This domain is suited to preliminary testing niching genetic algorithms given the ease at which simple (fast to solve) test problems can be prepared. Low dimensionality test problems provide a type of function optimisation problem that are both easy to visualise in their respective one or two dimensions in terms of parameter to fitness mapping, population coverage of the search space and total search coverage of the search space. The ease in identifying desired solutions in low dimensional search spaces provides a means of analysing the niching algorithms under study in terms of number and quality of desired optimum discovered. An important point for niching algorithm analysis is that it was made increasingly more difficult as the dimensionality is raised.

Given the appeal of artificial function optimisation problems, the use of multimodal test problems have been a large focus in the analysis and testing of niching genetic algorithms. Further, the behaviour and performance observed on simple artificial function optimisation problems is expected to scale (within reasonable limits) to more difficult function optimisation problems, and be somewhat representative of behaviour and performance in different types of search domains.

The term niche refers to the position of a group or species of biological organisms on a peak in the environmental adaptive landscape. Given the use of the niche biological metaphor test function optimisation problems are typically maximisation type problems as opposed to minimisation, and the desired solutions or function extrema in test functions are referred to as peaks or optima.

### **Selected Function Optimisation Problems**

Test function optimisation problems were selected from niching genetic algorithm literature to both provide a basis for comparison to previous work, and test unique properties of the algorithms under study. The selected test problems were transformed into maximisation problems as required. This was not a requirement of the niching algorithms under study rather was done for consistency of function types and convenience of implementation. All functions were tested in either one or two dimensional search spaces

for both easy of analysis and visualisation and thus all test results should be considered as preliminary only.

The following table provides a summary the five selected test function optimisation problems used.

<b>Function Number</b>	<b>Test Problems</b>
F1	$F1(x_1) = \sin^6(5\pi x_1)$
F2	$F2(x_1) = e^{-2(\ln 2)(\frac{x_1-0.1}{0.8})^2} \sin^6(5\pi x_1)$
F3	$F3(x_1, x_2) = 200 - ((4 - 2.1 \cdot x_1^2 + \frac{x_1^4}{3}) \cdot x_1^2 + x_1 \cdot x_2 + (-4 + 4 \cdot x_2^2) \cdot x_2^2)$
F4	$F4(x_1, x_2) = 2200 - (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$
F5	$F5(x_1, x_2) = \sum_{i=1}^2 -x_i \cdot \sin(\sqrt{ x_i })$

Table 2 – Equations for selected test problems

The following table provides a further summary of the selected function optimisation test problems used. It is important to note that on the F5 problem not all the maxima were desired, rather a select few of the maxima with higher fitness were chosen. The reason for this was to focus on a restricted portion of the domain with better fitness (higher elevation maxima) and observe the effects of deceptive areas of interest.

<b>Function Number</b>	<b>Name</b>	<b>Problem Dimensionality</b>	<b>Total Desired Optima</b>	<b>Input Range</b>
F1	Sine Function - 5 Peaks of equal height	1	5	$x_1 \in [0,1]$
F2	Sine Function - 5 Peaks of differing height	1	5	$x_1 \in [0,1]$
F3	Scaled six-hump camel back function	2	4	$x_1 \in [-2,2]$ $x_2 \in [-1,1]$
F4	Himmelbau's Function	2	4	$x_1 \in [-6,6]$ $x_2 \in [-6,6]$
F5	Schwefel's Function	2	3	$x_i \in [-500,500]$

Table 3 – Summary of selected test problems including name, dimensionality, number of optimum and input range

Test functions F1 and F2 are historic test functions from niching literature that are used for initial or preliminary algorithm analysis. The functions are both simple one-dimensional search domains that test the basic features of niching algorithms, namely peak location and peak maintenance. The functions further provide a simple test of a niching genetic algorithms ability to locate and maintain optima of same and differing

optima. For these reasons, both the F1 and F2 test problems were used for all preliminary experimentation and analysis. In the case where more general preliminary results are required, testing was performed on all five of the selected test problems.

Please see Appendix A – Test Functions for a graphical representation of the selected function optimisation test problems.

## ***Measures***

Niching genetic algorithms are defined as both capable of locating and maintaining multiple optima in a search space. The definition implies two main concerns when evaluating a niching genetic algorithm, namely the number and quality of optima located, and the stability or maintenance of located optima.

## **Algorithm Iterations**

Given that the number of algorithm iterations or generations executed for each algorithm is fixed at 500, the number of fitness evaluations executed for each algorithm can be calculated as follows:

$$F_{eval} = T(N + NG)$$

Equation 3 – Total fitness evaluations calculated for each algorithm tested

where  $T$  is the number of tests (100),  $N$  is the population size (100) and  $G$  is the number of algorithm iterations or generations (500). Thus the total fitness evaluations executed for each algorithm per selected test problem is simply 50100 per test. Given that the number of offspring generated per iteration is equal to the population size, this implies that the potential exists for each population member to compete for survival each generation. This configuration was constant across all tested algorithms and therefore the number of generations can be used as a measure of effort required to achieve or exhibit a characteristic under investigation such as convergence or stability concern.

## **Location and Quality of Optima**

Each selected test problem has a set number of desired optima. A solution can be considered at an optimum if it has a fitness value close to the fitness of the optimum and is within a minimum distance of the optimum's position in the search space. A summarised measure that provides an indication of the number and quality of optimum found is called the maximum peak ratio (MPR) and was suggested for use as an analysis tool by Miller and Shaw [11]. The maximum peak ratio is a single scoring of the sum of the fitness of the best solution  $F^B$  at each optimum divided by the actual fitness of each desired optimum  $F^A$ , calculated as:

$$MPR = \frac{\sum_{i=1}^P F_i^B}{\sum_{i=1}^P F_i^A}$$

Equation 4 – Maximum Peak Ratio (MPR)

Where  $P$  is the number of desired peaks and  $F$  is the allocated fitness scoring. A peak or optimum is considered located if a solution has a fitness  $\geq 80\%$  of the fitness of the peak and is within a minimum distance of the peak. For testing, Euclidean distance was used to measure the minimum distance between decoded values and a value of  $< 5\%$  of each decoded parameters range was used to define membership to the peak. A value of  $5\%$  of the parameters range was selected because it effectively encompasses the desired optima in each of the selected test problem domains without problem-specific or peak-specific configuration. The ratio is a niching genetic algorithm analysis measure and represents both the number and quality of the located desired peaks.

## **Stability**

Desired optima that are located by niching genetic algorithms are expected to be maintained. As has been mentioned, an extended-run duration is used to reveal any potential stability concerns both before and after the population has reached a state of minimal change or convergence. A full crossover rate is employed to provide potential population wide competition and test the algorithms ability to form niches under that potential full competition. Both of these configurations are designed to test the niching algorithms ability to maintain solutions at desired optima, thus a simple measure of population stability are required.

Given that the maximum peak ratio provides a measure of the quality of the best solution at each desired optimum, a simple measure of the number of population members at each optimum over time is a suitable analysis measure of stability. The distance from each population member to all desired optima is measured and the solution is allocated membership optimum with the minimum distance. The measure imposes no minimum distance so that each population member is allocated membership to the closest peak, and there is not a “no peak” classification.

The measure is not a summarisation of stability; rather it is an approximation of population distribution in time, the change in which provides an indication of population stability around desired optimum. The actual population size is not of concern with crowding based genetic algorithms as it is with other niching genetic algorithms such as fitness sharing. What is of interest is the stability in peak membership both before and after population convergence. Solutions that are allocated membership to a peak can be considered collectively as a subpopulation in close proximity to the peak or a niche, though no magnitude of proximity is provided in the measure.

## Previous Research

This section provides a brief review of crowding based niching genetic algorithms, focusing on the need, function and practical considerations. The common theme throughout the review is the underlying crowding concept and the manner in which that concept has been exploited in genetic search. A crowding principle is identified which can be summarised as localised competition, with the goal of maintaining general diversity. In addition this goal is extended to also require the promotion of useful diversity, as the goal of preserving diversity alone is insufficient to be referred to as niching. A common theme of intra-niche competition is shown, with a bias to discourage inter-niche interactions. This theme is further shown to be related to the implementation of the crowding principle in genetic search via what has is called restrictive replacement.

### *Crowding Factor Model*

The algorithm which inspired the class of crowding based niching techniques was proposed by De Jong [12], called the crowding factor model. De Jong's intent was to devise a technique that would maintain diversity and slow convergence, specifically on a multimodal minimisation problem called Shekels Foxholes.

The founding principle for the crowding factor model was drawn from parallels with nature where as more like individuals dominate a niche, the competition for limited resources increases, resulting in lower life expectancies and birth rates. To further aid in making connections to natural systems, an overlapping generational model was used in the algorithm. A generational gap parameter  $G$  was introduced which controlled the size of the reproductive set, the number of offspring produced and the number of individuals from the population that died each generation.

The basis of the crowding factor model technique treated slots in the population as a limited resource and worked to control and limit the growth of dominate building blocks. Growth was limited by implementing a replacement policy that causes offspring to be matched onto and replace similar bitstrings in the population. The replacement policy was shown to effectively limit the number of offspring produced by dominant building blocks, by increasing the chance of similar bitstrings replacing each other as prevalence of the building block increased.

In the crowding factor model, replacement for each offspring produced is considered individually. For each such individual, a sample of  $CF$  individuals are drawn from the population and searched for the most similar bitstring to the offspring in question. Similarity is measured as the number of point differences between the equal length bitstrings, called the Hamming distance. The most similar individual from the small sample is then directly replaced in the population by the offspring, without regard for fitness.

The following provides a pseudo code listing of the crowding factor model technique.

G: Generational Gap, a ratio of the population that reproduces each



- generation, typically 0.1 (10% of the population)
- N: The fixed number of individuals in the population
- CF: Crowding Factor, the size of sample taken from population and searched for the most similar bitstring, typically small values between 2 and 4
1. Initialise population with random bitstrings
  2. Evaluate fitness of population
  3. Loop until stop condition
    - a. Fitness proportionate select reproductive set of size  $G \times N$
    - b. Recombine to produce  $G \times N$  offspring
    - c. Evaluate fitness of offspring
    - d. Loop for each offspring
      - i. Randomly select sample size of  $CF$  from population
      - ii. Search for most similar in sample (Hamming distance)
      - iii. Replace most similar in population with offspring irrespective of fitness

Figure 2 - Pseudo code listing of the crowding factor model

Low generational gap values (such as 0.1) were used to allow the majority of the population to participate in recombination and thus indirectly in replacement. Crowding factor values of 2, 3, and 4 were used in a population size of 50, meaning the relative sample sizes were quite small at 4%, 6% and 8% of the population respectively. A  $CF$  value of 1 provides no crowding similar to the canonical genetic algorithm, and as the factor increases the chance of a similar building block being replaced is increased, providing more pressure to maintain diversity.

De Jong's results [12] showed the technique was capable of preventing premature convergence, and raising the crowding factor value showed a graceful increase in the maintenance of diversity. Improved results were observed on the Shekels Foxholes multimodal test problem similar to increases in mutation rates, though maintaining good solutions, rather than by disrupting diversity. It was shown that as the generational gap  $G$  was increased, the less effective the replacement policy became, requiring the crowding factor  $CF$  to be increased. The generational gap provides a means of keeping individuals in the population longer, providing more reproductive potential, and thus increases convergence. The crowding replacement policy was shown to increase diversity and slow convergence by replacing similar individuals thus limiting growth of dominant building blocks. This implies that the trade-off between generational gap and similarity based replacement could be considered a balance between exploitation (convergence) and exploration (diversity).

## ***Deterministic Crowding***

The crowding factor model is inspired by an ecological metaphor where individuals within the same species compete with each other for limited resources. Natural organisms that are dissimilar live in different areas of the environment, and thus do not directly compete with each other. The crowding principle indicates that individuals that are similar are of the same species and should replace older members of the same species, keeping subpopulation sizes relatively unchanging. Mahfoud [8,13] showed that this underlying idea used in the crowding factor model was suitable for use as a niching genetic algorithm technique and showed how this transformation could be achieved.

An extensive investigation [8,13] into the crowding factor model revealed that the technique was unable to maintain more than one or two optima in simple multimodal function optimisation domains. Investigation revealed that the convergence observed in the algorithm was caused by genetic drift which allowed dominant building blocks to takeover the population. Mahfoud sought an ultimate cause for the techniques inability to maintain stable niches, and showed that the low *CF* values used during replacement caused stochastic replacement errors leading to increased genetic drift.

Mahfoud proposed that each desired optimum in the problem domain could be considered a distinguishable class and that solutions within the region of an optimum could be considered members of the optimum's class. Adhering to the interpretation of crowding's goal of intra-niche replacement, replacement errors were seen as the replacement of a population member in a class by an offspring in a differing class. Results showed that nearly 33% of replacements in the crowding factor model were replacement errors indicating a primary cause for its inability to maintain subpopulations at desired optima.

To achieve a crowding based niching genetic algorithm, Mahfoud's goal was to maximise the number of optima that could be maintained by the population and minimise the number of replacement errors. A number of simple variations on the crowding factor theme were proposed [13] to investigate how crowding based niching could be realised. The generational gap was removed from the model because Mahfoud speculated that crowding based niching should be achievable under full crossover. This speculation was reinforced by De Jong's comments indicating that small generational gap values were required, and that as the gap increased crowding became less effective. Phenotypic comparisons were used to further reduce replacement error and provide more meaningful similarity results between individuals compared to bitstring based (genotypic) similarity.

A variation of the crowding factor model was tested with the *CF* set to *N* (population wide) in an attempt to provide exact replacement. Replacement errors were reduced to approximately 0.1%, though the crowding variation showed little selective pressure. It was speculated that some replacement error remained due to population members sitting on or close to class borders being replaced by members of either class with similar likelihood. Results also showed that the majority (83%) of exact replacements that occurred were of offspring replacing a parent. Mahfoud concluded that crowding (inter-niche replacement) could be approximated by only allowing offspring to replace a parent. Two offspring were produced from each pairing and offspring were matched to one

parent each so that the summed similarity between parent and child in the combined matching was minimised. Replacement errors in this combination were reduced to 0.7%.

Test results from this variation on crowding showed that selective pressure was non-existent, and in the case of the parent-based replacement, selective pressure was observed to be negative. This negative selective pressure was due to the fitness proportionate selection choosing the fittest parents which were then replaced by their offspring which were typically less fit. The final variation on the technique was to remove fitness proportionate selection up-front and implement selective pressure at replacement time. Selection of the reproductive set was changed to random without replacement, and offspring-parent match-ups became a fitness tournament for placement in the population. Mahfoud's final accumulated variation on crowding showed replacement error close to 0.2% and gave good results on simple test problems [8,13]. This accumulated variation became known as deterministic crowding.

The following provides a pseudo code listing of the deterministic crowding technique.

N: The fixed number of individuals in the population  
d(x,y): Distance evaluation of two individuals, either phenotypic (Euclidean)  
or genotypic (Hamming)  
f(x): Fitness evaluation of individual population member

1. Initialise population with random bitstrings
2. Evaluate fitness of population
3. Loop until stop condition
  - a. Randomly select reproductive set of size  $N$  without reselection
  - b. Recombine to produce  $N/2$  pairs of offspring
  - c. Apply mutation (optional)
  - d. Loop for each pair of offspring
    - i. If  $(d(\text{parent1}, \text{child1}) + d(\text{parent2}, \text{child2})) \leq (d(\text{parent2}, \text{child1}) + d(\text{parent1}, \text{child2}))$ 
      - A. If  $f(\text{child1}) > f(\text{parent1})$ , child1 replaces parent1
      - B. If  $f(\text{child2}) > f(\text{parent2})$ , child2 replaces parent2
    - ii. Else
      - A. If  $f(\text{child1}) > f(\text{parent2})$ , child1 replaces parent2
      - B. If  $f(\text{child2}) > f(\text{parent1})$ , child2 replaces parent1

Figure 3 - Pseudo code listing of the deterministic crowding technique

The approach was called deterministic crowding because of the deterministic manner in which replacements occur, as opposed to the stochastic sampling based method used in the crowding factor model. The replacing of parents was reminiscent of another technique

devised to increase population diversity called preselection. Cavicchio [3] devised a number of variations on the canonical genetic algorithm theme, one of which used the idea of preselecting the population members to replace in the population. The preselection scheme required that the offspring with the highest fitness compete with the parent with the lowest fitness for a position in the population. The preselection scheme, like the crowding factor model was proposed to maintain diversity rather than maintain niches, though the use of parental based replacement is central to deterministic crowding. It is not unreasonable to liken the final deterministic technique to approximate crowding via preselection.

Mahfoud [14] proposed simple analytical models to describe multimodal genetic algorithms, and specifically the behaviour of deterministic crowding. A simple two class model was proposed that assumed that a cross between two individuals from the same class produces two offspring in the same class, and a cross of individuals from two different classes produces one offspring in each class. The expected behaviour of crowding in this case was therefore stable with the population distribution always the same as the prior generation, with no restorative pressure and no genetic drift. Stability of this model is guaranteed regardless of class fitness differential as fitness did not influence population distribution. Any improvement offered occurred within each class. The predictions of the simple two-class model were confirmed by Mahfoud in preliminary test results.

A multi-class model was also provided which showed more complex inter-class behaviours. Mahfoud showed that the majority of crosses in a simple four peak problem results in same class offspring, though there was also the capability of parents from differing class producing offspring of an entirely different class. It was demonstrated that classes were capable of losing members to other more dominate classes on simple problems before the population had reached a state of stability.

This case of producing hybrid offspring was seen more readily when the replacement policy was adjusted to prefer offspring over parents in the case of a fitness draw. These results provided further justification for the technique to prefer parents over offspring with the goal of reinforcing population stability. Mahfoud referred to the case in which parents from two classes are crossed and produce a child in a different class as dominance where the class the offspring belonged to dominates one of the parent's classes. Dominance was shown to require three elements with specific behaviour under crossover, specifically; a dominant class, a dominated class and an assistance class (a class that aids a dominated class in contributing to the dominant class). It was proposed that this behaviour allowed crowding to sacrifice members of less dominant classes to more dominant classes, providing a means of redistribution of the population in a way that the two class model would not allow. Mahfoud speculated that this dominance effect could be useful when fitter classes are able to dominate less fit classes of close proximity in the search space. Mahfoud [15], showed the potential usefulness of the class dominance effect in the massively multimodal deceptive problem where the approximate 5 million deceptive optima were dominated by the 32 global optima.

Mahfoud revised and extended the use of the simple analytical models to provide insight into the influence crossover had on the techniques exhibited class migratory behaviour. In the two-class model, the initial population was proposed to be proportional to the width of the peak's base, given a uniformly random population distribution. As provided by the model, this proportionate distribution was then expected, and was shown to be maintained. Test results on simple test functions confirm the dependence on the initial population distribution in the two class case, even when the fitness ratio between the classes was increased to 256:1. This conjecture regarding initial population distribution was extended to the multi-class case. Here the model showed that the population distribution for a given class is proportional to the width of the base of its peak, and the base of the peaks of which the class dominates.

### ***Restricted Tournament Selection***

Tournament selection provides a means of selecting individuals for survival and reproduction, comparing fitness directly rather than using fitness proportionate information in the selection. It is a popular and useful technique when applied to unimodal problems, though is not as appropriate in the context of multimodal problem domains. In tournament selection, randomly drawn individuals compete with each other based on fitness. In the case of multimodal problems individuals may belong to differing classes meaning in a sense the selection approach compares apples to oranges. Harik [16,17] proposed that tournament selection could be adapted to follow the principle of localised competition for use in multimodal problem domains, called restricted tournament selection.

The technique was inspired by the concept of forcing local competitions in an effort to climb to local optima. To this end, tournament selection was adapted to discourage inter-niche competitions compared to a basic tournament selection implementation, and to encourage intra-niche fitness based tournaments for survival. Harik claimed that deterministic crowding was not suitable for practical application given its predisposition to losing optima (niche subpopulation crashes) through the class dominance effect. To address this need the idea of localised competition directed development of the similarity based restricted tournament selection technique.

Restricted tournament selection (RTS) is a selection technique that draws a sample of  $w$  (window size) individuals from the population, searches for the most similar individual in the sample to a produced offspring and performs a binary tournament between the two bitstrings. Selection of the reproductive set is random with replacement, meaning that stochastically the same population member can participate in recombination more than once.

The following provides a pseudo code listing of the restricted tournament selection technique.

- w: Window Size, the size of sample taken from the population and searched for the most similar bitstring to participate in a fitness tournament

$f(x)$ : Fitness evaluation of individual population member

1. Initialise population with random bitstrings
2. Evaluate fitness of population
3. Loop until stop condition
  - a. Randomly select reproductive set with reselection
  - b. Recombine to produce offspring
  - c. Evaluate fitness of offspring
  - d. Loop for each offspring
    - i. Randomly select sample size of  $w$  from population
    - ii. Search for most similar in sample (Euclidean distance)
    - iii. If  $f(\text{offspring}) > f(\text{most similar})$   
Then Offspring replaces most similar

Figure 4 - Pseudo code listing of the restricted tournament selection technique

Restricted tournament selection (RTS) has the flavour of crowding [16] in that selection of the reproductive set that has been made stochastic in nature. Generational replacements are similarity based as in the crowding factor model, though binary tournaments are performed with the match-ups as in preselection and deterministic crowding. Harik prescribed using a stochastic generational gap in algorithm application where a crossover percentage of 40% was applied. Analytical modelling by Harik [17] shows the technique theoretically has favourable population takeover times (building block dominance), as well as a high-dependence on the window size parameter. Harik proposed a heuristic for the window size that stated that the value should be a multiple of the number of desired optima in the problem domain. A multiple of four was used during testing on standard simple multiple modal problems from the literature, showing promising results.

Like crowding techniques, competition within a niche increases in proportion to the subpopulation size of the niche. The use of fitness tournaments in replacement provides immunity of fitness scaling and fitness proportionate concerns, and provides direct rank based competition. The replacement tournaments in RTS are biased to occur within the niche an offspring belongs, though inter-niche tournaments are not completely prevented. The strength or pressure of this bias is controlled by the width of the window size, where larger values provide a high bias towards intra-niche competition. RTS was shown to allow rapid exploitation of fit building blocks during the opening stages of the run, though it was claimed to not exhibit the almost-exponential growth observed by De Jong [12] in the crowding factor model.

The size of the search area is reduced as the population converges in the canonical genetic algorithm. The RTS technique maintains multiple subpopulations at optima across the search domain, providing a wider search area compared to the canonical genetic algorithm, even at convergence. Harik indicated that it was unknown whether or not this is a useful feature in genetic search. He speculated that it may be good to allow the continual testing of fit building blocks from different subpopulations, though a

potential downside was the increased production of lower fitness offspring that lie away from existing subpopulations, known as lethals.

### ***Multi-Niche Crowding***

Fitness proportionate selection (FPS) is a general class of up-front selection techniques that allocate reproductive potential based on a member's relative fitness to the current population. The crowding factor model used a FPS technique called roulette wheel selection which Cedeño et al [18] and Cedeño [19] believe causes the crowding factor model to converge, owing to its inability to maintain more than a few subpopulations at optima across the search space. Driven by the desire to remove fitness proportionate selection from the crowding factor model the multi-niche crowding technique was developed to maintain useful population diversity.

Cedeño and Vemuri [20] indicated that multi-niche crowding was a computational metaphor for survival of a species in ecological niches in the face of competition. The technique was designed to encourage intra-niche pairing and recombination as well as intra-niche competition for limited positions in the fixed sized population. In the crowding factor model, inter-niche recombination and replacement were still permitted though discouraged via the bias for intra-niche interactions. The goal of removing fitness proportionate selection to maintain diversity and providing an intra-niche bias to the search was to develop a niching genetic algorithm suitable for practical application.

Multi-niche crowding contains three main components, specifically crowding selection, a niche-oriented recombination operator called interval crossover, and a population replacement strategy called worst among most similar (WAMS). Crowding selection requires that an individual is randomly selected from the population and paired with the most similar member from a uniformly randomly drawn sample of CS population members with replacement. Unlike the crowding factor model that uses bitwise similarity (Hamming distance), all similarity in multi-niche crowding is measured using the phenotypic or decoded values (Euclidean distance).

Simple analysis by Cedeño [19] showed that one-point crossover provided too much disruption to the search, producing increased numbers of offspring with lower fitness that belong to neither parents niche – called lethal offspring. A recombination scheme called interval crossover was employed that produced a single offspring per pairing closer to the parents local areas in the decoded search space. The technique allowed offspring to be produced that were more likely to be within one of the parent's niches. Parents are decoded from their concatenated bitstring values into a single numeric value. The decoded parents are considered coordinates in the search space and are ordered such that  $parent_1 \leq parent_2$ . An offspring is generated randomly in the decoded artificial parameter space in the interval of  $\in [parent_1 - \varepsilon, parent_2 + \varepsilon]$ . The  $\varepsilon$  value is a small value typically 0.001% of  $parent_2$ 's decoded value that permits offspring to land within a region of both parents.

Once offspring are generated using interval crossover, the technique requires that they be inserted into the population using the worst among most similar (WAMS) replacement

strategy Cedeño et al [18], Cedeño [19]. Offspring are inserted sequentially, first requiring that  $CF$  groups of  $S$  samples are drawn from the population with reselection. Once the groups are drawn, the most similar member from each group to the offspring in question is found to provide a similarity group of size  $CF$ . The similarity group is then ordered by fitness, and the member with the lowest fitness is replaced in the population by the offspring. This replacement process is repeated for each generated offspring.

The following provides a pseudo code listing of the multi-niche crowding technique.

- N: The fixed number of individuals in the population
  - CS: Crowding Selection, the sample size drawn with reselection during upfront mate selection
  - CF: Crowding Factor, the number of crowding factor groups to prepare for an offspring during replacement
  - S: The sample size of each crowding factor group taken from the population with reselection
1. Initialise population with random bitstrings
  2. Evaluate fitness of population
  3. Loop until stop condition
    - a. Loop to prepare reproductive set of size  $2N$ 
      - i. Select random individual and add to reproductive set
      - ii. Prepare random sample of size  $CS$  from population with reselection
      - iii. Find most similar member from set to first drawn individual
      - iv. Add most similar member to reproductive set
    - b. Recombine to produce  $N$  offspring
    - c. Apply mutation (optional)
    - d. Evaluate fitness of population
    - e. Loop for each offspring
      - i. Prepare  $CF$  groups of samples of size  $S$  from population with reselection
      - ii. Find most similar member from each group to offspring
      - iii. Find lowest fitness individual from set of similar members
      - iv. Offspring replaces member with lowest fitness

Figure 5 - Pseudo code listing of the multi-niche crowding technique

Multi-niche crowding is different from other crowding based niching genetic algorithms in the following ways (Cedeño et al [18]):

1. Both selection and replacement using the idea of crowding
2. The technique has an intra-niche bias in both pairing of members in the reproductive set and the recombination operator



### 3. Competition for placement in the population has an intra-niche bias

A small stochastic generational gap is prescribed for this algorithm with crossover probabilities between 0.9 and full crossover (one). Cedeño [19] indicates that individuals with high fitness are expected to survive multiple generations and that lower crossover probability facilitates increases in duplicate genetic material which further increases convergence. Cedeño also indicates that duplicates are not required to maintain useful diversity as the bias towards intra-niche replacement allows useful diversity to be maintained. Further, clones (that is offspring that make it through the recombination and mutation operators without change), are discarded.

Cedeño provides a number of heuristics for practical application of the technique. The population size  $N$  should be  $\geq 20 \times P$ , where  $P$  is the number of desired peaks or optima in the problem domain. The selection group size  $CS$  and the replacement group size  $S$  should be at least  $2 \times P$ . Finally the number of crowding factor groups  $CF$  should be selected between two and four. Larger group selection size  $CS$  provides more pressure or bias towards intra-niche pairings, and intra-niche offspring given the conservative interval crossover scheme. Cedeño et al [18] also indicate that large  $CF$  causes population members with higher fitness to stay in the population longer, allowing fitness to play more of a role in the replacement strategy. Further, larger sample sizes  $S$  provide increased intra-niche competition for selection into the similarity set.

The selection scheme provides a bias to limit the areas of the search area that can be recombined to produce offspring, and the specialised interval crossover operator further limits the scope in the search space where a generated offspring can lie. Preliminary testing by Cedeño et al [18] as well as more detailed practical application (Cedeño [19]) has shown multi-niche crowding useful for maintaining diversity and stable subpopulations at optima throughout the search.

### ***Probabilistic Crowding***

The deterministic crowding technique is simple, fast and requires no additional parameters over the canonical genetic algorithm. Menshoel and Goldberg [21] indicated that besides the techniques benefits, the largest weakness of deterministic crowding is its lack of restorative pressure – that is allowing species of higher fitness to win over species of lower fitness. An extension of deterministic crowding was proposed that reused the idea of local competition, by allowing individuals to compete for survival probabilistically. This was called probabilistic crowding or deterministic crowding with probabilistic replacement.

Replacements in the canonical genetic algorithm occur without any regard to replacee-replacer similarity. Mahfoud's deterministic crowding algorithm was shown to match pairs of parents and offspring as to maximise the similarity between each pairing. Each parent-offspring combination then directly competes for placement in the population via a binary fitness tournament. The intent of Menshoel and Goldberg in adjusting the replacement strategy of deterministic crowding was to remove the rigid fitness tournament and allow members to win tournaments probabilistically according to fitness.

Given that  $x$  and  $y$  represents a pair of members competing for a place in the population, the probability for  $x$  being selected  $p(x)$  is calculated as:

$$p(x) = \frac{F(x)}{F(x) + F(y)}$$

Equation 5 – Fitness proportionate selection, probability for the selection of  $x$

where  $F$  is the fitness function.

The following provides a pseudo code listing of the probabilistic crowding technique.

N: The fixed number of individuals in the population  
 $d(x,y)$ : Distance evaluation of two individuals, either phenotypic (Euclidean)  
or genotypic (Hamming)

1. Initialise population with random bitstrings
2. Evaluate fitness of population
3. Loop until stop condition
  - a. Randomly select reproductive set without reselection
  - b. Recombine to produce offspring
  - c. Apply mutation (optional)
  - d. Loop for each pair of offspring
    - i. If  $(d(\text{parent1}, \text{child1}) + d(\text{parent2}, \text{child2})) \leq (d(\text{parent2}, \text{child1}) + d(\text{parent1}, \text{child2}))$ 
      - A. Pair  $\text{child1}, \text{parent1}$  and  $\text{child2}, \text{parent2}$
      - B. Allocate replacement probabilities
      - C. Probabilistically select winner
    - ii. Else
      - A. Pair  $\text{child1}, \text{parent2}$  and  $\text{child2}, \text{parent1}$
      - B. Allocate replacement probabilities
      - C. Probabilistically select winner

Figure 6 - Pseudo code listing of the probabilistic crowding technique

The principles of probabilistic crowding are to hold tournaments for survival between similar individuals and those tournaments to be conducted probabilistically. Three variants of the technique were proposed, analysed and preliminary tested. They consisted of mutation only, crossover and mutation and crossover only variants. A stochastic generational gap was allowed with mutation probabilities of 0.1 for the mutation only variant, 0.3 for the mutation and crossover variant and a crossover probability of 0.6 for both crossover based variants.

Like deterministic crowding, the probabilistic crowding technique was shown to be simple, fast, and added no additional specialised parameters. Preliminary tests showed

that the probabilistic based replacement policy provided stable and predictable niching results and was capable of providing some restorative pressure.

## ***Discussion***

The crowding factor model is not a niching technique. It was shown by Mahfoud [8,13] that the technique was unable to sufficiently maintain more than two niches on simple test multimodal problems. The strengths of the crowding factor model technique are both its founding principle of competition for limited resources based on similarity, and its practical ability to limit and control growth of dominant building blocks in the population through a simple replacement policy.

The deterministic crowding technique proved that the underlying concept of crowding can be used successfully as the basis of a niching genetic algorithm. The minimisation of replacement errors was the focus during the inception of the technique and class maintenance and dominance were the focus of the analysis of the technique.

Deterministic crowding provided a means of automatically forming subpopulations at peaks in multimodal problem domains using what was called restrictive replacement. The technique provided a means of maintaining the population distribution allocated in a uniformly random initialisation and allowed refinement and exploration within the distribution.

Mahfoud [13] mentioned that deterministic crowding was capable of maintaining the pre-existing mixture of classes, though it does not address the problem of how the population arrives at a mixture of classes. The crowding concept is a diversity maintenance technique that facilitates intra-niche competition and explicitly attempts to avoid inter-niche competition. In this light, the behaviour artefacts of class dominance (the syphoning of members by a dominant optimum) could be considered contrary to the goal of the crowding principle.

The inception of the restricted tournament selection technique was focused on local competition using a rank based binary fitness tournament between a produced offspring and a similar member of the population. A sampling method reminiscent of the crowding factor model was employed to locate a competitor for the tournament. Unlike the crowding factor model, the sample size used in restricted tournament selection was a multiple of the number of desired optima to provide a strong bias towards intra-niche competition in the tournament. This bias discouraged but did not prevent inter-niche competition and the technique was shown to provide stable maintenance of subpopulations as was seen in deterministic crowding.

The underlying principle of crowding based niching in genetic algorithms is local competition for limited resources, with the primary goal of maintaining general diversity. The multi-niche crowding technique was shown to also exploit this principle, focusing strongly on intra-niche refinement. Sampling during a fitness-neutral selection provided bias towards intra-niche exploitation which was further reinforced by the specialised recombination operator. This bias, however strong, is expected to have an influence on

the search algorithms ability to explore and was a concern raised by Mahfoud [8] for crowding based algorithms.

A more complex sampling method than restricted tournament selection is used in multi-niche crowding to find a population member similar to an offspring in question. Interestingly, unlike deterministic crowding and restricted tournament selection, multi-niche crowding does not use a binary tournament between the offspring and a selected population member, rather like the crowding factor model, it performs a direct replacement regardless of fitness scoring. Also interesting is the size of the crowding selection  $CS$  and group size  $S$  that are recommended as to be multiples of the number of desired niches, a theme which was also common to the restricted tournament selection technique.

Probabilistic crowding questioned the need for a rank based binary fitness tournament used in deterministic crowding and proposed a fitness proportionate selection as a replacement strategy. The strength of crowding based niching genetic algorithms has been shown to be their ability to maintain diversity. Mahfoud [8,13] indicated that rather than the goal of simple diversity, crowding methods are able to maintain useful diversity at areas of interest in the search space. This strength of crowding is a weakness if the initial population distribution is light in density or perturbed in some way. Probabilistic crowding also addressed the issue of providing an opportunity to distribute members in a fitness centric way called restorative pressure, which is unlike class dominance in deterministic crowding that redistributed members between classes influenced by the crossover path between optima.

The crowding principle shown to be a common theme between the five algorithms discussed in this section is centred on local competition for placement in the population. This principle provides a niching effect that maintains existing population diversity. The measure of comparison in the local competition typically is fitness scoring which has the effect of causing the quality of the local subpopulation to improve relative to the way in which fitness is calculated. Ultimately this improvement in quality effect of local competition facilitates increased differentiation between groups and increased similarity within groups, known as niching. The algorithms discussed in this section have been shown to directly use the crowding principle of local competition in some way, and thus in varying ways exploit the effects the principle has on the population as a multimodal search technique.

## **Quintessential Crowding**

This section provides insight into the crowding principle and its application to genetic search. The restrictive replacement concept that defines crowding based implementation is shown to be too broad a description. A more succinct definition is provided called localised generational competition. An abstracted conceptual model called the generalised crowding model is proposed that is shown to embody the functionalities of existing crowding based niching genetic algorithms. The model is shown to provide a template for crowding based implementations and an algorithm framework for applying the crowding principle to genetic search. The model is shown to facilitate a component based approach to localised generational competition where discrete functionalities are defined as independent and interchangeable, reminiscent of genetic operators in the canonical genetic algorithm. The idea of simple crowding is proposed that is shown to be predicted by the model. It is further shown that the reviewed crowding based algorithms are approximations to and variants of, the proposed simple crowding concept represented by the generalised crowding model.

## ***Crowding***

The inspiration for the De Jong's original crowding factor model was a biological metaphor for controlling growth or dominance of fit building blocks in the population. The base ecological metaphor for crowding was shown to imply that as similar organisms dominate an ecological niche, the competition for limited resources increases, resulting in lower life expectancies and birth rates in the subpopulation. The practical goal for the algorithm was to delay convergence and increase diversity in genetic search. The implementation of crowding was shown to be abstracted to the concept of restrictive replacement where members from the current generation compete for a limited place in the population with the previous generation based on similarity.

Mahfoud devised the concept and phase "restrictive replacement" influenced by the crowding factor model, as restrictions were required as to how offspring replaced members of the parental generation. Mahfoud further refined the definition of restrictive replacement by imposing fitness based competition before replacement, which was adopted in other crowding based niching genetic algorithms. Generational competition based on similarity is the common theme in the commonly accepted crowding based algorithms that have been discussed. The underlying theme and its goals have remained constant, though as has been shown, has been implemented and augmented in varying ways.

Restriction during replacement was shown to control the growth or domination of fit building blocks meaning the approach provides a way of controlling convergence. Mahfoud indicated that crowding could be approximated by pairs of offspring competing with their parents, because a significant majority of the time a parent was the most similar member of the parental population to an offspring. Mahfoud's definition of crowding centred on approximating intra-niche matching between offspring and parental generation. Then, a ranked based binary fitness tournament was used to control the quality of solution that is maintained within the localised area. He referred to this concept

as maintaining useful diversity, an effect of which is referred to as the promotion of useful diversity – or niching.

In restricted tournament selection (RTS), Harik [17] used stochastic sampling with replacement to bias tournaments towards intra-niche matching between offspring and parental generations. The bias could be equated to Mahfoud's approximation in deterministic crowding in that an error term is involved, except RTS provided the flexibility of replacing non-parental members. The multi-niche crowding (MNC) technique proposed by Cedeño et al and Cedeño also provided bias in the replacement towards intra-niche matching, though the sampling method was more elaborate and a rank based fitness tournament was not used. Both the RTS and MNC techniques provided the bias, though recommended usage of the techniques by their respective authors kept the bias relatively low when compared to deterministic crowding.

The MNC technique was the only one of the four crowding based algorithms capable of niching that was shown not to provide features of a steady state genetic algorithm under full crossover. This is caused by the complete acceptance of offspring without fitness tournament as was seen in the crowding factor model. Under full crossover, the crowding factor model has been shown by Mahfoud to be unable to maintain more than two niches on simple one-dimensional test problems. The lack of fitness tournament in MNC is seemingly addressed by the bias towards intra-niche pairing and low disruption crossover operator that keep offspring in close proximity of parents within the search space. Collectively these features have been shown to provide good intra-niche convergence.

The common theme in the reviewed crowding based algorithms is approximating intra-niche matching with the parental generation, and then performing a replacement or holding a competition for replacement. This summary of crowding based niching shows that replacement is an important part of a crowding based algorithm, though a more succinct description would be localised generational competition. The crowding principle requires localised competition for limited resources. Crowding based algorithms implement or apply this principle in genetic search by approximating similarity between a competitor and a similar member from the parental generation (localisation). Direct replacement of this localisation is insufficient to meet the principle of crowding, rather competition is required which is implemented by using relative fitness in some way. The term localised generational competition is a refined description of the manner in which the crowding principle has been implemented in genetic search which provides a niching effect desirable in multimodal problem domains.

### ***Generalised Crowding Model***

Localised generational competition was shown to be the common theme in the reviewed crowding based algorithms. Given this succinct definition of crowding based niching genetic algorithms, it is possible to hypothesise an abstracted general model to describe common crowding implementations. Commonality between the reviewed algorithm implementations was implicit during their description. This section aims to unify their commonality in an abstracted conceptual model or framework for the application of the crowding principle to genetic search.

The following listing provides an abstracted conceptual model of localised generational competition called the generalised crowding model (GCM).

1. Initialise population
2. Evaluate fitness of population
3. Loop until stop condition
  - a. Select reproductive set**
  - b. Recombine to produce offspring
  - c. Perform mutation (optional)
  - d. Evaluate fitness of offspring
  - e. For each offspring
    - i. Select member of parental generation to replace**
    - ii. Perform replacement with selected member**

Figure 7 - Pseudo code of an abstracted generalised crowding model (GCM)

The abstracted conceptual model provides a template for existing crowding based niching genetic algorithms, generally encompassing their functionality. Further, it provides a framework for localised generational competition in genetic search. The three components of the generalised crowding model (GCM) that are critical to meeting the goals of the crowding principle are as follows:

- 3a: The selection of the reproductive set known as upfront selection
- 3ei: The localisation selection strategy
- 3eii: The replacement strategy

## Upfront Selection

A canonical genetic algorithm uses selection to blindly direct the search through the problem space. Individuals are selected for reproduction by some means that provides a bias towards fitter members of the population providing direction to the search. The genetic material of the selected individuals is recombined and manipulated to produce new samples in the problem space that have elements of the individuals originally selected.

The principle of selection remains the same in the abstracted GCM, only the bias towards fitter individuals is implicit in the base population itself rather than the way the reproductive set is selected fitness-neutral. It is expected that additional fitness bias is unneeded given the strong rank based bias seen in most of the discussed replacement strategies. Fitness-independent selection schemes are used that provide the same exploratory and exploitive properties observed in fitness biased selection schemes.

The deterministic crowding technique used random selection without replacement providing each member of the population the reproductive potential of one. Each member contributes genetic material in a singular pairing and a uniformly random pairing of members provides useful balance between intra-niche (exploitive) and inter-niche

(explorative) recombination. Restrictive tournament selection also made use of random selection, though allowed replacement (reselection of members into the reproductive set). Reselection allows members to contribute more of their genetic material, and prevents other members from contributing to the search each generation. The expected impact of allowing reselection is increased convergence in unpredictable areas of the search space.

Random selection is expected to add balance to the search, though is also expected to provide positive feedback for growth of common building blocks. Multi-niche crowding proposed a more restrictive selection scheme that actively seeks to bias the balance towards intra-niche exploitation. The crowding selection method encourages intra-niche pairing and thus a means of accelerating intra-niche refinement or convergence. Intuitively this suggests use in situations that require increased intra-niche convergence at the cost of inter-niche exploitation of genetic material.

### **Localised Selection Strategy**

Given offspring or generated samples in the search space, the crowding principle indicates that localised competition is required to determine their survival. De Jong defined an implementation of competition as generational where offspring compete for a position in the population with a member of their parental generation based on member similarity. Mahfoud referred to this policy as restricted replacement. The role of the localised selection strategy is for a given offspring to select a single population member from the parental population using similarity. This strategy is therefore biased to selecting population members that belong to the same niche as the offspring in question, promoting intra-niche matching over inter-niche matching.

Given that an offspring competes for survival with a member of their parent's generation, a means is needed for selecting a member using a similarity measure. De Jong proposed using the Hamming distance to compare bitstrings of equal length in terms of their genetic material or genotype. Deterministic crowding and other niching genetic algorithms discussed are capable of using either the genotypic distance or a distance between the real valued decoded genotypes, called phenotypes. Common to function optimisation is the use of Euclidean distance between decoded value vectors of equal length to provide a phenotypic measure meaningful to the problem domain.

The foundation of localisation in crowding is to use similarity to match an offspring with a member of the parental generation so that the offspring is given the opportunity to compete by some means for placement in the population (survival). The localised selection strategy can be considered a matching process on a continuum between random and exact matching. The canonical genetic algorithm performs random matching and direct replacement and is incapable of maintaining more than a single niche. Deterministic crowding provides a matching process which is limited to one of the offspring's parents which is indicated to closely approximate intra-niche matching as Mahfoud showed that 83% of exact matches are of parents.

Restricted tournament selection samples the population with sample sizes a multiple of the number of desired niches. Multi-niche crowding also samples the population, though



provides more bias towards exact matching than RTS. Multiple samples are drawn, each sample the size of a multiple of the number of desired niches. The most similar from each sample then compete with each other, the loser of which is replaced by the offspring. Both techniques involve sampling which introduces error in the matching process. This error provides opportunity for replacement errors to occur when members of one niche replace members of another niche, a measure that deterministic crowding was designed to minimise. The smaller the drawn sample, the higher the error is expected to be.

The recommendation for RTS and MNC is to draw large enough samples to provide opportunity to include a multiple of individuals from each niche. The goal here is still to minimise replacement error, though not requiring exact matching and instead approximating intra-niche matching as closely as possible. Sampling provides a stochastic nature to the intra-niche competition potentially allowing competition between diverse areas of a local niche as well as facilitating occasional replacement error.

## **Replacement Strategy**

Under full crossover, a generational canonical genetic algorithm performs direct replacement of offspring into the next generation so that the entire population is replaced. The crowding principle indicates that offspring should compete with a similar member of the population for survival which is referred to as restrictive replacement. The first part of the restriction is similarity, and in deterministic crowding, restricted tournament selection and probabilistic crowding an additional rank based binary fitness tournament is performed as the second part of the restriction. Blind or direct replacement based on similarity is restrictive by nature, though it does not conform to the accepted idea of competition in genetic search which is fitness based. Further, as discussed the selection schemes used in crowding based niching genetic algorithms are fitness-neutral where the population is considered to be implicitly fitness biased. For this implicit bias to be facilitated, restrictive replacement needs to incorporate fitness in some way.

The localised selection scheme employed provides a replacement opportunity for each offspring sequentially. The role of the replacement strategy is to make a distinction as to whether or not the offspring can take the contested position. All replacement strategies employed in reviewed crowding algorithms perform a complete replacement in that the replacee is discarded and the replacer takes the free position in the fixed sized population. Deterministic crowding and restricted tournament selection perform a rank based fitness tournament to decide the replacement outcome in a deterministic way. Probabilistic crowding uses a fitness proportionate stochastic selection to determine if the offspring replaces the population member. The crowding factor model and multi-niche crowding perform a direct fitness-independent replacement, though MNC does always replace the worst among most similar selected individuals.

Using a ranked based fitness competition to further restrict the replacement is greedy and implies the discarding of offspring which is a wasteful concept in terms of computational effort to generate them. Blind replacement is complete acceptance of offspring without regard for increase or decrease in niche or population fitness and is not appropriate unless fitness was used in the localised selection process. Probabilistic replacement is not

deterministic by definition and though it is heavily biased towards fitter individuals in the selection process, it provides opportunity for replacements to not occur, potentially discarding fit offspring. A stochastic component is needed in the search, though the replacement strategy may not be an appropriate place unless the localised selection process is quite rigid, as in deterministic crowding.

For a diagrammatic representation of the generalised crowding model, the reader is directed to Appendix C – Generalised Crowding Model.

## ***Discussion***

The generalised crowding model (GCM) provides a simple abstracted understanding of how the underlying crowding principle has been and can be employed in genetic search. The concept of crowding requires localised competition for survival that encourages inter-niche interactions and discourages inter-niche competition. The goal of applying the crowding principle to genetic search is to maintain diversity and promote useful diversity thus providing a niching effect useful in multimodal problem domains. The generalised crowding model provides a simple overview of how restrictive replacement has been used in a number of crowding based niching genetic algorithms, in what can be referred to as localised generational competition.

## **Components**

The three main components that were identified to be critical to the implementation of the crowding principle were upfront selection, localised selection and replacement. All three of these components were shown to exist in the reviewed crowding algorithms and were implemented in varying ways to achieve the same goal of localised generational competition. Though the three mentioned components of the model are critical to parallel niching, it does not mean that other components common to genetic search are not critical to the operation of the general technique.

Niching algorithms that are based on the principle of crowding have been shown to be concerned with maintaining the existing mixture of classes, rather than with how the mixture of classes is provided. This fundamental feature of crowding implies reliance on the initial population distribution. A uniformly random population distribution was shown to be required in all reviewed crowding algorithms and is expected to be requirement of the general approach.

The reliance on population distribution further implies a minimum population size for a given problem. A heuristic used by Mahfoud in deterministic crowding was 20 members per desired niche. A population individual is considered a member of an optimum class if it is within the region of the optimum, such as within its basin of attraction, if the peak were considered an attractor. Crowding does have exploratory features, though primarily its strength has been shown to be its ability to maintain diversity or the mixture of classes. This implies that the random population distribution needs a sufficient number of samples to distribute a minimum number of members within each desired optimum's basin of attraction. This was a premise used by Mahfoud in the specification of his population size heuristic.

A final critical element not yet discussed is the manner in which new samples are generated, that is how offspring are produced. One-point crossover was common for reviewed crowding algorithms as its disruption to the genetic representation is minimal. Multi-niche crowding employed a niching centric recombination operator called interval crossover, which was shown to further decrease the disruption of genetic representation. The generation of new samples is critical to the search process. Given a replacement strategy that uses fitness, the quality and diversity of the samples generated needs to be useful to the search; otherwise they stand a high chance of being discarded by the greedy algorithm.

### **The Model**

The generalised crowding model is a simple abstraction, though it embodies an implementation of the fundamental crowding concept devised by De Jong referred to as localised generational competition. The simplification of the concept and its relations to practical implementations of crowding based niching genetic algorithms raises questions as to the practical effectiveness of the underlying principle.

The model implies that an exact implementation of the crowding principle is possible using elements and knowledge provided by reviewed crowding algorithms. The simplification provides a means of partitioning crowding-centric functionalities, treating each as a component or crowding operator that can be freely interchanged in a manner similar to the genetic operators in the canonical genetic algorithm.

The model implies that crowding is achieved in genetic search through localised generational restrictive replacement. Deterministic crowding and restricted tournament selection and to a lesser degree multi-niche crowding showed that restrictive replacement can be made the central theme of genetic search by moving the fitness bias into the replacement strategy. The model showed that the reviewed crowding algorithms have not provided an exact implementation of the crowding principle directly. In all cases the principle is approximated, sampled or varied.

Given a quintessential implementation of crowding as defined by the model, further questions are raised regarding the behaviour of the algorithm and the effect the crowding operators employed in reviewed crowding algorithms have on the search. Important to the understanding of the model are questionings into:

1. The type of the replacements that occur during search
2. The quantity of replacements that occur during search
3. The nature of samples generated during the search

Answering these questions will provide further insight into the validity of the underlying crowding model, and thus the reviewed implementations of the approach. For example, deterministic crowding is based on the principle of minimising replacement error and approximating local competition by replacing parents. Peak dominance is indicated to be a crowding artefact, observed in the application of deterministic crowding, and indicated to be minimised in other algorithms due to sampling during local competition. Multi-

niche crowding provides upfront selection and recombination operators that are biased towards preventing the production of hybrid offspring to improve the search.

It is expected that a better understanding of the quintessential crowding implementation will facilitate a better understanding of practical application of common crowding based niching genetic algorithms. Understanding the effect a crowding operator has on the search provides more information as to how that operator can be tuned when applied to practical problem domains.

## **Quintessential Crowding Investigated**

This section defines and proposes an implementation of simple crowding which provides an embodiment and archetype of the generalised crowding model. The proposed algorithm is investigated in detail in terms of the number and nature of replacements, the nature of pairings and their contribution to search, and finally in terms of replacement waste and replacement error. Insights are provided into how parental and non-parental replacement behaviour is not static, and is shown to change over the course of the run. Results further raise questions as to the need for specialised crossover operators in crowding and questions are raised as to the impact of individual-pair biases in upfront selection. Finally an analysis of matching behaviour and unused offspring provide insight into the algorithms true nature. A simple crowding replacement error measure is proposed that implies that approximation of intra-niche replacements may not be sufficient and that localisation of replacement within a niche may be the desired behaviour to approximate in crowding based niching algorithms.

### ***Simple Crowding Algorithm***

Simple crowding could be considered a direct implementation of the simple generalised crowding model that closely adheres to the crowding principle's implementation in genetic search called localised generational competition. Simple crowding could be interpreted as absolute similarity in the matching of an offspring to a member in the parental generation. As has been described, restrictive replacement was extended to require fitness based competition. Fitness tournaments at replacement time provides the competition component to the localised generational competition concept giving simple crowding a natural fitness bias. This further requires that an upfront selection policy be fitness neutral, respecting the implicit fitness bias provided in the population.

Development of the deterministic crowding algorithm involved isolating the cause for the crowding factor model's inability to maintain multiple niches. It has been shown that the investigation involved removing elements of noise from the algorithm to investigate the underlying behaviour of the algorithm which aided in isolating and minimising replacement error. The crowding based niching algorithms reviewed all introduced noise to the underlying crowding principle. In this case noise is considered a factor that disrupts or variates from the principle of localised generational competition in some way.

To this end, simple crowding attempts to remove noise from the search process. A similar iterative approach as that used in development of deterministic crowding algorithm was used to develop an implementation of the simple crowding algorithm. The goal of the algorithm was to verify the generalised crowding model and the underlying crowding principle that provides niching in genetic search. This is a different goal to that of developing an extension to the crowding factor model for niching as was the case in deterministic crowding, that sought to minimise replacement error and maximise the number of maintained optima.

Restricted tournament selection was used as the basis for this iterative development of simple crowding as it provided a close match to the goals of algorithm already defined.

Mutation was not used as it is expected to provide local hill climbing functionality which is not of interest in this investigation. The specific interest in this section is firstly the ability of the algorithm to maintain existing diversity, and secondly to promote useful diversity. These two points are the identified goals of the localised generational competition algorithm and are expected to be strongly exhibited in the simple crowding algorithm.

Testing was performed on the simple F1 and F2 test problems (see Appendix A – Test Functions) to provide a preliminary indication of the algorithms capability and embodiment of the crowding principle. All test results were taken as mean scores over 100 runs. An extended fixed run period of 500 generations was used in all cases in an effort to expose potential diversity maintenance issues. The ability to maintain diversity was measured by monitoring the change in population distribution throughout the run. In this case each population member was classified by the closest desired optima in the domain, and the tallied subpopulation sizes were captured. The algorithms ability to promote and maintain useful diversity was measured as the quality of the result at each desired optima, summarised by the maximum peak ratio.

### **A1 – Full Crossover**

The restricted tournament selection algorithm provided the base for this algorithm. The stochastic generational gap was removed by requiring the crossover percentage to be 100% as is the case in deterministic crowding. It has been shown that a belief in the construction of deterministic crowding was that the crowding principle should be able to work under full crossover.

Less than one hundred percent crossover provides a stochastic generational gap that allows more prevalent areas of the search domain to be sampled more often than without the gap. The increased attention to prevalent areas of the domain is expected to increase convergence. Removing the stochastic generational gap provides an opportunity to observe the intra-niche convergence provided by the base algorithm. In the case of both F1 and F2 a sample size of  $4 \times P$  was used where  $P$  represents the number of desired optima in the problem.

In terms of subpopulation stability, the results of this algorithm were less than desirable. Results for F1 showed that subpopulations were relatively stable for the first 100 generations at which point peak dominance began to degrade subpopulation size. From the halfway mark until the end of the run peaks 2 and 4 steady increased in size, syphoning individuals from peaks 1, 3 and 5. Similar results were seen on F2 though stability was only held for approximately 30 generations before variation began. In this case peaks 1, 2 and 3 showed a gradual increase which peak 4 showed a gradual decrease in size. Results showed that the subpopulation at peak 5 crashed consistently by the end of the run, with on average a size of 0.18 members.

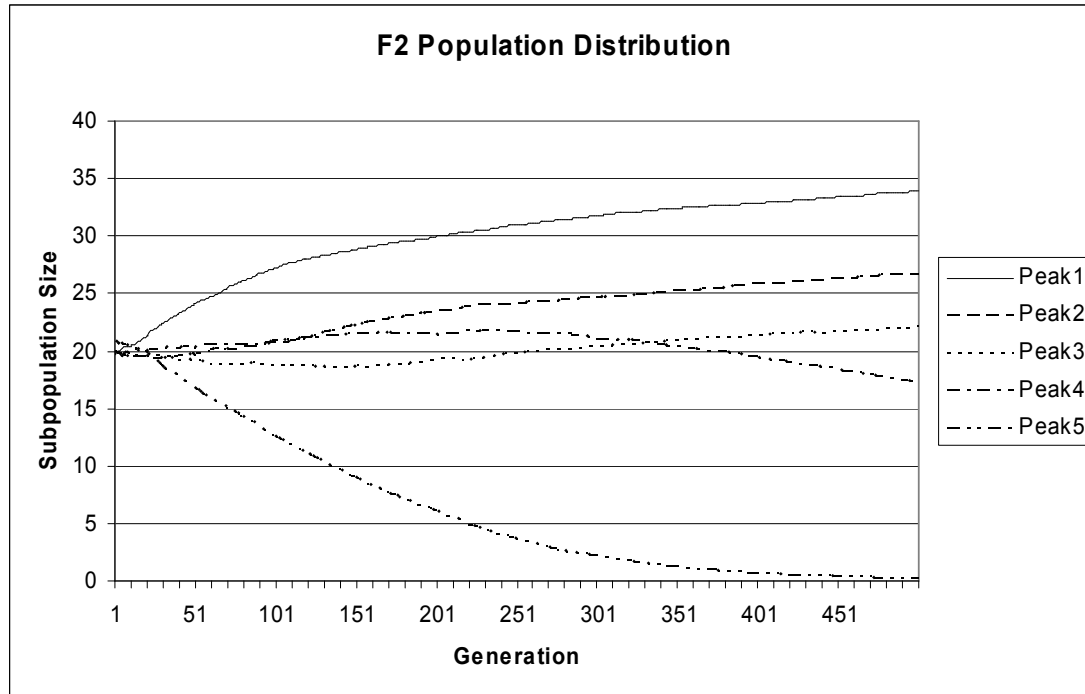


Figure 8 – Shows the subpopulation sizes at each of the five peaks in the F2 test problem for A1 (mean over 100 runs)

In terms of subpopulation quality, results on both functions showed rapid improvement to 100% of ideal within approximately 10-15 generations. In both cases after 100 generations, the summarised quality dropped a fraction. The results show a stepwise decrease in maximum peak ratio (MPR) in F1 over the course of the run. Results for F2 show a steady decrease in MPR over the last 250 generations of the run, corresponding with the crash of the subpopulation at Peak5.

## A2 – Population Wide Sampling

The results of A1 showed that populations are maintained without a stochastic generational gap, though general population diversity and useful population diversity showed signs of instability. The instability observed was over a moderate 500 generations on simple test problems and if the run were extended, poor stability is expected to continue and lead to further subpopulation crashes. Simple crowding is expected to exhibit strong stability, specifically on simple test problems over moderate and long durations. The select migration of samples between peaks is reminiscent of the class dominance behaviour shown to occur in deterministic crowding.

The A2 algorithm is the same as A1, except the sampling used in the localised selection strategy is removed to provide population wide matching of the most similar individual to each offspring. Sampling is expected to introduce noise in terms of intra-niche replacements, a reduction of which is expected to resolve stability issues.

The results for A2 on F1 and F2 show stable subpopulations from approximately generation 50 and complete subpopulation stability at approximately generation 104 and

60 on F1 and F2 respectively. Unlike A1, the population distribution results for A2 show minor movements within the first generation, until a point of equilibrium is reached and the population sizes collective remain fixed from that point onward. In the case of both problems, populations remain relatively close to their initial sizes with minor variations during the opening stages.

Results for maintaining useful diversity are also very stable. In both cases a sharp increase was observed over the first 15-20 generations, to the problems maximum. Unlike A1, the maximum peak ratio scores remained constant throughout the duration of the run from generation 100 and 85 on F1 and F2 respectively.

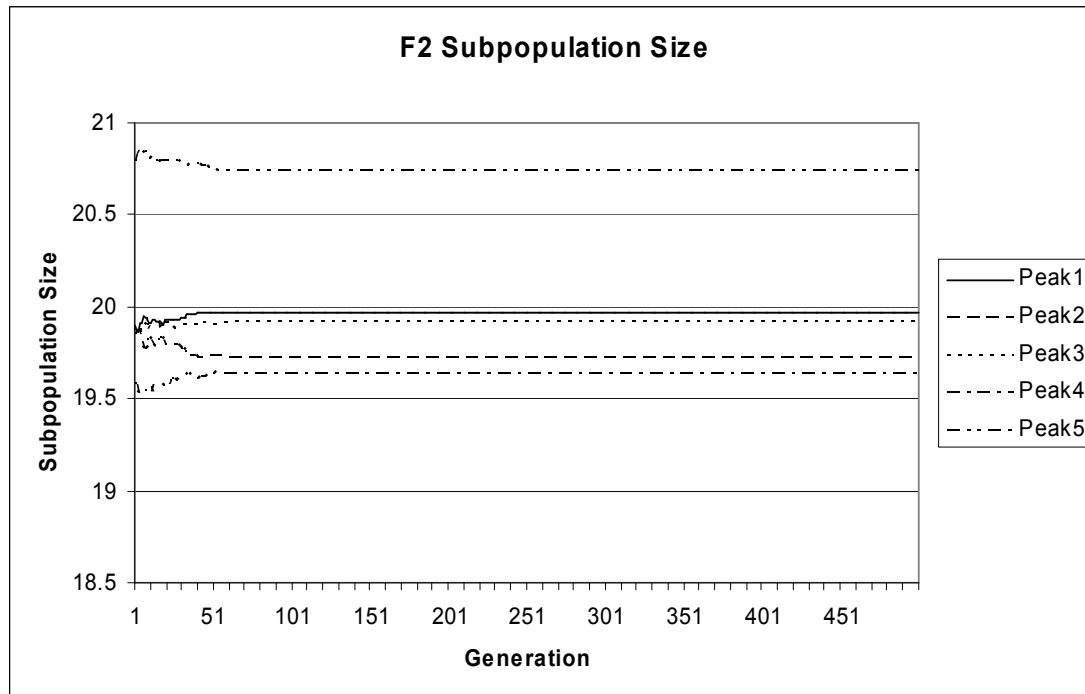


Figure 9 – Shows the subpopulation sizes at each of the five peaks in the F2 test problem for A2 (mean over 100 runs)

### A3 – Uniform Selection (Simple Crowding)

The test results for A2 show that absolute sampling or exact matching provides both stability in subpopulation size (diversity) and quality of result (useful diversity). A2 uses random selection with replacement allowing members to be selected more than once into the reproductive set. Deterministic crowding uses random selection without reselection providing the reproductive potential of one to each population member. Given that fitness is used in replacement implies the population maintained is implicitly fit. Each population member is potentially exposed to constant competition. To reinforce this concept of constant competition, simple crowding should allow each population member to contribute genetic material in the recombination process. The A3 algorithm is the same as A2, except random upfront selection is used without replacement.



The results for A3 show stable subpopulation sizes which exhibit very similar behaviour to A2 on the same test problems. Convergence is increased slightly on F1, and delayed slightly on F2, where complete stability in subpopulation size is achieved after approximately 90 and 80 generations on F1 and F2 respectively. Further, A2 was shown to provide a slight dampening of size fluctuations over the first 100 generations compared to A3. Maximum peak ratio scores are also practically the same as was observed with the A2 algorithm on the same test problems with rapid convergence over the first 15-20 generations, and stability of score after subpopulation size equilibrium was reached.

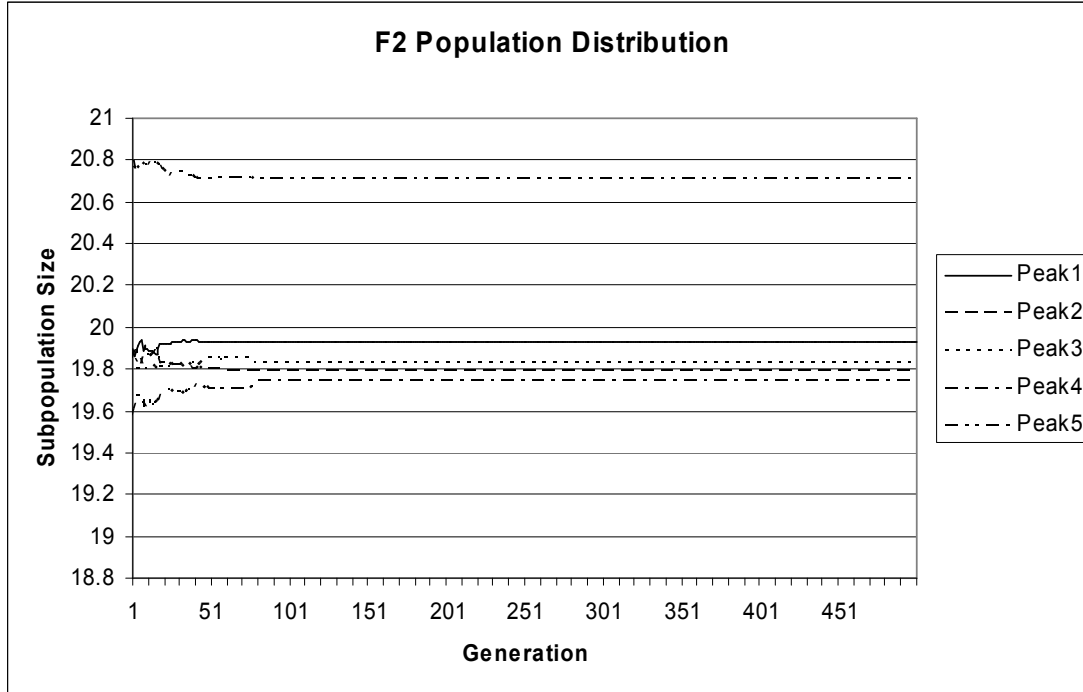


Figure 10 - Shows subpopulation sizes on the F2 test problem for A3 (mean over 100 runs)

## Discussion

Three crowding algorithms were tested; each improving on the last in an effort to isolate a simple crowding implementation which matches the behaviours expected by the generalised crowding model at least as far as F1 and F2 are concerned. The model provided a restricted replacement focused implementation of the crowding principle for niching genetic algorithms. The behaviour expected by a exact implementation of the crowding principle was maintenance of diversity and the promotion of useful diversity.

Components that were expected to introduce noise and potentially obscure the underlying crowding algorithm were removed. The first of which was the removal of the stochastic generational gap in A1. Results showed concerns of stability in terms of its ability to maintain subpopulations over a moderately long duration. This concern was reinforced by the matching decrease in the quality of result. A simple crowding algorithm was proposed to employ absolute similarity matching between an offspring and its parental generation, therefore the stochastic sampling used in the localised selection strategy was replaced

with population wide sampling in A2. Results showed that the simple change provided subpopulation stability that was achieved rapidly and maintained at static levels until the end of the run. Stability was mirrored in the quality of result maintained by each subpopulation.

Finally the A3 algorithm sought a refinement to A2 by employing a random upfront selection scheme without replacement. The goal was to ensure all population members participated in recombination once and thus potentially improve the search and facilitate population wide competition each generation. Results between A2 and A3 were practically indistinguishable, showing a minor dampening effect in initial subpopulation size fluctuations with A2, and differences in complete stability times. The stability discrepancies can be speculated to be caused by the uniform nature of random selection without replacement that provides more consistent population wide competition for survival. The final algorithm A3 was shown to meet the goals set out for simple crowding in both design and result. The implementation of the algorithm required absolute similarity matching and the removal of potential sources of noise during upfront selection and localisation selection. To this end, the algorithm can be considered an implementation and embodiment of the generalised crowding model.

The following recapitulates the specific implementation details of the simple crowding algorithm in terms of the generalised crowding model:

1. Upfront Selection: Random selection without replacement
2. Localised Selection Strategy: Population wide sampling for most similar member
3. Replacement Strategy: Rank based binary fitness tournament

The validity of the abstracted model is verified by the results of the algorithm. The A3 algorithm represents an archetype for localised generational competition. The primary goal of the archetype is the maintenance of general diversity and the promotion of useful diversity which has the effect of niching subpopulations around optima in the search space. Preliminary test results showed that A3 was shown to maintain stable subpopulations for the duration of the run on both F1 and F2 test problems. The algorithm A3 from this point will be referred to as the simple crowding algorithm.

### ***Nature of Replacements***

The generalised crowding model is an abstraction of localised generational competition based algorithms, historically called restricted replacement algorithms. Replacement is a large part of crowding based implementations and involves restriction in terms of member locality in the search space and restriction in terms of member fitness. Enforcing restrictions on replacement raises questions as to the nature of replacements that are permitted to occur. Deterministic crowding has been shown to be an approximation of the crowding factor model where only parents are permitted to be replaced by their offspring. The simple crowding algorithm is an implementation of abstracted generalised crowding model that makes no assumptions regarding the member of the parental population that is replaced. It requires only that the most similar individual to the offspring in question be selected in the localised selection scheme, and compete based on a fitness ranking.

Given the localised replacement strategy used in simple crowding, replacements can be classified as one of three types:

1. Parent Replacement – Offspring replacements one of their two parents
2. Sibling Replacement – Given that replacements occur sequentially it is possible for an offspring to replace a fellow sibling (same generation as offspring) that has already succeeded in a replacement tournament
3. Other Replacement – Offspring replacement that is neither parental or sibling

Understanding the nature of the replacements that occur within the simple crowding implementation is expected to provide insight into the effect the restrictive replacement strategy has on the population and on genetic search. This understanding can then be used to better understand existing crowding based niching genetic algorithms and to better refine their application. It can also be used to refine or design augmentations for the simple crowding algorithm for practical application.

The total replacement results showed that parent based replacements are the most frequent accounting for a total of 52% and 53% of replacements on the F1 and F2 test problems respectively. Other replacements account for a high percentage, showing 41% and 40% of all replacements on the test problems. The remaining 7% of replacements on both problems were sibling based replacements where an offspring was matched onto and replaced member of their own generation that had previously succeeded in a replacement tournament.

The number of replacements each generation was observed over the extended time period. For both problem domains, the number of replacements decreased steadily from the first generation until approximately generation 350 and 390 where the average number of replacements in both F1 and F2 problems dropped below one. The results of replacement types over generation time showed that in both F1 and F2 the number of parental and other replacements became approximately equal at approximately generation 180 and remained at similar levels until replacements dropped below one.

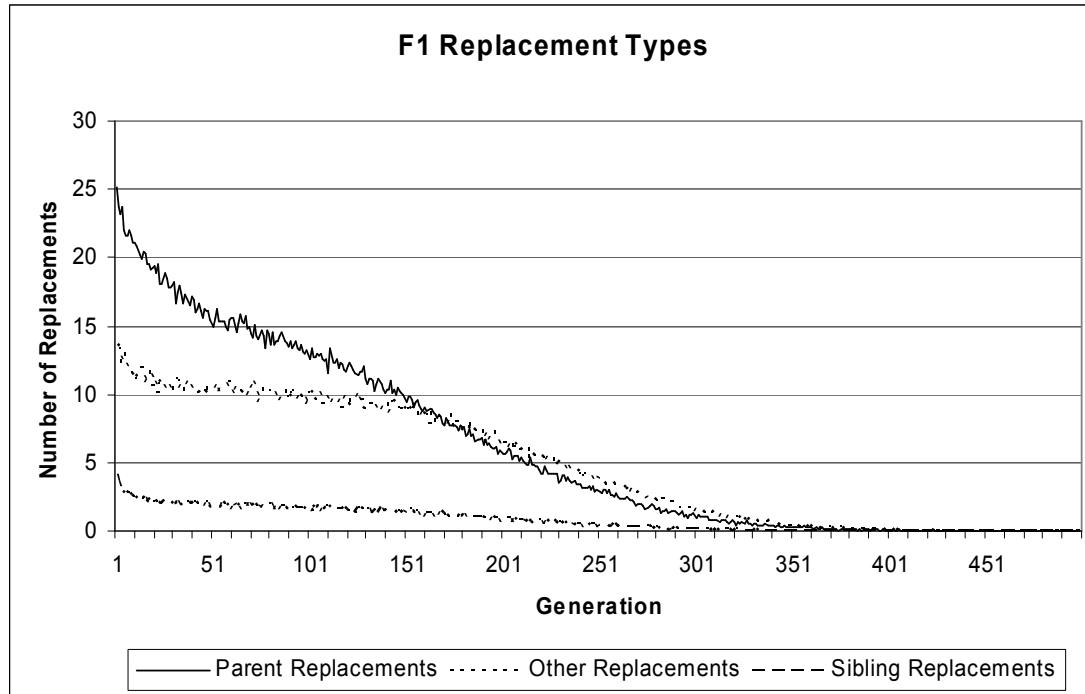


Figure 11 – Shows the number of each replacement type over 500 generations for simple crowding on the F1 test problem (mean over 100 runs)

The results provide useful insight into the nature of replacements in the prefect crowding algorithm. Given that simple crowding is an archetype for localised generational competitive methods, the preliminary results provide a strong indication that parental replacement is the more common form of replacement, though other replacements account for a large portion of the replacements that occur. The results show that sibling replacement on average is very low, accounting for approximately 4% of replacements initially, and dropping to below 1% rapidly. This shows that an explicit means of preventing it in the localised selection of siblings may not be necessary unnecessary.

Deterministic crowding has been shown to be an approximation of the crowding factor model that works by offspring replacing their most similar parent. The results provide no indication as to the actual localised matching that occurred, rather only the actual fitness based replacements that occurred. Given that parental selection was shown to account for on average 52%-53% of replacements, parent based limitations on replacement are not an approximation of the simple crowding algorithm. This further raises questions regarding class dominance shown to occur in application of the deterministic crowding algorithm. It is speculated that the effect is a product of the localised selection strategy, rather than an effect of the crowding principle itself. Converse to the deterministic crowding localised selection strategy, the results indicates that a strategy that encourages a 50%/50% replacements of parents and other members may be useful.

An unexpected product of the results was the observation that the number of replacements decreases to a point where the population remains unchanged. This is the product of intra-niche convergence which has the same effect as global convergence

shown to occur in the canonical genetic algorithm. Each niche ideally converges to a singular point at the summit of their respective optimum preventing any further improvement. An extended amount of time is required to achieve this state, given the global search properties used in the algorithm. Not really useful for practical use, this shows that the technique is capable of reaching a state of minimum change. Further, it indicates that a specialised algorithm stopping condition can be proposed for crowding algorithms that detects the change in the number of replacements that occur per generation.

### ***Hybrids and Normals in Search***

A hybrid is an offspring produced as a by-product of recombining parents that are members of different niches or subpopulations. These hybrids are called lethals in biology because a mating between members of two species typically produces an offspring with low survival and reproductive fitness. The crossover operator used in simple crowding is the standard one-point crossover method. Given two population members from two different niches, the operator is capable of producing offspring that belong to either parent's niche and other niches. Each generation  $\frac{N}{2}$  pairs are formed from  $N$  population members, which are needed to produce  $N$  offspring. All pairings can be classified as one of two types:

1. Inter-niche pairing – different niche pairing where parents belong to different niches
2. Intra-niche pairing – same niche pairing where parents belong to the same niche

Given each type of pairing, two possible classes of offspring can be produced:

1. Normal – offspring belongs to either one or both parents niche
2. Hybrid – offspring belongs to neither parents niche

It was shown that the multi-niche crowding method is biased towards selecting pairs of population members from the same niche. It has also been shown that the technique used a customised recombination operator that limits the area in which offspring produced from a pairing is positioned. The reasoning for these design choices in the algorithm was shown to be the reduction in low fitness hybrids produced during search.

This section investigates the nature of pairings that occur within the simple crowding algorithm. The areas of interest are the number of same niche and different niche pairings that occur over the extended run period on the simple F1 and F2 test problems. Given each type of pairing, two offspring are produced and each can be classified as either normal (same niche as one of its parents) or hybrid (different niche to both parents). Given the gathered pairing information, it is further useful to measure the number of replacements that occur for each pairing type. The number of replacements from offspring produced by each pairing type will provide an indication as to the manner in which each type of pairing contributes to the search over the course of the run.

Four observations were collected for each test problem as follows:

1. Number of same niche and different niche pairings per generation
2. Number of normal and hybrid offspring produced from same niche pairings per generation
3. Number of normal and hybrid offspring produced from different niche pairings per generation
4. The number of replacements permitted for offspring produced from same niche and different niche pairings

The test results for all four observations were nearly indistinguishable between both the F1 and F2 test problems. Simple crowding exhibited the same behaviours with very minor fluctuations during both of the long duration runs. In terms of niche based pairings, different niche pairings were shown to occur the majority of time (80%), leaving the remaining 20% to same niche pairings. The random upfront selection without replacement remained consistent with the ratio's with minor fluctuations throughout both tests, averaging a 40, 10 count of different niche, same niche pairings.

As was expected, the vast majority (98%) on average of offspring produced by same niche pairings also resided in the same niche as their parents. A similar result was observed for offspring produced by different niche pairings where on average 97% of offspring belonged to either one or the other parents' niche. These rates were observed to be consistent throughout the run on both test problems, though the same niche offspring showed a slightly more fluctuation over the duration than different niche offspring.

Given that more pairings were between parents of different niches, more offspring were produced by this type of pairing and the results showed that this resulted in more replacements of different niche pairings than same niche pairings on both test problems. The results show that the majority of successful replacements occurred as a result of different-niche pairings. The average number of replacements over the course of the run was approximately 84% and 17% for different niche and same niche pairings respectively on both test problems.

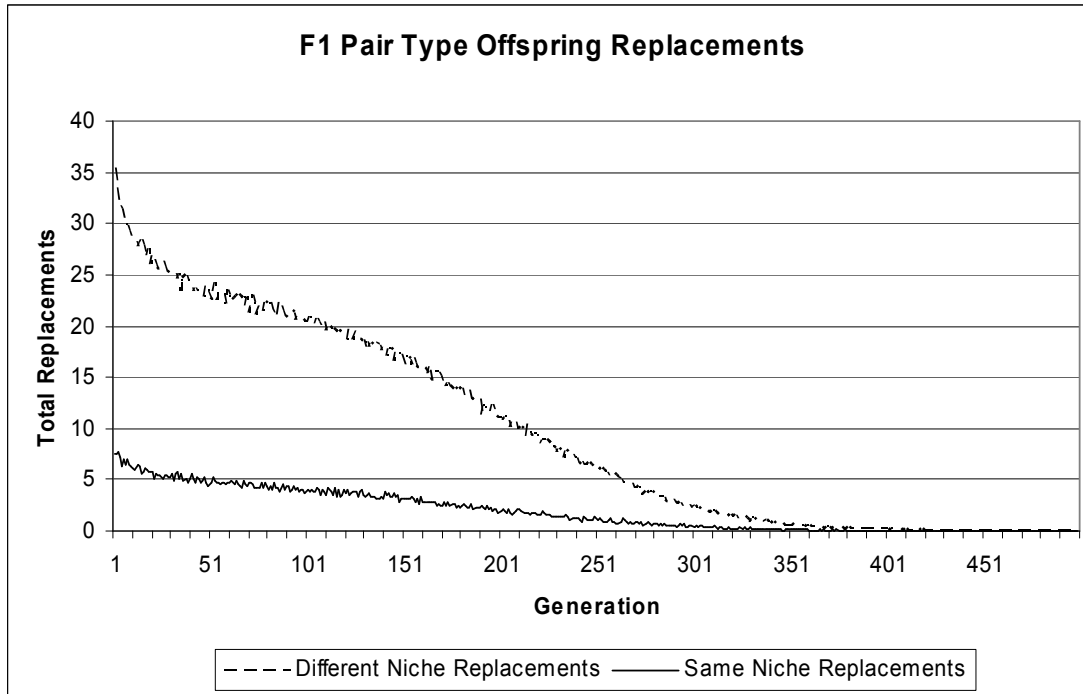


Figure 12 – Shows the number of total successful replacements of offspring generated from both same-niche pairing and different-niche pairings on the F1 test problem (mean over 100 runs)

One-point crossover has been shown to mostly produce offspring that belong to one of their parents' niches. The need for a recombination operator that reinforces this feature in the simple crowding algorithm may be useful, though has been shown in preliminary results not to be necessary. The results showed that different niche pairings are the majority of pairings that occur, and that if intra-niche pairing is useful to search, then a bias during upfront selection is required to accentuate its effect.

Offspring from a different niche pairing received the most replacements, though this is likely biased by the fact that different niche pairings were the more common form of pairings provided by the selection operator. The results raise questions as to the performance of the simple crowding algorithm given a strong bias towards intra-niche pairings as in the multi-niche crowding algorithm. It also raises the question as to the effect on the algorithm of a stronger bias towards inter-niche pairings, or a 50% balance between each type of pairing.

### ***Replacement Waste and Replacement Error***

For each generation in the simple crowding algorithm,  $N$  replacements are possible. Of those possible  $N$  replacements, only a portion will deterministically occur. Like deterministic crowding, the simple crowding algorithm does not use a probabilistic or stochastic process during replacement. Given an offspring, the replacement that will occur is known given a known parental population.

The algorithm is greedy in that only the fittest individual in each tournament is selected and permitted a place in the population. The greedy approach implies that a large portion of offspring produced each generation are discarded, and so do not participate in the search. This portion of discarded offspring can be referred to as replacement waste and is calculated by  $N-R$  where  $R$  is the number of replacements that occurred in a given generation. The deterministic crowding algorithm has been shown to be an approximation of the crowding factor model that seeks to minimise replacement error and maximise peak maintenance. Replacement error is a summary measure that represents the number of replacements that occur where an offspring replaces a member of a different niche to itself. This section provides a minor investigation into the concepts of replacement waste and replacement error observed in the simple crowding algorithm.

## **Replacement Waste**

Replacement waste is a result of the simple crowding algorithms elitist replacement strategy. Discarding samples is wasteful though is the trade-off made to maintain high fitness members in the population. The canonical genetic algorithm discards the entire parental population each generation given full crossover. Restrictive replacement provides a means of evaluating an offspring in terms of its contribution to the search and discarding it if the contribution is not higher than the contribution offered by existing population members. The offspring is not generated and discarded; rather it is generated and evaluated in relation to the existing population of samples in terms of usefulness to the search. Therefore it could be said that to achieve simple crowding the algorithm must be greedy or elitist, but this behaviour is not necessarily wasteful.

It is reasonable to evaluate wastefulness on a continuum defined by the usefulness of the offspring generated by pairings of population members. The result of pairing types that recombine to produce offspring that are not useful to the search could be considered wasteful. The more of the pairings and resulting offspring that exhibit this behaviour the more the algorithm could be considered wasteful. Therefore, a practical implementation of simple crowding requires a sample generation processes that is useful to the search, which maximises the use of offspring before convergence.



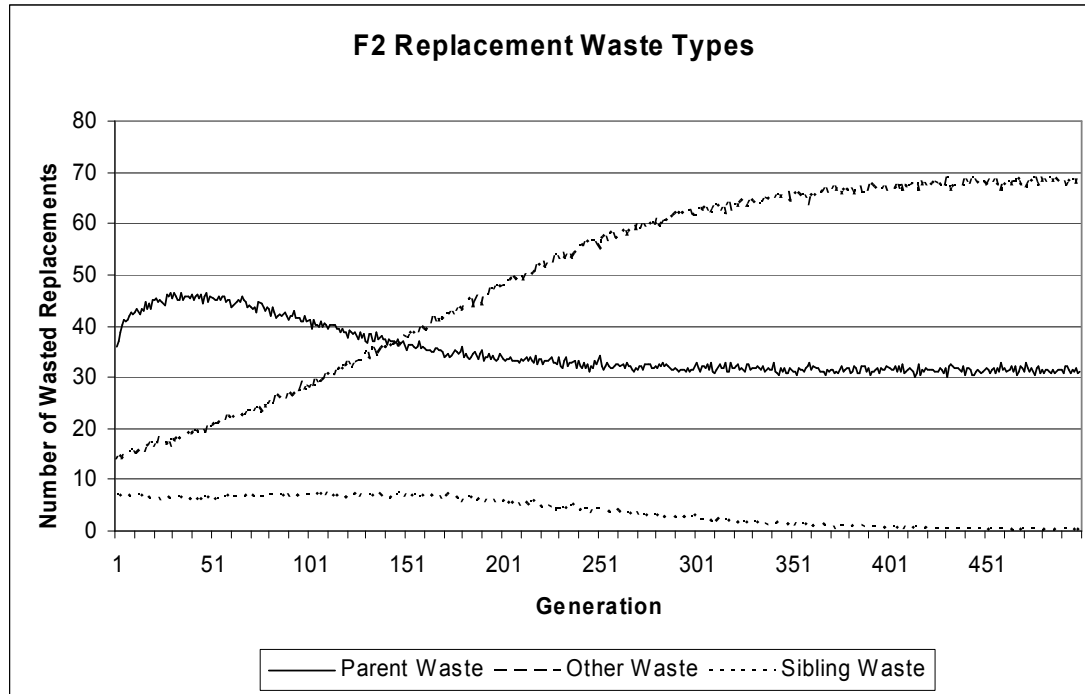


Figure 13 – Shows the number of wasted replacements of each type for the F2 test problem (mean over 100 runs)

Unlike the replacement statistics common to both test problems, the results show that on average the majority of waste is produced when offspring match to other population members (non-parents and non siblings). Test results show approximately the same results for both test problems where parental replacements accounts for 39% of waste, other replacements account for 57% of waste, and the remaining 4% of replacement waste are from siblings. The results of the total localisation matches that occur indicate that other matches are in the majority with 55%, with parental at 40% and sibling at 5%.

	<i>Replacements</i>	<i>Waste</i>	<i>Total Matches</i>
<b><i>Parent Matches</i></b>	52%	39%	<u>40%</u>
<b><i>Sibling Matches</i></b>	7%	4%	<u>5%</u>
<b><i>Other Matches</i></b>	41%	57%	<u>55%</u>
<b><i>Total Replacements</i></b>	<u>11%</u>	<u>89%</u>	<u>100%</u>

Table 4 – Shows a breakdown of the average replacement waste and average replacements for each type of replacement from the F1 test problem

The tabulated summary shows although other matches were the most common, that parental matches were the more common to succeed, and other matches were more likely to not succeed. The difference in total parental and other matches is relatively close, showing only a 5% discrepancy.

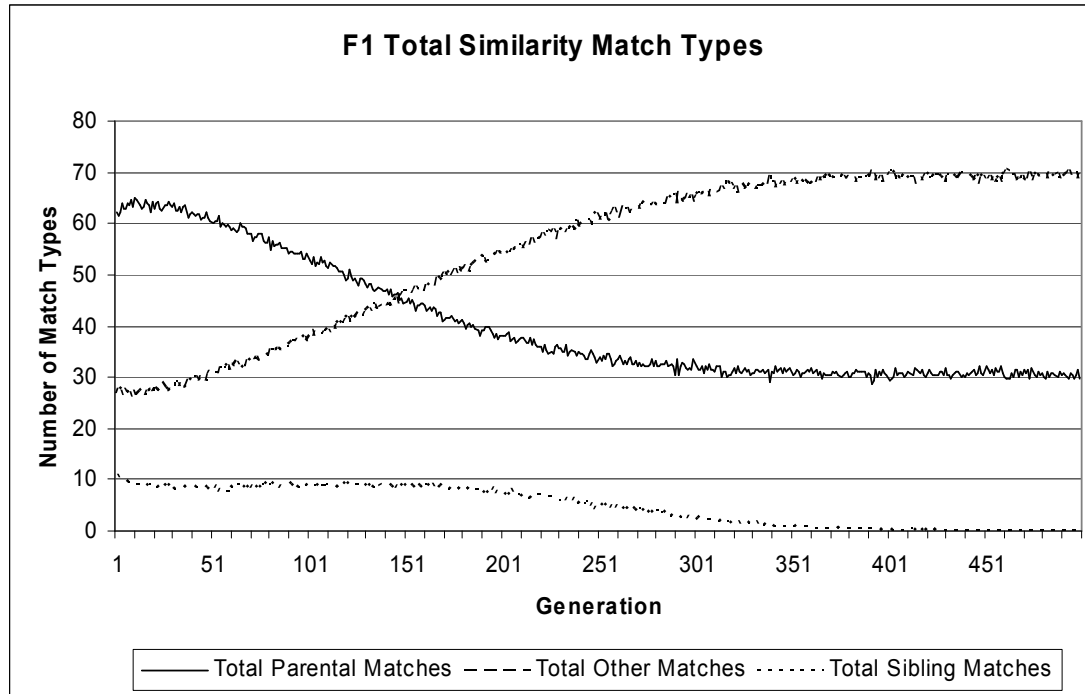


Figure 14 – Shows the number of matches (not necessarily replacements) of each type for the F1 test problem (mean over 100 runs)

Previous replacement results showed that other and parental replacements started off at different levels, though by the half way mark of the run, achieve approximately equal values. The results of waste over the run show change in waste matches the change in matching that occurs throughout the run, primarily differing in magnitude during the opening half of the run. The results indicate that parent based matching and replacement occurs in the opening stages of the run. As the run progresses the number of non-parental matches rises above parental matches, providing similar numbers of replacements until convergence.

## Replacement Error

The simple crowding algorithm does not seek to minimise replacement error, rather it seeks to eliminate it completely. Simple crowding has been shown to imply exact matching during the localised selection strategy, which, given niches means no replacement error. A measure of replacement error is not useful in simple crowding, though would be useful if simple crowding were approximated. Replacement errors measured in both test problems were low to non-existent. During the opening stages on both problems, replacement scores started at less than 0.35% of replacements and quickly dropped to 0% by approximately generation 50.

Replacement of members of the same niche is a generalised version of the simple crowding algorithm which requires replacement of the most similar member of the niche. It is therefore possible to define a new error measure that records when the best match in the niche does not occur. This measure is zero for simple crowding each generation,

though is expected to provide useful information as to replacement behaviour in sampling based replacement methods.

## ***Discussion***

The underlying principle of crowding based niching in genetic algorithms is local competition for limited resources, with the primary goal of maintaining general diversity and promoting useful diversity. This principle has been applied to genetic search using the idea of localised generational competition. The limited resource is the positions in the population and the offspring compete with their parental population for a placing. This form of competition is referred to as restrictive replacement.

The generalised crowding model (GCM) is an abstraction of localised generational competition based algorithms which provides a framework for an algorithm to implement the requirements of the crowding principle in genetic search. The goals of the crowding principle are therefore the same as those of the abstracted conceptual model, as well as the algorithms that are specialised implementations of that model. This goal of promoting and maintaining useful diversity has the effect of increasing similarity within groups and decreasing similarity between groups which is called niching. The boarder goal of maintaining diversity requires that the size of the groups be maintained with stability over an extended duration.

It has being shown that the reviewed crowding based algorithms are specialisations of the generalised crowding model that approximate its behaviour with the intent of achieving the two defined goals of the crowding principle. It was proposed that a simple crowding algorithm could be designed and implemented that met both of the defined requirements of the crowding principle exactly and could therefore be considered an embodiment of the generalised crowding model and an archetype of the crowding principle applied to genetic search.

An iterative design process was used to transition the restricted tournament selection algorithm into a simple crowding algorithm. Stochastic generational gap was removed and the localised selection process was made entirely deterministic providing exact similarly matches between the offspring and parental generation. Upfront selection was made stochastic and required that each population member contribute their genetic material in a single genetic recombination per generation. The results of the algorithm showed complete stability of subpopulations meeting the broad goal of maintaining general diversity. Results also showed that useful diversity was promoted and maintained through the convergence within each subpopulation to the local optimum.

The simple crowding algorithm was then used as the basis of a number of investigations into fundamental crowding behaviour in genetic search. The nature of replacements that occurred within the algorithm each generation showed that offspring replaced their parents the more often on average, though replacement of non-parent members occurs almost as frequently. This raised questions as to the desirability of replacing parents only as is the case in deterministic crowding. Results also showed that the number of replacements dropped off to nothing indicating a point of absolute stability or

convergence in the population. The results showed that simple crowding can maintain diversity during search or the formation of niches, and during convergence or the exploitation of niches.

A detailed investigation into the combined effects of random selection without replacement and one-point crossover revealed an insight into search in simple crowding. Results showed that the majority of pairings that occurred were of parents from different niches and that this causes the majority of offspring produced and replacements that occurred to be the product of inter-niche interactions. Results showed that a significant majority of offspring produced from same niche and different niche pairings were normal, meaning they belonged to either one or both parents' niche. These results raised the question for the need for a specialised recombination operator biased towards producing same niche offspring as was shown to be used in multi-niche crowding. The results also raised questions as to the effect of strongly biasing pair selection in favour of same-niche or different-niche pairings would have on the simple crowding algorithm.

Finally the issues of replacement waste and replacement error were discussed in the context of simple crowding. Replacement waste was seen as the ineffective use of population members to produce offspring related directly to the upfront selection and recombination operators. Replacement waste was shown to be measured by  $N-R$  where  $R$  represents the number of replacements for a given generation. Replacement waste was shown to be high in simple crowding, explained by the greedy or elitist deterministic localised selection and replacement strategies used.

The results showed an interesting transitional replacement/replacement-waste effect where parents were replaced more at the beginning of the run, and other members were replaced at a similar level as parental-replacements after approximately 180 generations. Results further showed that most of the matching at the beginning of the run was parental, supporting the higher successful replacements, and that as the run progressed parental matches decreased and the number of other matches increased above parental based matches, providing cause for the rise in successful other type replacements. This subtle yet complex transitional behaviour is believed to play an important role both in the algorithms search strategy and its ability to achieve a state of sustainable stability and minimal change.

A goal of simple crowding was not to minimise replacement error, but to eliminate it. This goal was implied by the requirement of simple crowding to achieve exact matching between an offspring and their parental generation. Results showed that replacement error did exist in the opening stages of search during niche formation, although it accounted for on average 0.004% of replacements that occurred over the course of the run.

In terms of subpopulation stability, the preliminary results of the simple crowding indicate that there may be a relationship between replacement error and replacement waste which may be a cause for subpopulation size instability issues. An increase in error is expected to provide an increase in convergence, given that more replacements are expected to occur (reduced replacement waste). A trade-off is believed to exist between

subpopulation stability at the cost of increased convergence time (replacement waste), and decreased convergence time (decreased replacement waste) at the cost of decreased subpopulation stability. In addition, a new exact replacement error measure that better represents the goal of exact similarity matches in simple crowding. It is speculated that a reduction in conventional replacement error is insufficient to achieve stable subpopulations; rather a minimisation of exact replacement error is desired. This speculation further implies that the approximation of intra-niche matching is too broad, and that the approximation should be of specific locality within a niche.

The investigation into the simple crowding algorithm revealed a number of useful analysis tools that are applicable to crowding based algorithms. The two basic measures of the crowding algorithms ability to achieve the goals of the crowding principle were the stability of subpopulations (maintenance of general diversity), and the quality of subpopulation (promotion of useful diversity). Crowding is implemented in genetic search through localised generational competition, a portion of which is referred to as restrictive replacement. Given a generation, a fixed number of replacements are possible. Measuring the quantity of replacements that succeed and fail provides an indication of the effectiveness of the algorithm in terms of search efficiency. Evaluating the nature of the replacements and replacement waste such as parent, sibling, other and error provides insight into the behaviour of the search. Finally, the effectiveness of the search can be analysed by evaluating the effects of the upfront selection and recombination operators. The nature of paring and offspring replacement provides insight into the behaviour of offspring production, and analysis of pairing based replacement provides further insight into the sample generation behaviour on the effectiveness of the search.

Simple crowding was shown to an embodiment of the crowding principle and an archetype implementation of the generalised crowding model. Given that the reviewed crowding based algorithms can be considered approximations or specialisations of the simple crowding algorithm, it is expected that the insights provided into the nature and behaviour of simple crowding can be used to both analyse and tune the behaviour of crowding based algorithms.

## **Augmented Simple Crowding**

This section provides an investigation into the effect specific components from the generalised crowding model used by reviewed crowding based algorithms have on the simple crowding algorithm. Biased upfront selection schemes are investigated in detail showing that an intra-niche bias delays convergence, and that a bias towards inter-niche pairings is more beneficial to the search.

The effect of sampling during localised selection is evaluated on simple crowding. Results show that minimisation of replacement error does not imply subpopulation stability, and rather the goal of localised sampling should be the minimisation of exact replacement error. Finally three alternate replacement strategies are investigated, the results of which show that exact matching in the localised selection strategy is sufficient to maintain population diversity, with fluctuations. Further results show that although general diversity can be maintained, strong fitness based competition is required to meet the goals of the crowding principle in genetic search.

### ***Bias in Upfront Selection***

Upfront selection controls how the search proceeds. It is a genetic operator which chooses which population members can reproduce, and the pairings of population members in the reproductive set. In the canonical genetic algorithm selection was seen as an abstraction of natural selection that biases reproduction towards solution fitness. This bias was shown to be implicit in the generalised crowding model, therefore fitness-neutral upfront selection schemes are employed to prepare the reproductive set and internal set pairings.

A preliminary analysis was performed into the nature of random upfront selection without replacement (referred to as uniform selection) and its impact on replacements. Results showed that the majority of pairings provided by random selection were inter-niche, and it was speculated that the number of successful replacements for inter-niche pairings was higher than replacements by intra-niche pairings given the natural bias. The question was raised as to the effect of biased or forced pairings on the simple crowding algorithm.

This section provides a preliminary analysis as to the effect of a complete intra-niche and inter-niche pairing bias has on the simple crowding algorithm. It has been shown that the multi-niche crowding algorithm provides some bias towards selecting pairs of population members that belong to the same niche. Preliminary results of the simple crowding algorithm have shown that random upfront selection without replacement provides an implicit bias towards pairing members of different niches. This section provides a preliminary analysis of a forced 100% bias for both cases on the simple crowding algorithm and provides further insight into the impact that offspring produced from each type of pairing has on genetic search in localised generational competition based algorithms.

Two completely biased upfront selection operators were tested. The first provided a complete 100% bias towards same-niche pairings, where as the second provided a complete 100% bias towards different-niche pairings. In each case random selection was used with replacement to select the first parent of the pair, and then random selection with replacement is continued until a population member that fulfilled the bias was found as the second parent in the pair. This process was continued until the reproductive set was filled with the required number of biased pairs of parents.

Preliminary test results showed that the different-niche pairing bias achieved a state of stability (in terms of subpopulation size) in fewer generations than the same-niche pairing bias. For both test problems, stability for the same-niche bias was achieved in almost double the number of generations of the different-niche bias. The impact on convergence was further shown in the total number of replacements: the number of replacements for different-niche bias dropped below one on approximately generation 340 and 380 for F1 and F2 respectively, whereas the total number of replacements for the same-niche bias did not drop below one until the end of the run (more than generation 490) for both problems.

For the same niche pairing bias, the results showed that the number of replacement types that occurred throughout the run were relatively consistent as the number of replacements decreased over the generations. This behaviour was observed on both test problems where approximately 61% of the replacements were of offspring replacing a parent, and approximately on 33% of replacements were of an offspring replacing another population member. Replacement waste measures showed that the majority of replacement waste was from parental matching (approximately 76% of all matches) whereas other matches accounted for approximately 20% and sibling for approximately 4% of all replacement waste.

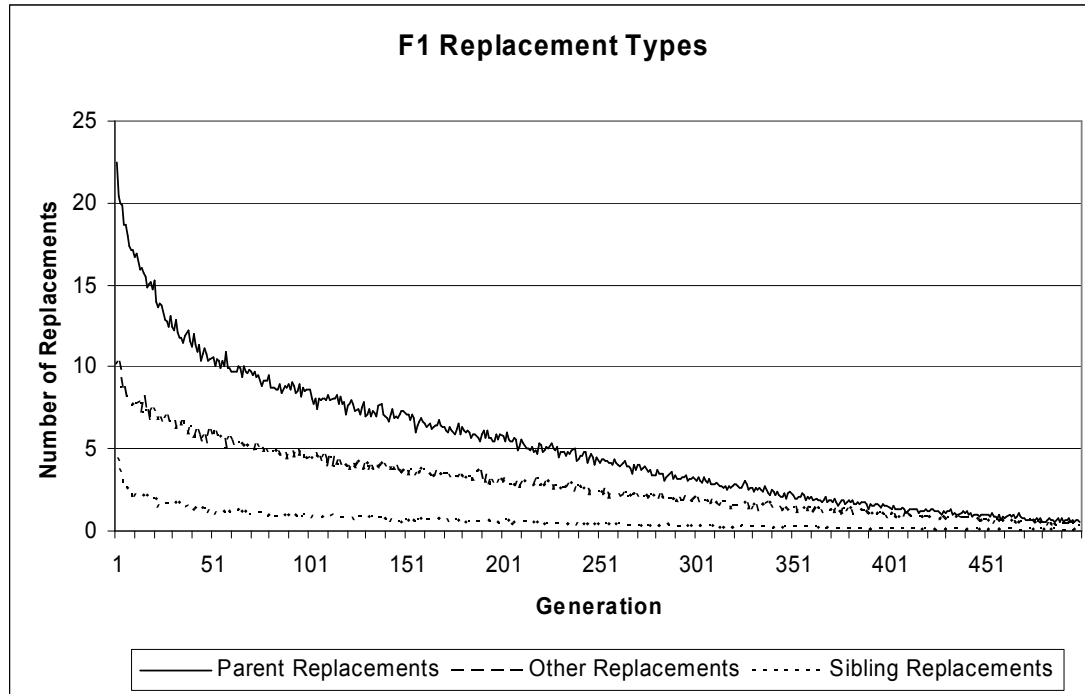


Figure 15 – Shows the replacement types for the same-niche biased pairing selection operator on the F1 test problem (mean over 100 runs)

The results for different-niche bias more closely resembled the simple crowding algorithm. The majority of actual replacements were parental replacements at approximately 47%, followed by other replacements at approximately 43% and siblings at approximately 10% of all replacements over the course of the run. Over the active generations at the opening half of the run, the parental replacements started higher than other replacements, though by approximal generation 150 on both problems, the number of replacements of each type became relatively equivalent. The number of replacements also showed similar behaviours to uniform random selection without replacement. Results showed the majority of replacement waste was of other matches at 67%, where as parental matches accounted for approximately 27% of total replacement waste.



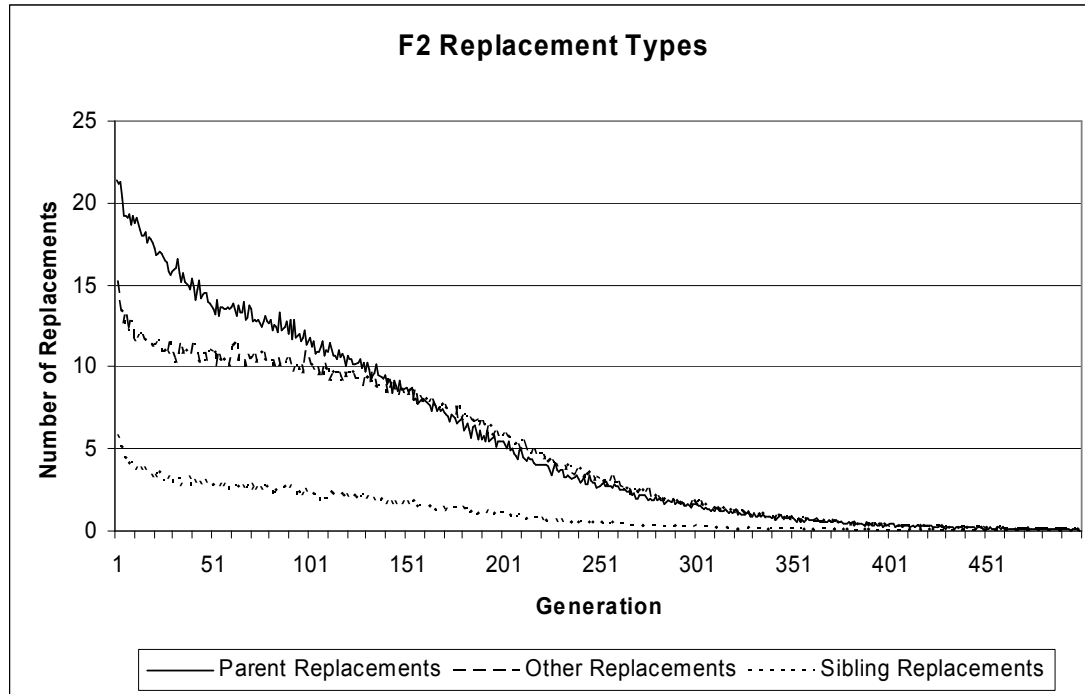


Figure 16 – Shows the replacement types for the different-niche biased pairing selection operator on the F2 test problem (mean over 100 runs)

The preliminary results showed that neither of the biased upfront selection schemes have a direct impact on subpopulation stability or the quality of subpopulations, the primary goal of the crowding principle. However, results indicated that inter-niche pairing provides faster convergence than intra-niche pairings by an approximate factor of 2. The replacement and replacement waste results show that for the same-niche pairing bias, that parental based matching was in the majority at 75%, as where in the different-niche biased pairing matching of other members was in the majority at 63%. This further leads to the hypothesis that intra-niche biased pairing causes more parental replacements, where as inter-niche biased pairing causes more non-parental replacements.

The results showed that inter-niche biased pairing provided better convergence than intra-niche biased pairing. This implies that the increase in non-parental based replacements is more useful in meeting the goals of the crowding principle of maintaining diversity and promoting useful diversity, and thus is also more useful for genetic search. It can therefore be speculated that an upfront-selection operator that provides opportunity for more inter-niche pairings than intra-niche pairings may be more beneficial to the search. In both inter and intra niche biased pairing, parents were the most common population members replaced. This indicates that parental replacement is a required element in the simple crowding algorithm. Given the increase in non-parental (other) based matches in inter-niche pairing and that parental based replacements were still in the majority, this indicates that parents are the most important type of matching for replacement. It further shows that offspring from inter-niche pairings more commonly match onto non-parental population members and that they are typically less fit in the matching, though their

contribution to the search is still more beneficial than offspring produced from inter-niche pairings.

### ***Sampling in Localised Selection***

The simple crowding algorithm provides an opportunity for an offspring to compete with the most similar population member within its niche, which also happens to be the most similar member in the complete parental population. When population sampling is used in the localised selection strategy, it is expected to introduce noise to the selection process. A heuristic that was shown to be useful to restricted tournament selection and multi-niche crowding was to use a multiple of the number of niches as the sample size. The reason was shown to be that it provided opportunity for a number of population members from each niche to be included in the sample, and thus decrease the chance of replacement error.

Replacement error was shown to be a replacement of a population member with a different niche, and was shown to be minimised in the deterministic crowding algorithm. Results as to the offspring produced from same niche and different niche pairings showed that for the majority of cases, offspring belong to one of their parents' niches. The instability of sampling shown in the A1 algorithm, and shown to occur in deterministic crowding, is therefore speculated to be caused by an effect other than replacement error. This section investigates the effect both small samples and large samples have on population stability and convergence. A new error measure is proposed and used called "exact replacement error" that measures when an offspring does not replace the most similar population member.

Two sample sizes are tested, a small sample size of 30% of the population, and a large sample size of 70% of the population. These two sample sizes represent both a small ( $6 \times P$ ) multiple and large ( $14 \times P$ ) multiple of the five optima ( $P$ ) in the F1 and F2 test problems.

The results show that subpopulation sizes for the small sample size fluctuate throughout the run. The beginnings of class dominance is apparent in all peaks, where peaks 2 and 4 show a steady rise in population size towards the last half of the run, whereas peaks 1, 3, and 5 show a steady decrease in size that shows signs of increase as the run is continued. Results for the larger sample show significantly more stability, reminiscent of the simple crowding algorithm. Variations in subpopulation size do occur throughout the run after stability is achieved after 20 generations, though the variations are quite minor. Subpopulation quality is unaffected in both runs after a perfect score of one is achieved within the first 100 generations for both sample sizes.

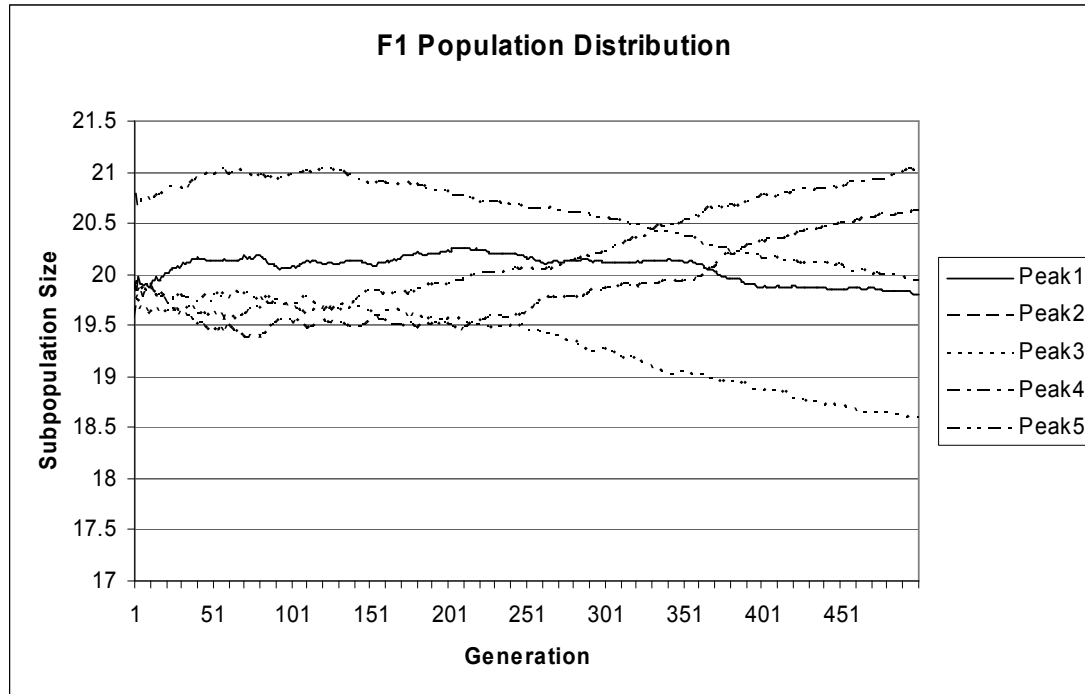


Figure 17 – Shows the subpopulation size for each of the fix peaks in the F1 test problem where a small sample size is used in the localised selection strategy (mean over 100 runs)

Speed of convergence was noticeably affected for both sample sizes. In the case of the small sample size, total replacements decreased to below 1 at approximately generation 150, where as convergence does not occur in the large sample until approximately generation 220 on both test problems. Both sample sizes differ in convergence from simple crowding which achieves total replacements below 1 at approximately generation 350. The results show that the smaller the sample size, the faster the algorithm will converge, with population wide sampling taking the longest time to converge during preliminary testing.

The results show that replacement error is higher for the smaller sample than the larger sample, but in both cases is below 1 replacement error after the first four generations for the small sample, and below 1 per generation after 2 generations for the larger sample across both test problems. This indicates that replacement error does increase as sample size decreases, though is insufficient to explain the increase in convergence in both cases and the significant decrease in subpopulation stability for the small sample size.

Exact replacement error is high for both sample sizes from the beginning of the run until convergence. Results for the small sample size show that on average the number of exact replacement errors for the first 150 generations was 70%. The results showed that over the first 220 generations that the average exact replacement error for the large sample was 63%. In both cases, the replacement errors were low, yet more than half of the intra-niche replacements that occurred were not the most similar population member. It is important to note that as the number of replacements decreased, the number of exact replacement

errors increased, inflating the mean error scores. The median exact replacement errors for the same periods were 69% and 58% for small and large sample sizes respectively.

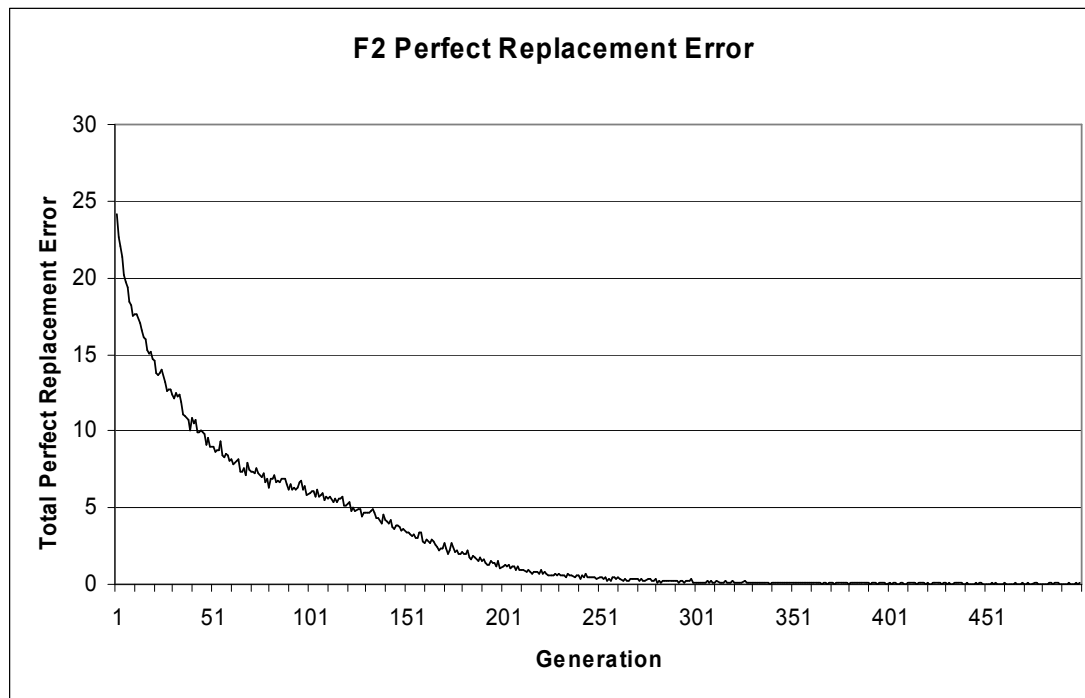


Figure 18 – Shows the exact replacement error for the F2 test problem when a large sample size is used the localised selection strategy (mean over 100 runs)

The preliminary test results of sampling in the localised selection strategy provided empirical support for the previous speculated hypothesis that practical application of crowding based algorithms provides a trade-off between convergence time and subpopulation stability. The results show that it is not replacement error as previously defined that is important, but a refined version of replacement error that measures when offspring do not replace the most similar parental population member. This was shown particularly in the small sample size results where replacement error was very low, yet subpopulations showed strong signs of instability and signs of potential subpopulation crashes.

A measure of exact replacement error showed that of the majority of replacements that occurred within the same niche, more than half of those replacements were of population members that were not the most similar to the offspring. Results showed that as the sample size is increased, the mean exact replacement error decreases, and results for simple crowding that uses population wide sampling show a zero exact replacement error. The decrease in exact replacement error with increased sample size is matched by the increase of subpopulation stability. It can therefore be speculated that the goal of localised replacement strategies is to minimise exact replacement error, in addition to minimising conventional replacement error.

The results showed that an increase in stability is associated with a decrease in exact replacement error at the cost of increased convergence time and increased replacement waste. Conversely the results showed that decreased convergence time has an association with increasing exact replacement error and decreased waste, at the cost of decreased subpopulation stability. The decrease in stability was shown to be somewhat offset by the increase in convergence, meaning that stopping the run at convergence may preserve subpopulations, though the extent of this preservation in practical application can only be speculated.

### ***Alternative Replacement Strategies***

Simple crowding uses a deterministic localised selection and replacement strategy. The replacement strategy controls the promotion of useful diversity or fitter members of the population. The canonical genetic algorithm, crowding factor model and multi-niche crowding use a direct replacement strategy that does not take fitness into account, though fitness is used in the localised selection strategy in multi-niche crowding. Restricted tournament selection, deterministic crowding and the simple crowding algorithm use a rank based fitness tournament. Probabilistic crowding uses a fitness proportionate stochastic selection scheme. This section evaluates the effect different deterministic, probabilist and stochastic replacement strategies have on the simple crowding algorithm.

The effect three additional replacement strategies on the simple crowding model were evaluated. The replacement strategies were as follows:

1. Probabilistic – As was shown in the probabilistic crowding algorithm
2. Stochastic – Randomly perform replacements
3. Deterministic – Perform replacements directly without fitness consideration

The replacement strategy is the component of the generalised crowding model that facilitates the promotion of useful diversity. Both the probabilistic and random replacements are stochastic in nature, though probabilistic provides a fitness bias in proportion to relative fitness in the pairing. Direct replacement provides no replacement waste, forcing each offspring to be kept each generation.

The results show that probabilistic replacement provides the most stable subpopulation sizes of the three approaches, resembling the rank based tournament results, though showing minor variance throughout the run. Random and direct replacement provide similar proportions that show properties of stability in terms of variance from the mean population size, though shown increasing generational fluctuations. Direct replacement shows the largest functions, though over the 500 generations, none of the strategies show indications of class dominance or signs of future subpopulation crash.

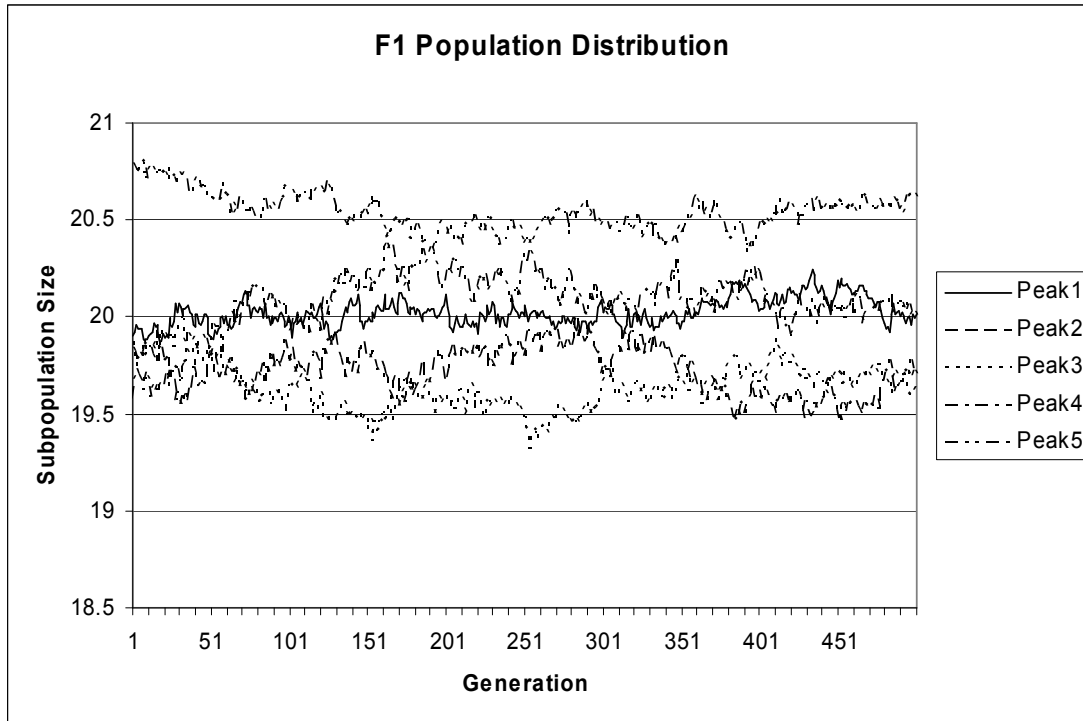


Figure 19 – Shows the subpopulation sizes of the five peaks in the F1 test problem when using the direct replacement strategy (mean over 100 runs)

The three strategies showed relative subpopulation stability with fluctuations, the results for the subpopulation quality (maximum peak ration) reflect similar behaviour. Probabilistic replacement provides the more stable results compared to stochastic and direct replacement that both showed a decline in subpopulation quality over approximately the last 200 generations of the run. All three techniques maintained maximum peak ratios above 97% for the duration after a period of initial convergence in the first approximate 20 generations of the run.

The replacement results were of interest showing that parents were replaced in the majority for all three replacement strategies. The results for direct replacement showed that sibling type replacements occurred more often than other replacements with a total of 21% and 17% respectively for all replacements. The number of replacements and replacement waste was high for both probabilistic and stochastic replacement, showing 48%/52% and 50%/50% average ratios per generation respectively. Direct replacement provided 100%/0% replacements and replacement waste as naturally implied by the strategy.

When fitness is not the primary motivation for replacement competition, the goal of the simple crowding algorithm is adjusted from that of the crowding principle and therefore the generalised crowding model. Stochastic and fitness-neutral replacement strategies were shown to maintain relative population stability though with seemingly random fluctuations. Given the simplicity of the problem domains, subpopulation quality was promoted implicitly, though not explicitly as is required by the crowding principle. The

lack of pressure to maintain useful diversity showed a decrease in subpopulation quality that towards the end of the run showed signs of degrading further for the random and direct replacement strategies.

The results show that an exact localised selection policy is only capable of maintaining general diversity. Useful diversity must be maintained by other means, the maintenance of which provides further increased stability to subpopulation sizes. It is speculated that if diversity without the promotion of useful diversity is the desired feature, then it can be provided by exact localised selection. It can be further speculated that it may be possible to achieve crowding like results by reintroducing fitness biased upfront selection and using a fitness-neutral replacement strategy, though still employing exact localised selection for replacement. This was not explicitly tested, though may provide an alternative means of competition to the generalised crowding model more familiar to the canonical genetic algorithm.

## ***Discussion***

Simple crowding provides a base of test and comparison for variations, approximations and augmentations of localised generational competition. This section provided preliminary investigation into some approximations, augmentations and variations employed by crowding based niching genetic algorithms and evaluate their effect on the base simple crowding algorithm. The standalone effect of the augmentations to the crowding principle had not been investigated in as much depth using the selected analysis measures in previous work and their effect on the underlying model has mainly been predicted by analytical models or assumed from global algorithm performance. This section provided a basis for comparison and preliminary empirical observations as to the effect of the major features employed in crowding based algorithms.

Sampling in the upfront selection was shown to be used in multi-niche crowding to bias pairing of population members in the same niche. It was shown that the intent was to limit the generation of a hybrid offspring with low fitness referred to as lethals, as well as to improve intra-niche convergence. Test results showed that a bias towards inter-niche pairing decreased convergence time compared to a full bias towards inter-niche pairing on the simple crowding algorithm. The results indicate that in simple crowding, intra-niche pairings increase convergence time, having the opposite of the desired effect intended in multi-niche crowding. Convergence here is measured by the number of replacements that occur, implying a state of minimal change or stability which is a more meaningful definition for crowding based algorithms.

A bias in pairing during upfront selection was shown to have no measurable impact on maintenance of diversity or the maintenance of useful diversity within each subpopulation (quality). This implies that a selection technique that is biased towards inter-niche pairings, (as the uniform selection scheme is naturally), is more useful to the search given the increase in convergence without an observed associated cost to stability or quality. The goal of reducing low fitness hybrids by biasing towards same-niche pairings was shown to be achieved. The results showed that by increasing the inter-niche hybrids produced in upfront selection did increase the number of low fitness hybrids

(lethals). It also increased the production of high fitness hybrids, which were shown to be more useful to the search as evidenced by the decrease in time until convergence.

The restricted tournament selection (RTS) was shown to provide sampling in the localised selection strategy. Multi-niche crowding was also shown to provide sampling though of a more elaborate nature than RTS. In both cases, sampling was shown to be employed to approximate intra-niche matching, with the goal of minimising replacement error. Deterministic crowding was shown to use a localised selection strategy limited to selecting parents, with the goal of minimising replacement error. Results using both small and large sample sizes showed that the goal of minimising replacement error through approximation is achieved successfully, though low replacement error alone was not indicative of subpopulation stability.

Localised generational competition is the definition of how the crowding principle is employed in genetic search and an integral part of that definition is the similarity based selection in the localised selection strategy. Results showed that the degree of localisation has a measurable effect on the maintenance of diversity, specifically the stability of subpopulation sizes. The results showed that the more accurate the localisation (larger the sample size), the more stability was exhibited in subpopulation size. In sampling tests, the goal of achieving intra-niche replacements was achieved with generational replacement error below one for all but the opening generations. Results showed that the goal of same niche replacements is too broad, and rather the goal of same locality replacements within the niche is more accurate, where exact replacement is replacement of the most similar parental population member to the offspring.

The simple crowding algorithm has been shown to achieve exact matches in the localisation selection strategy and thus has exact replacement error of zero. Exact replacement error is a measure of the number of replacements per generation where an offspring does not replace the most similar parental generation population member. Results showed that during sampling that the number of exact replacement errors accounted for more than half of the replacements that occurred. The larger sample showed lower average exact replacement errors than the smaller sampling strategy.

In summary, the results indicate that localisation selection strategies should approximate exact matching, rather than same-niche matching. An approximation scheme should use the principle that each population member has a distance from the exact similarity match, and the strategy should seek to minimise that distance. The closer the approximation to exact similarity match, the better the subpopulation size stability.

It is interesting to note that replacement errors can still be recorded in cases where exact replacement error is zero. The reason for this is the difference in what is being measured. Replacement error is concerned with the broader issue of same-niche replacements, where as exact replacement is concerned with the binary problem of exact similarity replacement or not. It is speculated that when sampling methods are employed in crowding based algorithms; the exact replacement error analysis measure could be used. The magnitude of this score would provide an indication of the accuracy of the localised



selection strategy used where no a priori information about the number or location of optimum in the problem domain is required.

Localised competition requires means for competing for a position in the fixed sized population structure. The simple crowding algorithm uses deterministic rank based fitness to decide the outcome of placement in the same manner as restricted tournament selection and deterministic crowding. Other decision processes were shown to be used in crowding based algorithms such as fitness proportional selection and direct replacement. The results of testing alternative fitness biased and fitness neutral replacement strategies revealed that localised replacement alone provides capability of maintaining stable subpopulations (albeit with fluctuations around the stable point) throughout an extended run duration. The associated cost was shown to be the lack of promotion of useful diversity providing a drop in subpopulation quality, particularly noticeable for strategies that performed replacement without regard for fitness.

This section provided a preliminary investigation into the effects of alternative components of the generalised crowding model on the simple crowding algorithm, providing further insight into the resulting effects of the components on the crowding principle and further insight into the nature of the simple crowding. Upfront selection was shown to control the direction of the search and was shown to benefit from pairing diverse population members. Localised selection was shown to have a majority influence over subpopulation stability, and was shown to benefit from approximations of exact similarity selection, rather than approximations of same niche selection. Finally the replacement strategy was shown to control the promotion of useful diversity, and it was speculated that the promotion of diversity could also be provided by a fitness biased upfront selection instead of at replacement time. This section provided insight into the components effects on simple crowding in isolation of each other. Practical implementation of a crowding based algorithm will use combinations of components and the components are expected to influence the function of each other and the search. The insight provided in this section of the components isolated behaviour is expected to be of some use in predicting and evaluating their combined effects in practical application.

## Simple Crowding Compared

This section provides a further preliminary investigation onto crowding based niching genetic algorithms. A set of common crowding based algorithms are tested and analysed the results of which are compared to the simple crowding algorithm and the expectations provided by previous exploratory work. Previous defined speculation and hypothesis are tested and provided with additional support from a larger set of function optimisation test problems. The analysis of results confirms the convergence-stability trade-off observed in localised generational competition. Further empirical evidence supports the need for a balance between other and parent type replacements in the localised selection strategy, and the goal of exact replacement error minimisation is shown to be useful, and is extended for practical use in the development of future approximations of exact similarity matching.

Crowding based algorithms were selected that provided an implementation of the components investigated in previous work. The tested algorithms were limited to restricted tournament selection (RTS), deterministic crowding (DC), probabilistic crowding (ProbC) and simple crowding (SimpC). All algorithms were tested with full crossover and no mutation in an effort to measure and accurately evaluate the algorithms underlying behaviour and performance characteristics. The selected algorithms were tested on all of the selected function optimisation problems (F1-F5). The focus of the testing was on the algorithms ability to maintain stable subpopulations and to maintain stability in terms of the quality of result in each subpopulation. To this end, the localised matching, replacement and replacement error behaviours were observed to both confirm previous speculation and hypothesis, and gain additional insight into the algorithms respective components effect on genetic search.

### *Maintenance of Diversity*

The primary goal of the crowding principle, and hence of an algorithm that exploits this principle is the maintenance of general population diversity. This can be measured by observing the change in subpopulation size at each desired optimum over the run duration. The table below provides a summary of the mean squared deviation from subpopulation size summed over all desired peaks for each of the selected test problems. The scores represent a summary of the total change in subpopulation size and hence subpopulation size stability over the course of the run.

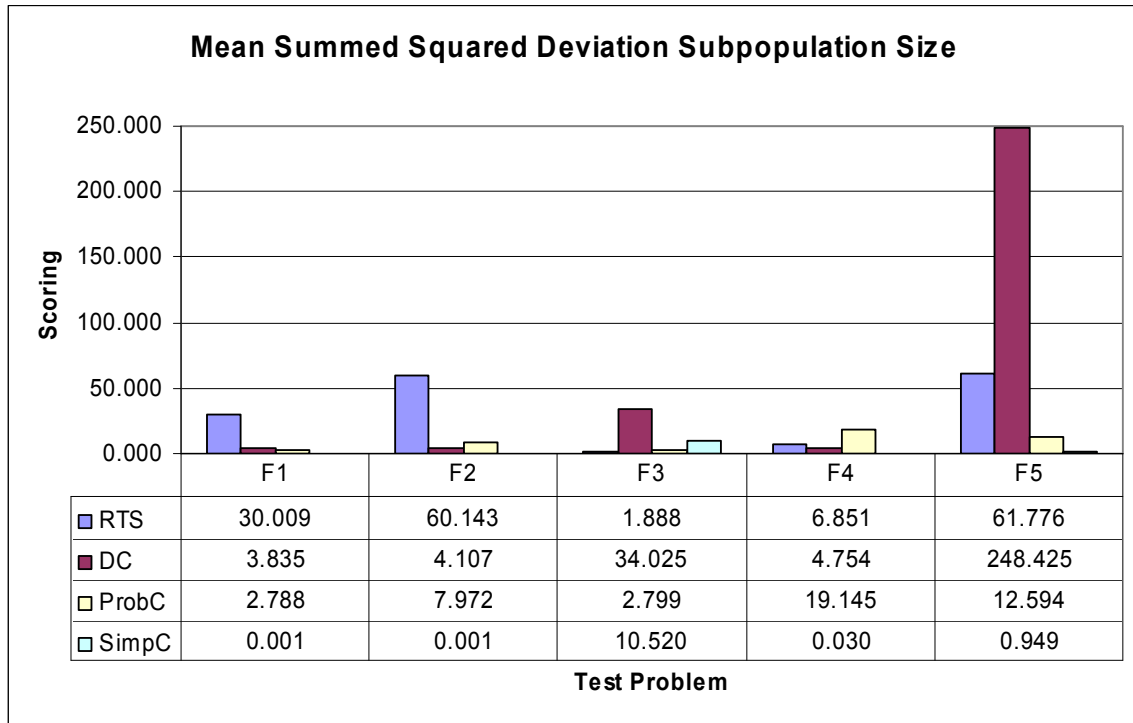


Figure 20 - Shows the mean summed squared deviation from mean subpopulation size for tested crowding algorithms on the selected test problems

The results clearly show that simple crowding had the least deviation in subpopulation size on all but the F3 test problem. The scores show that RTS was unable to maintain stable subpopulations on F1, F2 and F5, where as deterministic crowding showed significant deviation on F5, and to a lesser degree F3. The summary scores show that simple crowding achieved significantly more stable subpopulations for the majority of the selected test problems.

Previous research by Mahfoud [15] led to the expectation that class dominance should be noticeable during algorithm testing. Actual observations of subpopulation size over the course of the run on selected test problems confirmed this expectation, showing strong signs of rapid growth in the opening stages of the search before convergence. Results showed extreme growth on F5 that lead to the crash of the subpopulation at peak3 at approximately generation 150. Previous test results on sampling in the localised selection strategy provided the expectation that the RTS algorithm should show signs of rapid growth and subpopulation instability given the expected increased exact replacement error. The results showed RTS exhibited peak instability on F2, leading to peak6 subpopulation crash at approximately generation 365. Results showed signs of potential crash on F1 if the run duration were further extended. Results for F3, F4, and F5 showed signs of strong class dominance though less severe than the results for DC.

Investigatory results from testing with a probabilistic replacement strategy lead to the expectation that probabilistic crowding would maintain approximately stable subpopulations though with large almost random fluctuations in size. These expectations

were confirmed, though results for F1 and F2 showed signs of early peak dominance expected to be an artefact of the parent based replacement policy employed. Results further confirm that without a strong deterministic replacement strategy, a level of subpopulation stability can be achieved by the localised selection strategy used.

Simple crowding achieved excellent subpopulation stability results that were held consistently from convergence until the end of the run. As expected, simple crowding did not show signs of class dominance in the opening stages of the run on the majority of the test problems. All results showed minor positive and negative growth fluctuations in the opening stages of the run as was observed previously on F1 and F2. The results showed that simple crowding is susceptible to class dominance on F3 even with an exact localised selection strategy. On the F3 problem, simple crowding showed a similar dominance effect as was observed by deterministic crowding, though the effect was shown to be more gradual. This provides evidence that as Mahfoud showed class dominance is an effect caused by the crossover operator. The results further indicate that although class dominance is possible, it is only exhibited by simple crowding on some problems, and less severely than by RTS and DC.

### ***Promotion of Useful Diversity***

The promotion of useful diversity is the secondary goal of genetic search algorithms that exploit the crowding principle. Useful diversity can practically be measured as the quality of result in subpopulations, the stability and magnitude of which provide an indication as to the algorithms ability to meet this secondary goal. The figure below provides the mean maximum peak ratio for the test algorithms over the selected problems.

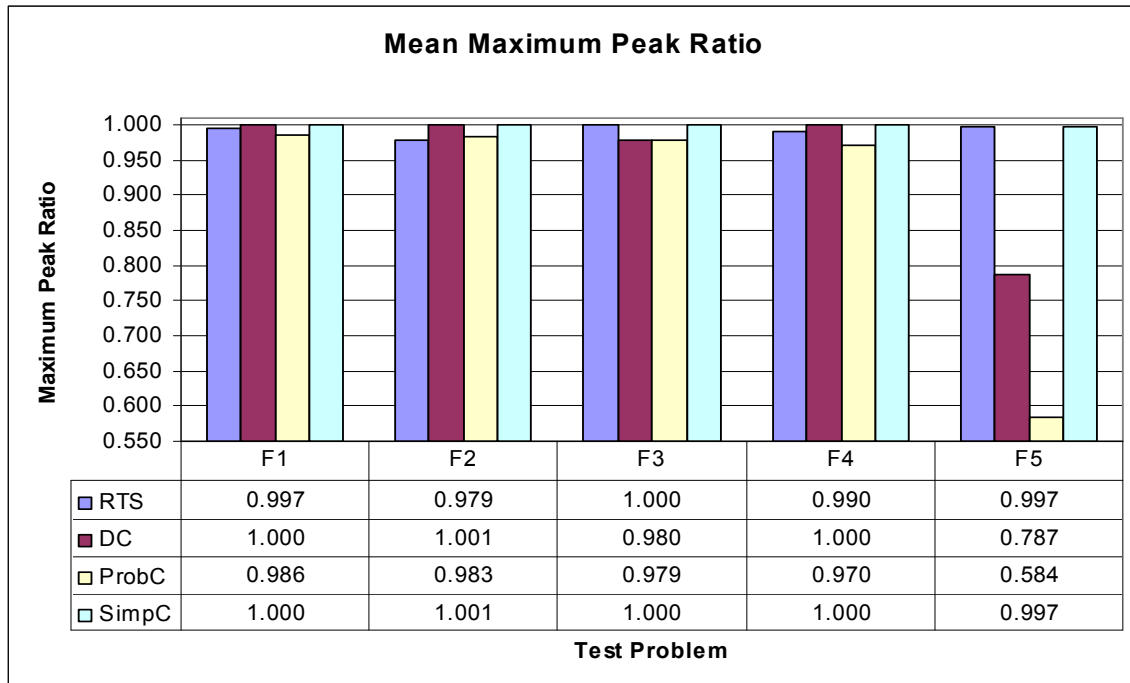


Figure 21 – Shows the mean maximum peak ratio for the test algorithms over the selected problem domains.

The table shows that simple crowding achieved the highest or equal highest mean maximum peak ratio for all test problems. This indicates that the algorithm was able to both locate the local optimum of each niche and maintain it with stability (less deviation) over the duration of the run on all test problems. The results show that deterministic crowding achieved stability in the quality of results which confirm the stability observed in subpopulation size after convergence observed on the majority of test algorithms, F3 and F5 being the exceptions as the scores show.

Given the stability of the subpopulations observed for the simple crowding algorithm, quality of result maintained by each subpopulation was expected to be equally as stable, as was confirmed by the test results. Class dominance expected and observed in RTS and DC provided the cause for both algorithms to show stability concerns in the maintenance of quality of result. RTS showed a stepwise decrease in MPR over the course of the run after convergence on F1, F2 and F4. As mentioned, DC showed problems on F3 and F5, which are explained by the decrease in MPR on both problems that match the observed decrease in subpopulation sizes. Class dominance is confirmed as the factor that leads to both subpopulation instability and decrease in quality of result, however slight.

### ***Time until Convergence***

The results of previous investigations into simple crowding and the localised selection strategy identified a trade-off in the application crowding between convergence time and subpopulation stability. The number of generations until convergence for the test algorithms provides further confirmation for the hypothesis. The results showed that RTS was by far the fastest algorithm to reach a state of convergence on the selected test problems, yet showed rapid growth in subpopulation size before convergence, and continued class dominance after convergence on the majority of test problems. The figure below provides a summary of convergence times for the test algorithms in terms of the number of generations before the number of replacements dropped below one.

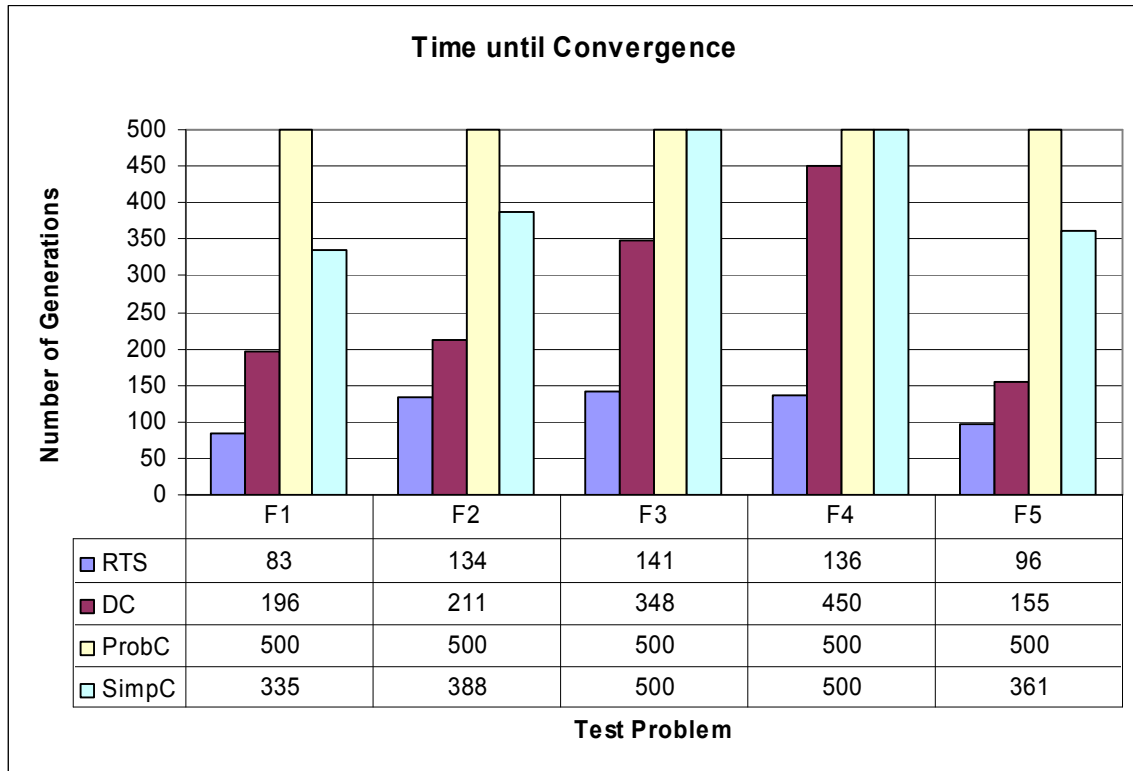


Figure 22 – Provides a summary of the number of generations before replacements dropped below one for the crowding algorithms on the selected test problem domains.

Deterministic crowding was also shown to achieve good relative convergence times compared to the other tested algorithms, though showed less stability in mean maximum peak ratio and subpopulation size than the simple crowding algorithm. Simple crowding showed longer convergence times than DC and RTS, yet more stability in terms of maintenance of general diversity and useful diversity. Given the stochastic component of replacements in probabilistic crowding, the algorithm shows no signs of convergence. The trade-off between convergence and subpopulation stability is a heuristic that can be used in practical application of crowding algorithms, where a suitable balance can be sought given problem complexity, desired quality of result and available algorithm running time.

### ***Localisation Matching Behaviour***

The matching behaviour provided by the localised selection scheme provides insight into the manner in which the crowding algorithm searches the domain, and how it may be better tuned for a given type domain. Previous investigation into the matching behaviour of simple crowding revealed that parental based matching was the main type of matching that occurred, though runtime observations showed that parental matching was more common in the opening stages of the run when the most replacements occurred, and that other type matches were more common in the final stages of the run after convergence. The results for simple crowding on the selected test problems exhibited similar behaviour.

Both the deterministic crowding and simple crowding algorithms only perform parental based matching given the imposed limit in the localised selection strategy. The RTS algorithm showed less intuitive results, revealing that on all test problems that other based matches occurred the majority of the time at approximately  $\geq 92\%$  of all matches, followed by approximately 6% of all matching being parental. Although parental based matches were the more dominant form in simple crowding, given the transitional behaviour observed, other type matches typically accounted for 40%-50% of all matches that occurred.

The matching behaviour in localised selection for crowding algorithms has been shown to be a critical component of the realisation of the crowding principle in genetic search. The results show that a parent-only matching policy used in deterministic crowding increases convergence and adversely effects subpopulation stability and quality. The results also show that a sampling strategy used in restricted tournament selection provides an other-type majority matching policy that also leads to increased convergence and stability concerns. The simple crowding algorithms shows that an overall 40%/50% or 50%/50% matching policy for other and parent type matches is desirable to achieve strong subpopulation size and quality stability. The results further show that a transition of the matching policy may be desirable to meet the goals of the crowding principle in genetic search. This is where parental based matches are more common before convergence, and other type matches are in the more common as the population converges. The results show that a majority in either direction does not provided as stable subpopulation sizes over extended duration.

### ***Replacements Behaviour***

The replacements that occur in the algorithm are firstly a result of the localised selection strategy, and ultimately controlled by the replacement strategy employed by the algorithm. The replacements that do occur control the maintenance of diversity and the promotion of useful diversity, thus the general focus on the restrictive replacement nature of crowding based algorithms. Sampling in the localised selection strategy decreases the chance of an exact match and was shown to promote other type matching in all test problems. The matching policy observed was shown to cause the main type of replacements that occurred on the test problems to be of non-parental and non-sibling members of the population with a ratio of approximately 79%-84% of all replacements. Deterministic crowding and probabilistic crowding replaced only parent members of the population given the employed localised selection strategy.

Previous investigation into simple crowding showed that parental replacements were just barely the more common form of replacements, which observations over the selected test problems further confirmed accounting for approximately 51%-53% of replacements of F1, F2, F3 and F5, and 61% of all replacements on F4. On all but the F4 test problem, the observed replacement behaviour by simple crowding showed approximately equal other and parental replacements from the point of convergence on F1 and F2, or the equal ratio of each type from the beginning of the run on the remaining problems.

This behaviour was not observed in RTS, DC or probabilistic crowding and is speculated as being a large part of the explanation for the algorithms stability concerns compared to simple crowding. The replacement behaviour of simple crowding was shown to require a balance of other and parental replacements over the duration of the run on all problems. This balance is supported by the transitional matching behaviour of the exact similarity localised selection strategy used. The RTS algorithm was shown to use a sampling approximation which lead to a majority of other matching and other replacements, where as DC and probabilistic crowding provided an approximation by only matching and replacing parents. When a greedy deterministic replacement strategy is used, both the RTS and DC approximated localised selection strategies were shown to exhibit strong signs of peak dominance, leading to subpopulation crash on at last one problem each. This provides further evidence to speculate that to maintain stability of subpopulation size and quality, a balance of both type of replacements is required over the duration of the run.

### ***Replacement Error Behaviour***

Previous investigation into localised selection identified the hypothesis that a minimisation of replacement error may be sufficient to approximate the crowding principle, but it is insufficient to adequately meet the goals of the crowding principle. Previous work revealed a new measure of error called the exact replacement error which implies an approximation of intra-niche replacement should instead be an approximation of replacement in specific locality within a niche, the exact version of which is exact similarity matching. Exact replacement error is a count of the number of non-exact replacements that occur over the duration of a run.

Previous work on sampling in localised selection showed high exact replacement errors. These high errors were expected in the approximation techniques used by RTS and DC. The RTS algorithm showed high measures of exact replacement error which accounted for approximately 80%-85% of all replacements on the selected test problems, and was as high as 91% on F5. Deterministic crowding showed approximately 50% of all replacement were exact replacement errors on F1, F2 and F3, 39% on F4 and 63% on F5. These results provide further indication that a parent only selection and replacement policy approximates only half of the desired replacements required to achieve the stability shown by the simple crowding algorithm. Conventional replacement error was low on DC accounting for approximately 4% of all replacements, where as with the RTS algorithm it accounted for approximately 8%-10% of all replacements.

These results lead to the speculation that an increased exact replacement error accentuates the class dominance effect which can be minimised with a reduction in exact replacement error. It further provides evidence that minimisation of conventional replacement error is too broad a goal to achieve maintenance of diversity and promotion of useful diversity on the selected test problems.

### ***Discussion***

This section provided test results and analysis of niching genetic algorithms that use components whose effects were previously investigated in isolation on the simple



crowding algorithms. The analysis of the results provided further supporting evidence and revealed new insight into previously presented hypotheses regarding the convergence-stability trade-off, replacement behaviour required to achieve stability, and the reduction of exact replacement error in localised selection approximation.

The RTS and DC algorithms showed increased convergence over the simple crowding algorithm, and both algorithms showed increased class dominance effects and decreased subpopulation stability. Simple crowding showed overall superior stability of subpopulation size and stability quality of subpopulation result, though increased convergence times compared to RTS and DC. Similar results were observed when sampling was used in localised selection, and collectively the results show that a crowding algorithm with a deterministic rank based replacement strategy has a user configurable trade-off between decreased stability through approximation which leads to decrease convergence time.

The results further confirmed previous speculation that a balance close to 40%/50% to 50%/50% of other and parent type replacements over the course of the run provides the stability observed in simple crowding. The skew provided in RTS though sampling and in DC though approximation were shown to be a large contributing factor to the algorithms growth before convergence and stability concerns. The results further confirmed that the replacement that occurred in simple crowding were caused by the replacement opportunities provided by the localised matching strategy, that showed a transitional behaviour on all problems from parental-majority matches before convergence to other-majority matches after convergence.

Finally the results confirmed that a minimisation of replacement error is insufficient to facilitate subpopulation stability. The sampling provided by RTS and the approximation provided by DC in the localised selection strategy provided high levels of exact replacement error. The highest levels were observed for RTS which also showed the least stability in terms of subpopulation size. Deterministic crowding showed approximately 50% exact replacement error and more stability than RTS, providing further supporting evidence that a minimisation of exact replacement error is desired for approximate localised selection strategies. The count of exact replacement errors is insufficient to provide indication as to the degree by which an approximation scheme is successful in this goal. Therefore it is reasonable to speculate that a magnitude of replacement distance from the best match is proposed the sum or mean may be useful. This analysis measure is expected to provide a more meaningful scoring as to the degree an approximation technique is able to perform matches and replacements within the desired locality within a niche, the exact implementation of which is represented by exact similarity replacement.

## Conclusion

This section provides a succinct summary of the main findings and observations from the empirical investigations performed in this work. Future potential research directions are discussed that provide extensions to the exploratory work provided. Alternative perspectives of crowding in the context of genetic search are suggested that may provide additional insight into crowding based niching genetic algorithms and potentially frameworks for proposing and testing new crowding inspired algorithms. The crowding principle is seen as a simple yet powerful tool for application in search domains that contain multiple approximate solutions. Further, it is speculated in this section that this simple approach may be applicable to other classes of problems in addition to search such as data reduction and compressed search space representation techniques.

## Summary

Localised generational competition is an implementation of the crowding principle in genetic search that has been shown in previous work to be useful for locating multiple approximate solutions in multimodal problem domains. Localised generational competition is one implementation of the crowding principle and may not be the only or best implementation of the principle in genetic search. This work did not address this question; rather it has addressed the question as to the nature of localised competition in genetic search. The crowding principle has been adopted and exploited in multimodal search by numerous algorithms as shown, though the question as to the actual behaviour and desired goals of the underlying principle in search had not being explicitly addressed.

This investigation analysed the design and implementation of existing crowding based genetic algorithms. A simple abstract model called the generalised crowding model was presented that represented quintessential crowding implemented in all of the reviewed crowding based niching algorithms. It has been shown that the general goal of crowding is to maintain general diversity and promote useful diversity. The general method of applying crowding to genetic search was shown to be implemented as localised generational competition which is a more succinct expression of how crowding is implemented in genetic search than restrictive replacement.

The main findings and useful observations from the preliminary empirical results using the proposed conceptual model and analysis measures were as follows:

1. Upfront Selection  
Uniform random pairing of population members under population wide selection provides a natural bias towards different-niche pairings of parents. A selection operator that forces a different-niche pairing policy provides similar results to a forced bias towards same-niche parental pairing though was shown to converge faster. This implies that different-niche pairing in upfront selection may be more useful to search and questions the need for a same-niche pairing bias as in the multi-niche crowding algorithm.
2. Recombination

One-point crossover provides a natural bias towards producing offspring that belong to either one or both parents' niche. This questions the actual empirical benefit of a specialised recombination operator as is used in the multi-niche crowding algorithm.

3. Localised Selection

Preliminary results showed an interesting transitional localised matching behaviour between parent and non-parent type matching over the course of the run. This is more complex than localised matching and may be beneficial or required for to promote niche-size stability and or quality under full population-wide selection and crossover.

4. Replacement Strategy

Preliminary test results showed that a trend towards equal parent and other-type replacements over the course of the run. This more detailed understanding of actual replacements questions the desire to only replace parent population members as in deterministic crowding. Further this questions whether the accentuated class dominance effects seen in the deterministic crowding are caused by the limited matching and replacement policy.

Preliminary test results lead to the proposal and further successful testing of a crowding algorithm stability verses convergence trade-off. The trade-off may prove to be a useful heuristic for the practical application of crowding based algorithms in multimodal function optimisation as well as in crowding algorithm design and analysis. The trade-off indicates that stability of niche size and quality can be sacrificed for decreased convergence time by being less precise or restrictive with the localised selection and replacement strategies. Relaxing the replacement mechanism leads to reduced replacement waste meaning more of the samples generated are kept and incorporated into the population which has the effect of prematurely converging the population.

## ***Future Research***

This section provides a summary of some simple potential improvements and augmentations to the simple crowding algorithm that may result in an improved search algorithm and/or reveal more regarding the founding crowding principle. An inverted crowding algorithm is proposed that better relates crowding based niching genetic algorithms to the canonical genetic algorithm by treating restrictive replacement as a different type of inter-generational upfront selection operator. Finally a crowding map is proposed that may facilitate the application of the crowding principle to other forms of search. Further it may provide a means of applying the underlying principle to different classes of problems such as data reduction or a compressed sample persistence system.

## **Improvements to Simple Crowding**

Proposed in this section are a number of augmentations and research directions not pursued in regard to the simple crowding algorithm. These include both changes to the algorithm expected to provide advantage of some kind, as well as investigatory directions which may reveal more regarding the application of the crowding principle in genetic search.

### Alternative Replacement Mechanism

The crowding based algorithms investigated were implemented in such a way that replacements occurred sequentially in the same population, providing the opportunity for pairs of offspring potentially to replace each other and members of the parental generation being selected for replacement more than once. An alternative approach would be to treat the replacement process as a population selection scheme each generation. This approach would require  $N$  localised selections where an offspring would be matched to a member of the parental population, a tournament held, and the winner of which is permitted entry to the next generation.

The difference with this approach would be that instead of just discarding an offspring that lost in the fitness tournament, parents too would be discarded and not placed back into the parental set. Further the approach would also prevent selection of offspring during the localised selection process.

The following figure demonstrates the alternative selective approach.

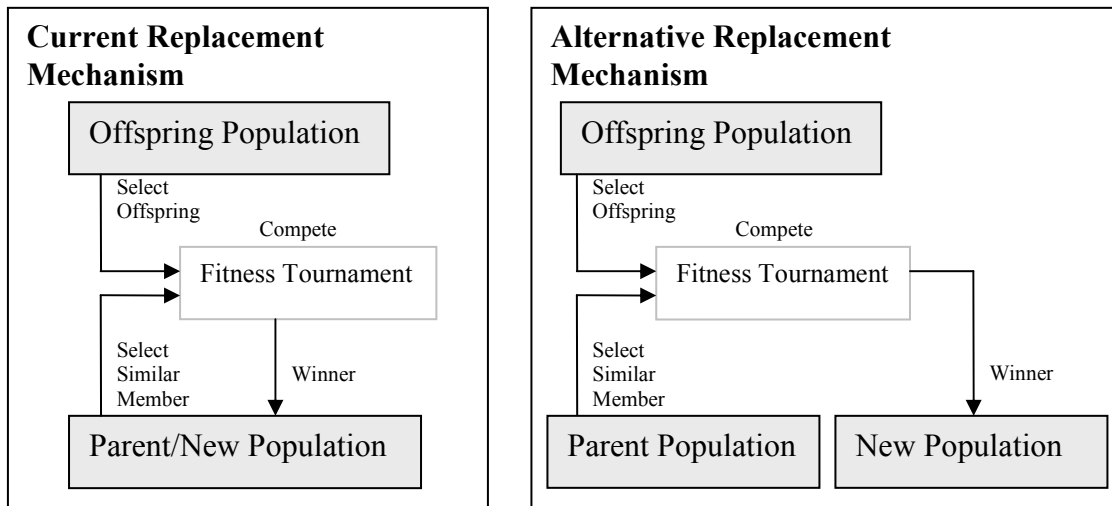


Figure 23 - Demonstrates the current and an alternative replacement mechanism for crowding based algorithms

It is expected that this alternative replacement mechanism may make better use of the offspring produced by completely eliminating same-generation replacement. Given that the parental set would reduce in size at the same rate as the offspring population, this is expected to provide a decrease in computational complexity in locating the most similar member of the parental set. It is also expected that the decrease in parental population size will have an effect on the offspring that actually make it into the next generation. It is unclear what effect on niche size stability and quality this change would have, though the approach may show similar effects as sampling in the localised selection strategy. These effects may include decreased stability or increased disuse of offspring or parental solutions that may in fact be valuable to search.

### **Fitness Bias in Upfront Selection**

Preliminary test results showed that exact localised selection in the simple crowding algorithm is capable of providing limited stability of subpopulation size (maintenance of general diversity). It may be interesting to investigate this effect further. This work could involve an investigation into the addition of exact localised selection to the canonical genetic algorithm both with and without a fitness-biased replacement strategy. The element of interest in this research direction would be making effective use of the fitness proportionate selection provided in the canonical genetic algorithm, exact similarity matching and some form of restrictive replacement strategy.

Mahfoud has provided some work into the effects of exact localised selection with fitness proportionate upfront selection and direct replacement. Preliminary work showed concerns with selective pressure (the promotion of useful diversity). It is expected that further research into finding a means of incorporating exact localised selection into the canonical genetic algorithm with a replacement strategy capable of promoting useful diversity may reveal a new hybrid of crowding based niching in genetic search.

### **Domain Dependence and Population Size**

Little work has been completed in the area of the crowding approaches dependence on the problem domain. An investigation into crowding algorithms reliance on difference in fitness between areas of interest may reveal more regarding the nature of the technique. Work in this research direction would address questions such as; how does the simple crowding algorithm select an area of interest to pursue over other areas, and what is the minimum initial population size and difference in fitness between the areas that influence the algorithms focus during search. These fundamental questions are expected to reuse work by Mahfoud into class dominance and provide more of a theoretical framework for analysing crowding based algorithms in general.

Further work into the reliance on the recombination operator is related to this area of research. Investigation into the location within the search space where offspring are placed given parent pairings from the same and differing niches will be useful in this regard. Preliminary results in this work revealed that the majority of offspring produced by the one-point crossover operator belong to one of their parents' niche, but no work was completed into the relation between produced offspring and parents in the search space. A detailed investigation into the effects and reliance of the recombination operator will likely provide a clearer indication as to what is required from a recombination operator in a crowding algorithm for application to practical problem domains.

### **Analytical Modelling**

This work has focused on an empirical observation to formulate behavioural hypothesis and speculation. What is required to both provide additional insight and strengthen hypothesis and speculation is the development of some analytical models of the crowding principle and the simple crowding algorithm.

The analysis of simple models of selection and sample generation are expected to provide further insight into how crowding algorithms direct the search towards areas of interest.

Analysis of analytical models of the localised selection and replacement mechanisms are expected to provide better understanding of how exact matching leads to increased levels of stability in terms of niche size and quality. It is expected that through modelling the crowding principle and its embodiment in the simple crowding algorithm that an explanation can be found as to how and why the algorithm exhibits interesting transitional matching behaviour in application. Further it is expected to provide insight and further explanation as to the minimum subpopulation size and distribution required in each peaks basin of attraction for the model to maintain and explore the peak throughout the duration of the search.

Previous work was shown to include some simple analytical modelling of class dominance and schema dominance in localised selection sampling for crowding based algorithms. It is expected that this previous work can be adapted or used as a starting point for a more detailed analysis of some of the suggested areas of focus.

### **Inverted Crowding**

This work proposed a crowding template and framework called the generalised crowding model which allowed the functional elements of existing crowding based algorithms to be considered as discrete logical units or components. When crowding algorithms, including the simple crowding algorithm, are considered in terms of this template it is possible to liken the algorithm to the canonical genetic algorithm though with rearranged priorities. By inverting the template for crowding algorithms, it is possible to consider crowding a canonical genetic algorithm with specialised genetic operators. The critical element to this transformation is in treating the crowding concept of restrictive replacement as in fact an upfront selection strategy in the canonical genetic algorithm. This is made easier by considering that in the simple crowding algorithm the entire population participated in reproduction via a uniformly random fitness-neutral selection strategy. This is a similar process as occurs within a selected reproductive set in the canonical genetic algorithm, and in this way the replacement strategy can be considered a means of preparing a reproductive set which then recombines to produce the next generation.

An important point is that the offspring do not replace the parental population in this version of the canonical genetic algorithm. Instead a “new” selection operator considers both parental and offspring populations when preparing the reproductive set. Moreover a localised competition is held to determine the contents of the reproductive set before the recombination means of generating the next generation of offspring.

The following figure demonstrates the transition of the simple crowding algorithm onto a canonical genetic algorithm template.

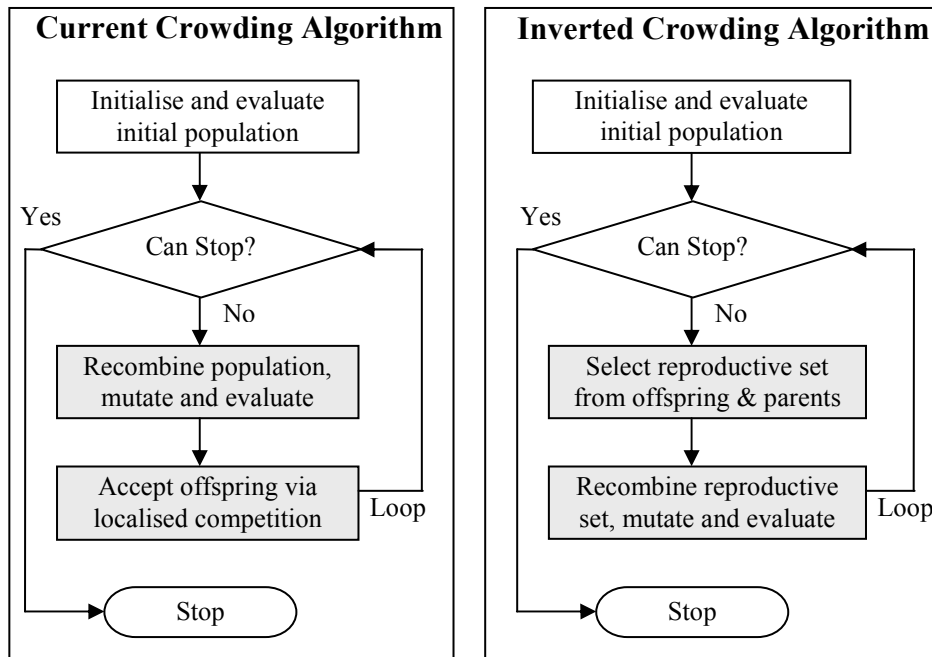


Figure 24 - Shows the current and inverted crowding algorithms

The inversion does not change the functionality of goals of the algorithm; rather it provides a new perspective for the manner in which the crowding principle is implemented in genetic search. This perspective closely aligns crowding with the canonical genetic algorithm by viewing restrictive replacement as a inter-generational upfront selection scheme. This altered perspective may provide a new facet of research into the use of inter-generational upfront selection operators in the canonical algorithm and their ability to providing niching effects. Further, this perspective may provide further insight as to how to intergrade useful augmentations to the canonical genetic algorithm into crowding such as the fitness sharing approach of formulating a fitness proportionate population distribution across areas of interest in the search space.

## Crowding Map

Localised generational competition was shown to be greedy or elitist in nature. The population is not replaced each generation, meaning that population members may live for extended number of generations. Collectively the localised selection strategy and replacement strategy can be considered as operating on the population for each offspring. Alternatively, it could be said that the population responds to offspring sequentially, where the resulting effect is a replacement or a non-replacement. This conceptional observation permits the population to be treated as an entity that adapts based on the samples it is exposed to.

Treating restrictive replacement as a population entity that responds to samples is a further abstraction of the generalised crowding model (GCM). Unlike the GCM, this concept indicates that localised selection and the replacement strategy collectively are considered a response mechanism of the population entity. The concept also makes no

assumption as to how offspring are produced. This further higher-level of abstraction of the crowding implementation is called the crowding map.

The following listing provides a further abstraction of localised generational competition called the crowding map.

```
1. Initialise map
2. Loop until stop condition
    a. Generate samples
    b. For each generated sample
        i. Update map in response to sample
```

Figure 25 – Pseudo code the crowding map conceptual model

The population entity is referred to as a map, because it implies structured representation of the solution space, which is expected to be a condensed version of areas of interest in the search space itself. Samples are generated by some means then the map is exposed to the samples sequentially, responding in some way. The generalised crowding model fits onto this conceptual model and can be a specialisation of the process. Given that the model is called a crowding map, this implies that the underlying nature of the structure is defined by and assumes the same goals as the crowding principle. In this way, crowding provides a means of managing diversity of samples given finite map storage space. As mentioned, it provides a means of condensing the search space to areas of interest given some measure of solution usefulness such as fitness.

Finite map size is assumed given practical space and time complexity concerns in search. The crowding principle is exploited because it is capable of providing stability in terms of areas of interest as well as the promotion of useful diversity, or improvement within areas of interest. The usefulness of the conceptual model is that it imposes no restriction as to how the map responds to samples, the actual structure of the map, or the manner in which samples are generated.

The conceptual model is still suited to variants of genetic search through providing a framework as to how crowding could be incorporated in a more general manner. The map is expected to respond to samples in a localised way, though the model does not explicitly specify how. An alternative implementation may measure locality in a problem specific space, or may promote problem specific characteristics in replacement competition. The map is also not limited to restrictive replacement, it is possible for example to move members closer or further away to generated samples, providing some form of plasticity to the population. Other recombination approaches could be used to incorporate samples into the map based on locality, reminiscent of self-organising data structures.

The structure and management of the map may also be varied from the genetic search convention of a population of samples. Members of the map could potentially be represented differently or transposed into a problem specific space. It may also be useful to partition the map into localities to allow easier or faster matching during replacement. The conceptual model provides more flexibility in sample generation. It further facilitates



more elaborate distributed configurations where potentially a number of conventional multimodal searches could be contributing samples to a central super-map.

The crowding map concept provides a further means of abstracting the application of the crowding principle to genetic search and may prove useful in applying the principle to other forms of search. More than an application in search, the map implies that the principle may be applicable to other classes of problems such as data reduction, clustering or a compressed sample persistence system. Further research into the capabilities of the crowding principle may reveal innovative applications of the technique both inside and outside the field of genetic search algorithms.

## ***Conclusions***

The crowding principle is defined as requiring localised competition for limited resources. Localisation implies the use of similarity between solutions, which has been shown to be implemented as similarity in representation space such as encoded or decoded values. Competition implies the use of a decision process for survival based on solution usefulness or fitness. The limited resource in this case is the fixed sized population structure. These three factors describe localised competition in genetic search, and the abstraction represented by the generalised crowding model. The goals of the crowding principle and thus localised competition are to maintain general diversity and to promote useful diversity. Simple crowding is an implementation of the abstracted model that seeks to meet the definition of localised generational competition and meet the goals of the crowding principle exactly.

The implementation of simple crowding uses crowding components that were selected to be the most representative of their function inline with the crowding principle. Preliminary results showed that the combination of selected components provided complete stability of subpopulation size and the promotion of useful diversity within each subpopulation, meeting the goals and requirements of the crowding principle in genetic search. In addition, the population was shown to exhibit two important behaviours critical to crowding and evolutionary computation.

1. Ability to achieve and maintain equilibrium (complete stability) in terms of subpopulation size, useful to the definition of crowding
2. Ability to reach a point of minimum change or convergence, useful to genetic search

The achievement of the second point by a population negates the need for the first point from then onward. Equilibrium implies a state of balance within the population both in terms of intra-niche and inter-niche interaction and is critical if diversity is to be maintained. Convergence implies a state of oneness within each subpopulation and the population as a whole, representing an endpoint for the search. These two behaviours have been shown to be compatible and indicative of quintessential restrictive replacement as a search technique.

Simple crowding was not designed to be a replacement crowding based niching genetic algorithm. It was designed to provide an embodiment of the generalised crowding model that both verifies the usefulness of the model as a framework, and validates the underlying crowding principle exploited by crowding based algorithms. To this end, the investigation was shown to be successful. The framework and template offered by the generalised crowding model was shown to be useful in describing and partitioning logical functionalities of existing crowding based algorithms. It further proved useful in the development of the simple crowding algorithm, and the augmentation of the simple crowding algorithm with functionalities from existing crowding based algorithms. The model proved that functionalities in crowding based algorithms can be isolated and interchangeable components like genetic operators in the canonical genetic algorithm.

The investigation and development of the simple crowding algorithm proved with preliminary empirical evidence that the underlying crowding principle of all the reviewed crowding based genetic algorithms is capable of maintaining diversity and promoting useful diversity. Crowding is simple, powerful and a useful tool to have in search, the effects of which is to create a condensed representation of areas of interest in the search space where population samples within an area become more similar to each other and groups become less similar to each other, in what is referred to as niching.

## Bibliography

- [1] Charles Darwin. *On the origin of species by means of natural selection, or, The preservation of favored races in the struggle for life*, Champaign : Project Gutenberg, 1859.
- [2] John H. Holland. *Adaption in Natural and Artificial Systems*, MIT Press (MIT CogNet), 1975.
- [3] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, USA, Canada: Addison Wesley Publishing Company, Inc., 1989.
- [4] Darrell Whitley, A Genetic Algorithm Tutorial *Statistics and Computing*, vol. 4, pp. 65-85, 1994.
- [5] David H. Wolpert and William G. Macready, "No Free Lunch Theorems for Search," Santa Fe Institute, Sante Fe, NM, USA, Technical Report SFI-TR-95-02-010, 1995.
- [6] Robert E. Ricklefs. *Ecology*, USA: Thomas Nelson and Sons Ltd., 1973.
- [7] Niles Eldredge. *Macroevolutionary Dynamics: Species, Niches and Adaptive Peaks*, USA: McGraw-Hill Publishing Company, 1989.
- [8] Samir W. Mahfoud, Niching methods for genetic algorithms 1995. University of Illinois.
- [9] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," *Genetic algorithms and their applications : Proceedings of the Second International Conference on Genetic Algorithms*, Massachusetts Institute of Technology, Cambridge, MA, USA, pp. 41-49, 1987.
- [10] Jean-Paul Watson, "A Performance Assessment of Modern Niching Methods for Parameter Optimization Problems," *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, Orlando, Florida, USA, pp. 702-709, 1999.
- [11] B. L. Miller and M. J. Shaw, "Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization," *Proceedings of IEEE International Conference on Evolutionary Computation (ICEC '96)*, Nayoya University; Japan, pp. 786-791, 1996.
- [12] Kenneth A. De Jong, An Analysis of the Behavior of a Class of Genetic Adaptive

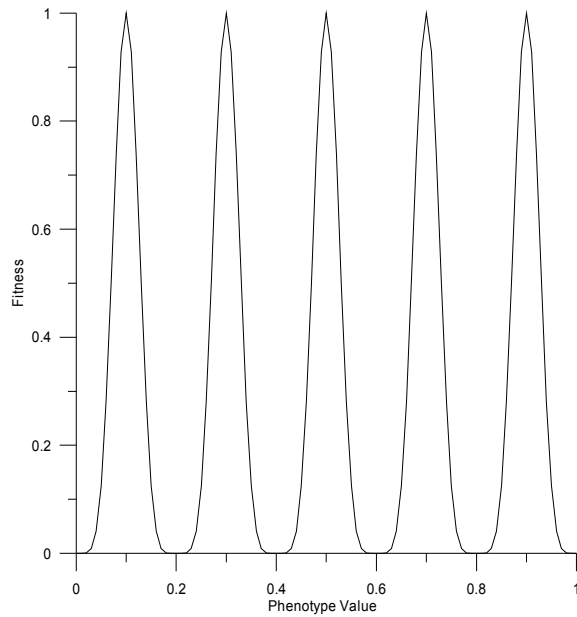
Systems 1975. University of Michigan.

- [13] S. W. Mahfoud, "Crowding and preselection revisited," *Parallel Problem Solving from Nature 2*, Brussels, Belgium, pp. 27-36, 1992.
- [14] S. W. Mahfoud, "Simple Analytical Models of Genetic Algorithms for Multimodal Function Optimization," University of Illinois, IlliGAL Report No 93001, 1993.
- [15] S. W. Mahfoud, "Crossover Interactions Among Niches," *Proceedings of the First IEEE Conference on Evolutionary Computation; IEEE World Congress on Computational Intelligence*, Orlando, Florida, USA, pp. 188-193, 1994.
- [16] G. R. Harik , "Finding Multiple Solutions In Problems Of Bounded Difficulty," University of Illinois, IlliGAL Report No. 94002, 1994.
- [17] G. R. Harik , "Finding Multimodal Solutions Using Restricted Tournament Selection," *Proceedings of the Sixth International Conference on Genetic Algorithms (ICGA)*, pp. 24-31, 1995.
- [18] W. Cedeño, V. R. Vemuri, and T. Slezak, Multi-Niche Crowding in Genetic Algorithms and its Application to the Assembly of DNA Restriction Fragments *Evolutionary Computation*, vol. 2, pp. 321-345 , 1994.
- [19] Walter Cedeño, The Multi-Niche Crowding Genetic Algorithm: Analysis and Application 1995. University of California Davis.
- [20] W. Cedeño and V. R. Vemuri, "On the use of niching for dynamic landscapes," *IEEE International Conference on Evolutionary Computation (ICEC)*, Indiana University Purdue University; Indianapolis; USA, 1997.
- [21] O. J. Menshoel and D. E. Goldberg, "Probabilistic crowding: deterministic crowding with probabilistic replacement," University of Illinois, IlliGAL Report No. 99005 , 1999.

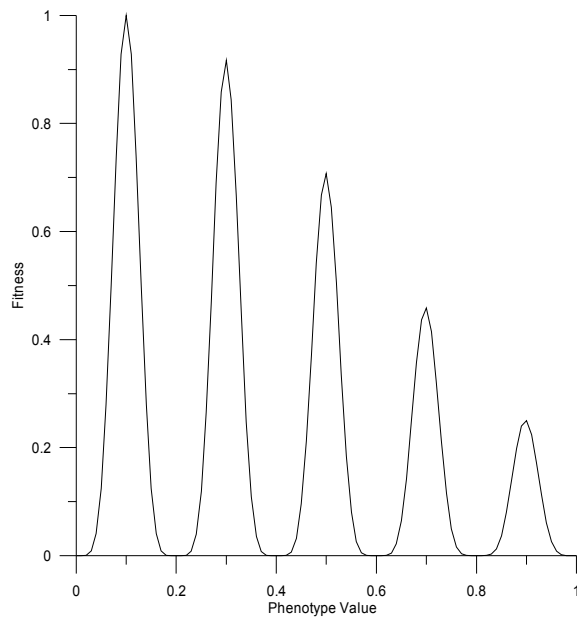
## Appendix A – Test Functions

This section provides a visual representation of each of the selected test problems.

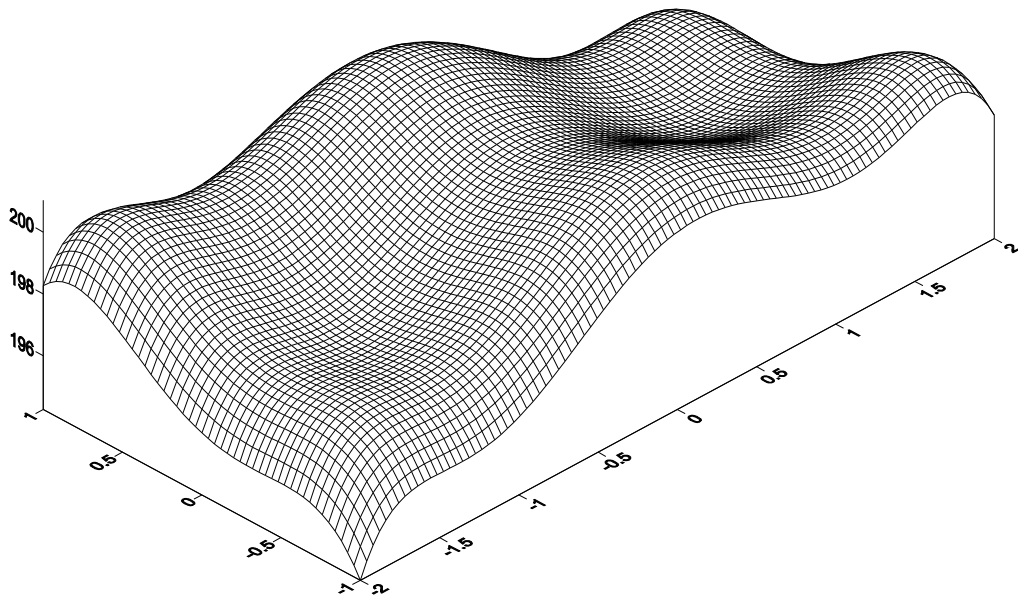
### ***F1 - Sine Function - 5 Peaks of equal height***



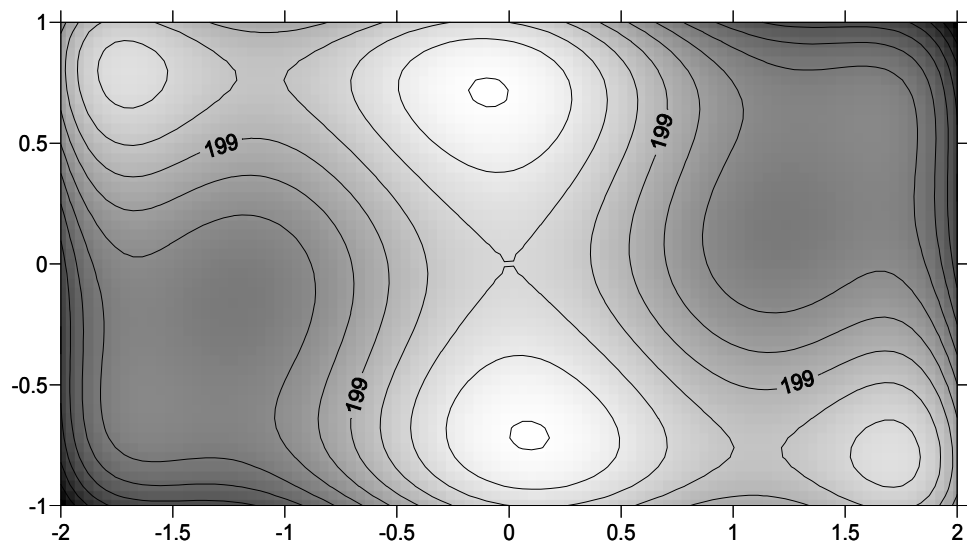
### ***F2 - Sine Function - 5 Peaks of differing height***



***F3 - Scaled six-hump camel back function***

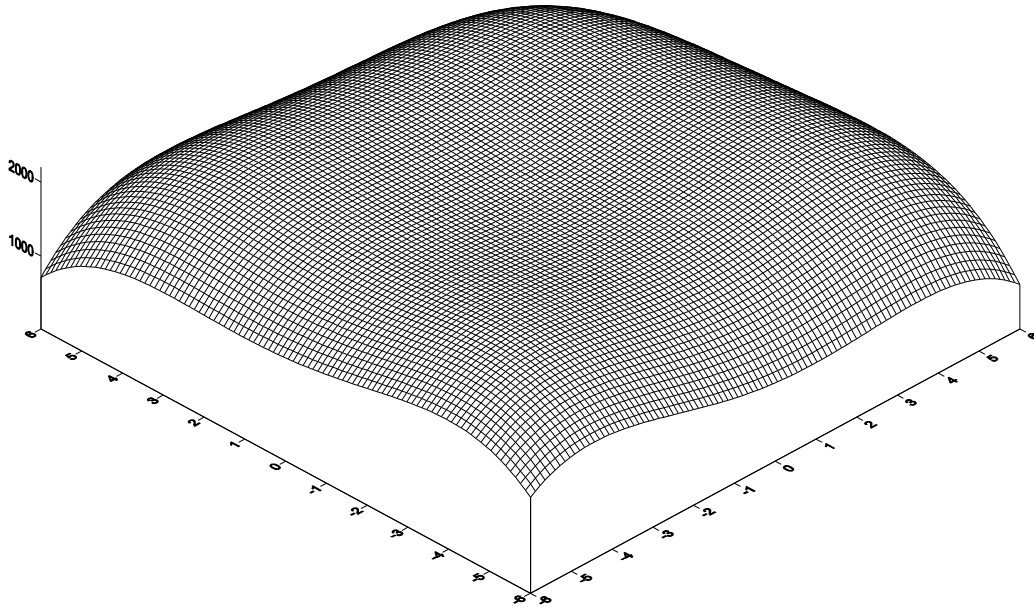


F3 - Wireframe Plot

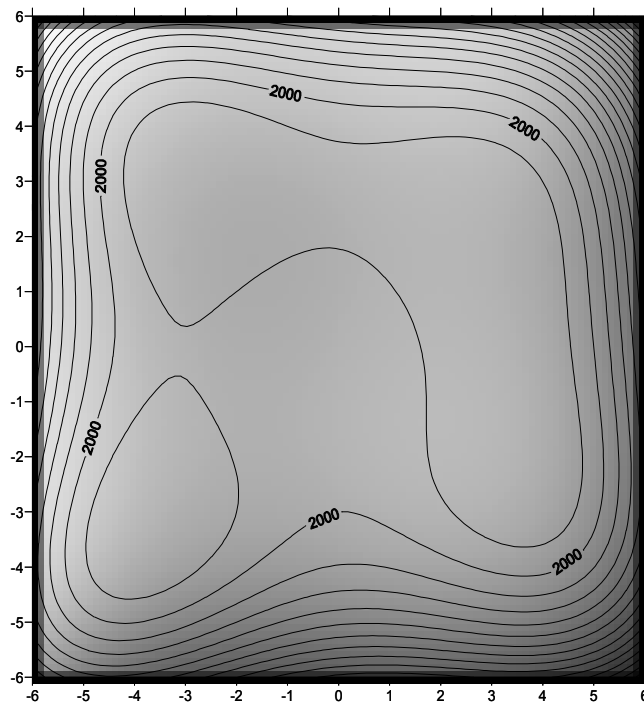


F3 - Contour Plot

## *F4 - Himmelbau's Function*

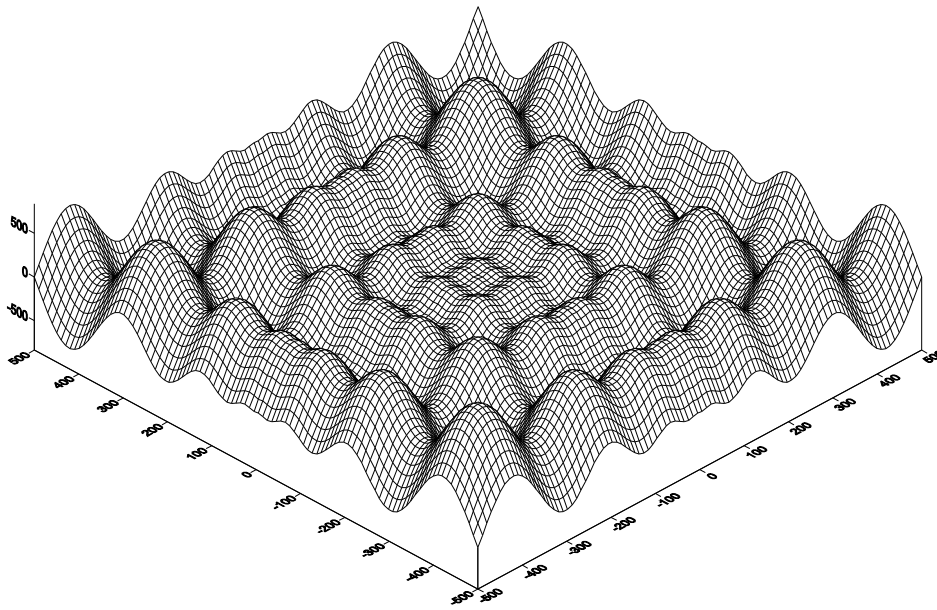


F4 - Wireframe Plot

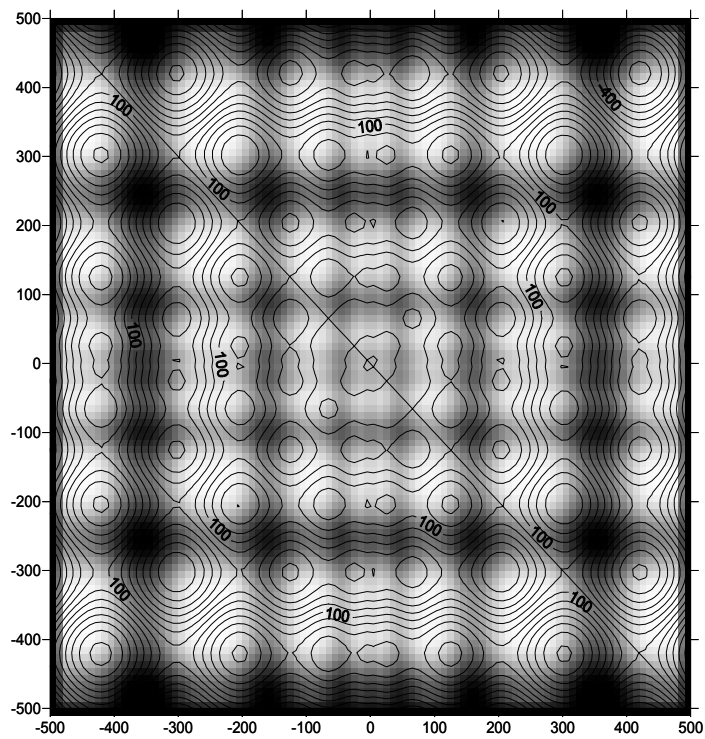


F4 - Contour Plot

## *F5 - Schwefel's Function*



F5 - Wireframe Plot



F5 - Contour Plot



## Appendix B – Test Results

All test results are provided in Microsoft Excel format including both raw result and graph representations of the results discussed in this work. All results are provided on the associated CD-ROM and are organised into the sections in which they appeared in this work.

### *Investigation into Quintessential Crowding*

<b>Directory</b>	<b>Filename</b>	<b>Description</b>
1_Quintessential_Crowding	A1.xls	Test results for the A1 algorithm on F1 and F2
	A2.xls	Test results for the A2 algorithm on F1 and F2
	A3.xls	Test results for the A3 algorithm on F1 and F2

Table 5 – Summary of the directory and filenames where the investigation into quintessential crowding test results are located on the CD-ROM

### *Augmented Simple Crowding*

<b>Directory</b>	<b>Filename</b>	<b>Description</b>
2_Augmented_PC	1_Upfront_Selection.xls	Test results for the simple crowding algorithm with biased upfront selection on the F1 and F2 test problems
	2_Localised_Selection.xls	Test results for the simple crowding algorithm with biased localised selection strategies on the F1 and F2 test problems
	3_Replacement_Strategies.xls	Test results for the simple crowding algorithm with alternative replacement strategies on the F1 and F2 test problems

Table 6 – Summary of the directory and filenames where the augmented simple crowding test results are located on the CD-ROM

### *Simple Crowding Compared*

<b>Directory</b>	<b>Filename</b>	<b>Description</b>
3_PC_Comparison	1_RTS.xls	Test results for the restricted tournament selection algorithm for test problems F1-F5
	2_DC.xls	Test results for the deterministic crowding algorithm for test problems F1-F5
	3_ProbC.xls	Test results for the probabilistic crowding algorithm for test problems F1-F5

	4_SimpC.xls	Test results for the simple crowding algorithm for test problems F1-F5
--	-------------	---

Table 7 – Summary of the directory and filenames where the simple crowding compared test results are located on the CD-ROM

## Appendix C – Generalised Crowding Model

The following diagram provides a summary of the generalised crowding model in genetic search with a focus on the three components critical to achieving the goals of the crowding principle.

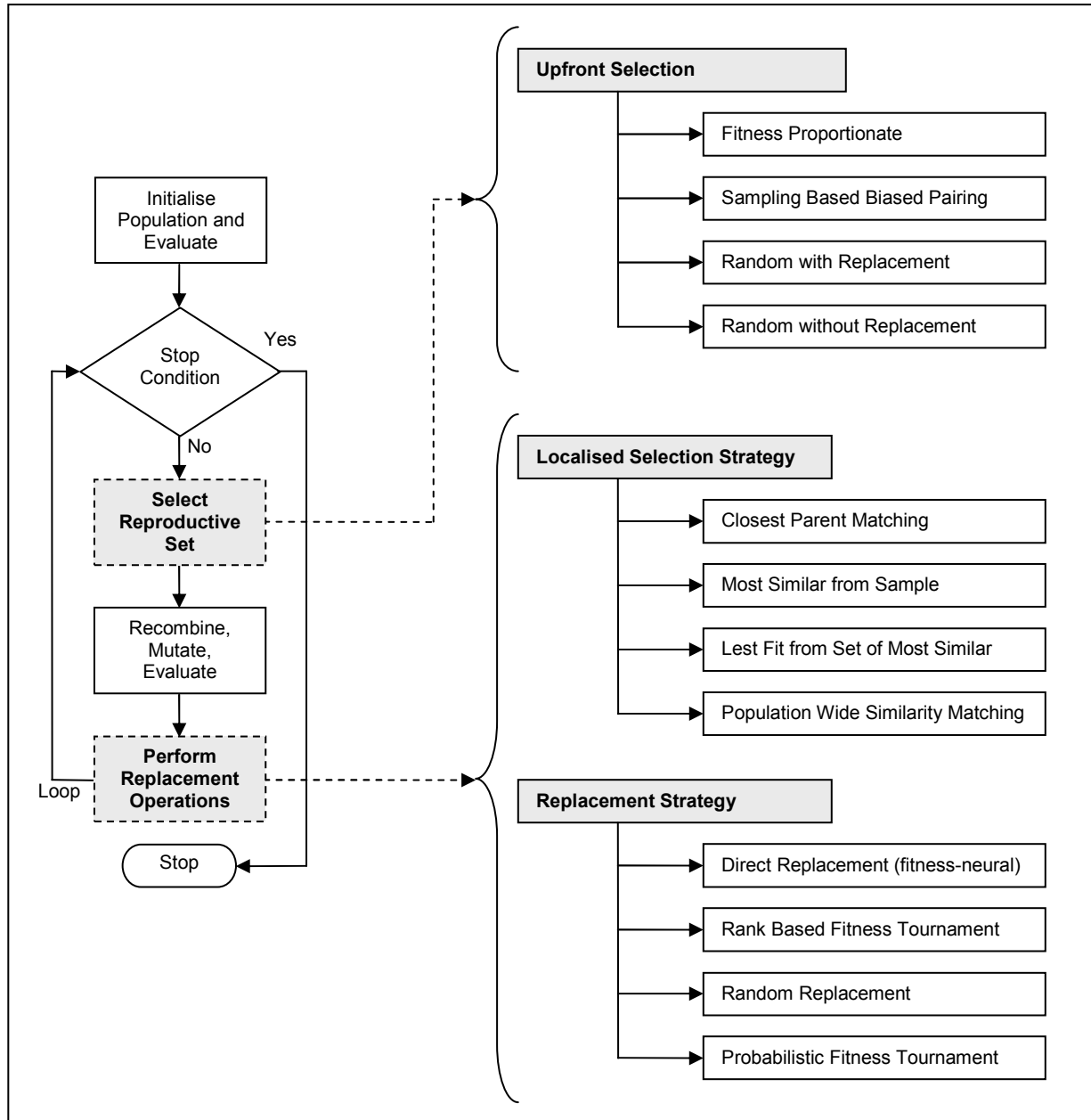


Figure 26 – Shows a summary of the generalised crowding model in genetic search