

A STUDY OF MATE SELECTION IN GENETIC ALGORITHMS

by

Chien-Feng Huang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2002

Doctoral Committee:

Professor John H. Holland, Co-Chair

Associate Research Scientist Rick Riolo, Co-Chair

Professor Robert K. Lindsay

Professor N. Harris McClamroch

© Chien-Feng Huang 2002
All Rights Reserved

To my family.

ACKNOWLEDGEMENTS

I would like to thank my advisors, Professor John H. Holland and Dr. Rick L. Riolo, for numerous valuable discussions concerning this research; for motivation, guidance and their willingness to take on a new student back in 1998.

It is with appreciation and gratitude that I acknowledge the assistance and perspective afforded to me by the other members of my doctoral committee: Professor Robert K. Lindsay and Professor N. Harris McClamroch.

I am also fortunate to have several colleagues in the Royal Road group at the University of Michigan: Ted Belding, Leeann Fu, Tommaso Bersano-Begey, and Bill Rand. I would like to thank them for their valuable comments and suggestions.

I would like to acknowledge the full support of my parents, Shih-Tsung Huang and Shou-Hwa Huang, my brothers Art Huang and Cliff Huang, and my sister Mei Huang. I would also like to thank Kai-Chi Lu, for her generous support through the period of constructing my thesis.

Finally, I would like to acknowledge the generous financial support I have received from the Center for the Study of Complex Systems at the University of Michigan.

ABSTRACT

A STUDY OF MATE SELECTION IN GENETIC ALGORITHMS

by
Chien-Feng Huang

Co-Chairs: John H. Holland and Rick Riolo

The process of information exchange among the population of individuals manipulated by Genetic Algorithms (GAs) involves two key components: crossover and mate selection. The central theme of this thesis concentrates on the investigation of effects of **mate selection** in GAs. The importance of mate selection in biology is widely recognized, yet a systematic investigation of this subject in GA research is still lacking. The goal of this thesis is to propose a framework that facilitates exploration of mate selection in GAs in order to (1) gain a deeper understanding of how GAs work, (2) how to design more robust GAs, and (3) shed more light on why mate selection matters in biology.

The first four chapters of this thesis present motivations for this work, and describe investigations of the basic properties of mate selection in the context of GA. I employ the Schema Theorem and a Markov model as analytical tools to facilitate the study of mate selection. A number of empirical results are also presented to enhance our understanding of the GAs' behavior. The results based on simple test problems highlight the importance of mating choices in improving the GA's performance.

Next, this study focuses on two classes of more complicated, building-block-based problems—the Royal Road functions and the hyperplane-defined functions. With the

results further obtained, I introduce an important hypothesis regarding the role of mate selection in GAs. That is, if one’s goal is to improve the GA’s search for best-so-far solutions, then on easy problems a dissimilarity-based mate selection scheme is more beneficial. If problems present sufficient difficulty, the GA’s search power can be further improved by by reducing the selection pressure toward higher-fitness individuals while selecting mates.

Chapter 6 presents a test of this hypothesis based on several more realistic, non-building-block-based benchmark testbeds. The test problems used are of increasing complexity in terms of various aspects of fitness landscapes. The first two testbeds, a sphere function and a step function, represent unimodal problems. The following four testbeds are multimodal in which characteristics of fitness landscapes such as deception and non-separability are included—the generalized Rosenbrock Saddle, an optimal control problem, a modified version of the Schaffer function F_7 and Michalewicz’s epistatic function. All the results of the experiments validate this hypothesis. This is encouraging—it implies that the ideas of mate selection proposed in this thesis can be applied to practical problems.

Chapter 7 discusses a more general setting in the context of multimodal function optimization, engineering and machine learning. Identifying multiple peaks and maintaining subpopulations of the search space are two central themes. An immune system model is employed to study these two problems. The experimental results indeed shed more light on how mate selection schemes compare to traditional selection schemes.

The final chapter provides a summary of this thesis, and highlights its contributions to GA research. It discusses paths for future research, and draws overall conclusions from the research presented in this thesis.

TABLE OF CONTENTS

DEDICATION	ii
ACKNOWLEDGEMENTS	iii
LIST OF FIGURES	iv
LIST OF TABLES	vii
LIST OF APPENDICES	ix
CHAPTERS	
1 INTRODUCTION	1
1.1 Evolutionary Algorithms	1
1.2 Genetic Algorithms	3
1.2.1 Sex in Evolution	6
1.3 Goal and Outline	9
2 MOTIVATION	12
2.1 Introduction	12
2.2 Biological Background	15
2.3 Non-Random Mating in Prior GA Research	17
3 METHODOLOGIES AND GENERAL ANALYSIS	23
3.1 Introduction	23
3.2 Mate Selection Schemes	25
3.3 Schema Theorem Analysis	27
3.3.1 Review of The Schema Theorem	27
3.3.2 Interpretation of the Effects of Mate Selection By The Schema Theorem	29
3.4 Empirical Analysis	30
3.4.1 Hitchhiking	31
3.4.2 Founder Effect	37
3.4.3 Rate of Improvement and Creation of Lethal Hybrids	42
3.4.4 Diversity	47
3.4.5 Tournament Mate Selection	53

3.4.6	Effects of Deceptiveness	60
3.4.7	Effects of Population Size	62
3.5	Summary and Discussions	64
4	MARKOV CHAIN ANALYSIS	67
4.1	Introduction	67
4.2	The Nix and Vose Markov Model	68
4.3	Incorporation of Mate Selection in the Nix and Vose Markov Model	70
4.3.1	Visualizing Markov Models and Diversity	75
4.4	The GAFO Theory	81
4.4.1	Expected Waiting Time Analysis	81
4.5	Summary and Discussions	85
5	BUILDING-BLOCK-BASED TEST FUNCTIONS	87
5.1	Introduction	87
5.2	Experiments on Royal Road Functions	89
5.2.1	Royal Road R_1	91
5.2.2	Royal Road R_2	95
5.2.3	Incompatible Royal Road IR_1	100
5.3	Experiments on Hyperplane Defined Functions	105
5.3.1	Hyperplane-Defined Test Function H_1	107
5.3.2	Hyperplane-Defined Test Function H_2	111
5.3.3	Hyperplane-Defined Test Function H_3	112
5.4	Summary and Discussions	116
6	EMPIRICAL VALIDATION	118
6.1	Introduction	118
6.2	Modified De Jong Function F_1	121
6.3	De Jong Function F_3	125
6.4	Generalized Rosenbrock Function	127
6.5	An Optimal Control Test Problem	131
6.6	Modified Schaffer Function F_7	134
6.7	Epistatic Michalewicz Function	138
6.8	Summary and Discussions	139
7	IMMUNE SYSTEMS	142
7.1	Introduction	142
7.2	Immune System Applications	144
7.2.1	Fault Diagnosis	144
7.2.2	Control	145
7.2.3	Computer Security	146
7.3	Fitness Sharing	147
7.4	Binary Immune System Model	150
7.4.1	Emergent Fitness Sharing in the Immune System Model	152

7.5	Experimental Results	155
7.5.1	Effects of Mate Selection on Maintaining Subpopulations	158
7.5.2	Effects of Crossover Rate	164
7.5.3	Effects of Mate Selection on the Discovery of Peaks .	167
7.6	Summary and Discussions	169
8	CONCLUSIONS	171
8.1	Summary	171
8.2	Future Work	173
	APPENDICES	176
	BIBLIOGRAPHY	188

LIST OF FIGURES

Figure	
3.1	Schema dynamics for observing hitchhiking (Maximum Similar Mating). 34
3.2	Schema dynamics for observing hitchhiking (Maximum Similar Mating). 34
3.3	Schema dynamics for observing hitchhiking (Maximum Dissimilar Mating). 36
3.4	Schema dynamics for observing hitchhiking (Maximum Dissimilar Mating). 36
3.5	Schema dynamics for observing the founder effect (Maximum Similar Mating). 39
3.6	Schema dynamics for observing the founder effect (Maximum Similar Mating). 39
3.7	Schema dynamics for observing the founder effect (Maximum Dissimilar Mating). 41
3.8	Schema dynamics for observing the founder effect (Maximum Dissimilar Mating). 41
3.9	Best-so-far performance on S_1 44
3.10	Best-so-far performance on IS_1 44
3.11	Averaged population fitness on S_1 46
3.12	Averaged population fitness on IS_1 46
3.13	Proportions of lethal offspring on IS_1 48
3.14	Several averaged fitness on IS_1 48
3.15	Averaged allelic diversity on S_1 52
3.16	Averaged allelic diversity on IS_1 52
3.17	Best-so-far performance on IS_1 54
3.18	Averaged population fitness on IS_1 54
3.19	Best-so-far performance on S_1 57
3.20	Best-so-far performance on IS_1 57
3.21	Averaged population fitness on IS_1 59
3.22	Proportions of lethal offspring on IS_1 59
3.23	Best-so-far performance on D_1 61
3.24	Averaged Best-so-far on IS_1 for various population sizes. 63
4.1	Visualization of Q^k ($l=2, n=3$). 76

4.2	Visualization of Q^k ($l=3, n=3$).	76
4.3	Number of states with transition probability over the threshold $\frac{1}{N}$	78
4.4	Averaged allelic diversity.	80
4.5	Interacting effects of mate selection and mutation.	82
4.6	Interacting effects of mate selection and crossover.	83
4.7	Effects of Scaling.	84
5.1	Best-so-far performance on R_1	93
5.2	Best-so-far performance on R_2	97
5.3	Ratio of the normalized averaged best-so-far of R_2 to that of R_1	99
5.4	Best-so-far performance on incompatible IR_1	102
5.5	Ratio of the normalized averaged best-so-far of IR_1 to that of R_1	104
5.6	Histogram for best-so-far performance on H_1	110
5.7	Best-so-far performance on H_1	110
5.8	Histogram for best-so-far performance on H_2	113
5.9	Best-so-far performance on H_2	113
5.10	Histogram for best-so-far performance on H_3	115
5.11	Best-so-far performance on H_3	115
6.1	Modified, two-dimensional De Jong function F_1	122
6.2	Averaged best-so-far performance on the modified De Jong function F_1	124
6.3	Vector field of the two-dimensional modified De Jong function F_1	124
6.4	Two-dimensional De Jong function F_3	125
6.5	Modified, two-dimensional Rosenbrock Saddle.	128
6.6	Vector field of the two-dimensional Rosenbrock Saddle.	128
6.7	Averaged best-so-far performance on the generalized Rosenbrock function.	130
6.8	Histogram for the best-so-far values on the generalized Rosenbrock function.	130
6.9	Optimal control test problem.	133
6.10	Magnified schematic of the optimal control test problem.	133
6.11	Averaged best-so-far performance on the optimal control test problem.	135
6.12	Histogram for the best-so-far values on the optimal control test problem.	135
6.13	Modified Schaffer function F_7	136
6.14	Averaged best-so-far performance on the modified Schaffer function F_7	137
6.15	Histogram for the best-so-far values on the modified Schaffer function F_7	137
6.16	Modified, two-dimensional Michalewicz function.	139
6.17	Averaged best-so-far performance on the epistatic Michalewicz function.	140
6.18	Histogram for for the best-so-far values on the epistatic Michalewicz function.	140
7.1	The sharing function $sh(d_{ij})$ for $\alpha=0.5, 1$ and 2	148
7.2	The number of antibodies that correctly recognize antigens.	161
7.3	Averaged allelic diversity.	161

7.4	The number of antibodies that correctly recognize antigens (based on the tournament similar mating), where all portion of the solid line (i.e., corresponding to 000...0) after generation 10 is on the 0 level.	165
7.5	Schema dynamics for observing the effect of the tournament similar mating.	165
7.6	The number of antibodies that correctly recognize antigens (crossover rate = 0.7).	166
7.7	The number of antibodies that correctly recognize antigens (crossover rate = 0.3).	166
7.8	The number of antibodies that correctly recognize antigens (crossover rate = 0).	167

LIST OF TABLES

Table

1.1	The structure of an evolutionary algorithm.	2
1.2	Mechanism of a simple GA.	4
3.1	Small royal road function S_1	32
3.2	Incompatible small royal road function IS_1	37
3.3	A Deceptive, Order-5 Problem.	61
4.1	The Z matrix ($n=2$ and $l=2$).	70
4.2	Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the maximum similar mating.	73
4.3	Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the maximum dissimilar mating.	73
4.4	Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the proportional similar mating.	74
4.5	Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the proportional dissimilar mating.	74
5.1	Schematic of Royal Road R_1	91
5.2	Mean function evaluations to the optimum or the number of runs out of 50 in which the optimum is found on R_1	92
5.3	The mean function evaluations of first appearance of a schema or schema combinations of increasing order for R_1 (over 50 runs).	94
5.4	Schematic of Royal Road R_2	96
5.5	The mean function evaluations of first appearance of a schema or schema combinations of increasing order for R_2 (over 50 runs).	98
5.6	Schematic of the incompatible Royal Road IR_1	101
5.7	The mean function evaluations of first appearance of a schema or schema combinations of increasing order for IR_1 (over 50 runs).	103
5.8	Number of runs reached fitness value 80 in 100 runs on H_1	108
5.9	Number of runs reached fitness value 221 in 100 runs on H_2	112
5.10	Number of runs reached fitness value 430 in 100 runs on H_3	114
6.1	Mean function evaluations to the optimum (population size 50).	123
6.2	Mean function evaluations to the optimum (population size 20).	123
6.3	Mean function evaluations to the optimum.	126

6.4	Number of runs out of 50 in which the optimum is found.	129
7.1	Building blocks of antigens.	157
7.2	Illustration of the immune-GA's mechanism.	159
7.3	The mean function evaluations of discovering antibodies 111...1 and 000...0 (over 50 runs).	168
7.4	The number of runs (out of 50) in which antibodies 111...1 and 000...0 are discovered.	168

LIST OF APPENDICES

APPENDIX

A	Description of HDF H_1	177
B	Description of HDF H_2	179
C	Description of HDF H_3	182

CHAPTER 1

INTRODUCTION

1.1 Evolutionary Algorithms

During the last two decades there has been a growing interest in algorithms that are based on the principle of evolution, i.e., survival of the fittest. A common term, accepted recently, refers to such techniques as evolutionary algorithms (EAs) (or evolutionary computation). EAs use computational models of evolutionary processes as key components in the design and implementation of computer-based problem solving systems. There are a variety of evolutionary computational models that have been proposed and studied. They are conveniently presented using the metaphor of natural evolution: a randomly initialized population of individuals (set of points of the search space at hand) evolves following an idealized model of the Darwinian principle of the survival of the fittest. New individuals are generated through variation introduced by operators such as recombination and mutation. The probability of survival of the newly generated solutions depends on their fitness (perceived performance in the environment): the probability of selection is an increasing function of fitness, so that the best are kept with a high probability, the worst are rapidly discarded. Although simplistic from a biologist's viewpoint, these algorithms are sufficiently complex to provide robust and powerful adaptive search mechanisms.

There are several variants of EAs and each of them may be distinguished by different selection schemes and variation operators, yet the structure of any evolutionary

Table 1.1: The structure of an evolutionary algorithm.

1. $t = 0$.
2. Randomly generate an initial population of n individuals.
3. Evaluate each individual's fitness.
4. Repeat until n offspring have been created.

 $t = t + 1$;
 select parents;
 generate offspring by altering the parents;
 evaluate the offspring.
5. Replace the current population with the new population.
6. Go to Step 3 until terminating condition.

model is very similar. Table 1.1 outlines a sample EA structure. Typically, a population of n individuals is initialized randomly and then evolved from generation t to generation $t + 1$ by repeated fitness evaluation, selection, and variation. Fitness evaluation may be as simple as computing a mathematical function or as complex as running an elaborate simulation. Individuals are then selected for survival and reproduction. Many selection schemes have been developed, including (1) fitness-proportionate selection (Holland, 1975), where the probability of selection is proportional to the individual's fitness, (2) sigma-scaling selection (Forrest, 1985), which keeps the selection pressure relatively constant over the course of the run rather than depending on the fitness variances in the population, (3) rank selection (Baker, 1985), in which the individuals in the population are sorted according to their fitnesses, and the probability of an individual being selected depends on its rank in the population rather than on its absolute fitness, and (4) tournament selection (Goldberg and Deb,

1991), where some number of individuals (usually two) compete for selection and this competition step is repeated population-size number of times. After selection, individuals are perturbed by variation operators such as crossover, mutation, and others.

The origins of evolutionary algorithms can be traced to at least the 1950's (Fraser 1957; Box 1957). As of today, there are four dominant methodologies: "Genetic Algorithms" (Holland, 1975), "Genetic Programming" (Koza, 1992), "Evolutionary Programming" (Fogel, Owens, and Walsh, 1966), "Evolutionary Strategies" (Rechenberg, 1973; Schwefel, 1981). In the next section, I briefly describe the ideas and mechanism of the most well known EA, the Genetic Algorithm. This type of EA is the central theme of this thesis.

1.2 Genetic Algorithms

The Genetic algorithm (GA), developed by Holland (1975), is based on the mechanics of natural selection and natural genetics. The term *genetic algorithm* describes the basic idea: algorithmic techniques, inspired by genetic principles, are used to study the phenomenon of adaptation as it occurs in nature. The objective is to design and implement robust, adaptive systems, following nature's paradigm for the evolution of genetic structures.

Table 1.2 depicts the process of a simple genetic algorithm, where the variation step in Table 1.1 is implemented by crossover and mutation operators. Traditionally, GAs have used a domain independent representation, namely, binary strings (chromosomes) for population individuals. A population of n individual structures is initialized, each bit value taking 1 or 0 with equal probability. After initialization, pairs of parent chromosomes are selected for reproduction, and the probability of individuals being selected is an increasing function of fitness. A common selection scheme in GAs is *fitness-proportional selection*, in which the number of times an indi-

Table 1.2: Mechanism of a simple GA.

1. Randomly generate an initial population of n individuals.
2. Evaluate each individual's fitness.
3. Repeat until n offspring have been created.
 - a. select a pair of parents for mating;
 - b. apply crossover operator;
 - c. apply mutation operator.
4. Replace the current population with the new population.
5. Go to Step 2 until terminating condition.

vidual expected to reproduce is equal to its fitness divided by the average of fitnesses in the population. With this selection method the average fitness of the population is monitored. Those individuals that have higher than average fitness produce (on the average) more than one child, while those that have less than average fitness produce (on the average) less than one child. This is normalized appropriately to produce n children, which are created via crossover and mutation from the n parents. The n children then replace the n parents in the population.

Holland (1975) formulated the theoretical foundation of GA by the notion of *schemata*, which are sub-structures of strings. Holland likened these schemata to the random variables associated with K-armed bandit problems, and argued that the GA maximizes accumulated payoff by optimizing the allocation of trials to those random variables. His analysis suggests that selection increasingly focuses the search on subsets of the search space with estimated above-average fitness. These results form the fundamental theorem of Genetic Algorithm, namely, the **Schema Theorem** (Holland, 1975):

“Short, low-order, above-average schemata receive exponentially increasing trials in subsequent generations of a genetic algorithm.”

Another key idea that parallels this theorem is that GAs explore the search space by short, low-order, high-fit schemata which, subsequently, are recombined to form even more highly fit higher-order schemata by crossover. This statement is well known as the **Building Block Hypothesis** (Holland, 1975; Goldberg, 1989).

The ability to produce fitter and fitter partial solutions by combining building blocks is believed to be a primary source of the GA’s search power; thus crossover has been considered to be the primary search operator that distinguishes GAs from most other search algorithms. Mutation, on the other hand, was regarded as a background operator that serves as the role of an insurance policy, ensuring that genetic diversity is never lost at any locus.

It is worth noting that higher mutation rates increasingly disrupt higher order schemata, while higher crossover rates increasingly disrupt higher defining-length schemata. When selection is factored in, a population converges at a rate proportional to the ratio of the best individual’s fitness to average population fitness. This ratio is one measure of selection pressure (Back, 1994). Increasing either crossover or mutation rate, or decreasing the selection pressure, leads to wider sampling of the search space, but does not allow as much exploitation of useful schemata that the GA locates. On the other hand, decreasing either crossover or mutation rate, or increasing the selection pressure, leads to increased use of the beneficial building blocks, but does not allow as much exploration of the search space. This is commonly known as a tension between “exploration” (the search for new, useful building blocks) and “exploitation” (the use and propagation of these building blocks). In general, a robust GA must maintain an often delicate balance between exploration and exploitation so as to well adapt to its environment.

1.2.1 Sex in Evolution

The idea of information exchange among a GA's population of individuals through crossover, in a broad sense, can be regarded as modeling sex. When population geneticists talk about sex, what they usually mean is reproduction involving (1) meiosis with crossing over; and (2) mating between unrelated individuals, such as occur during random mating (Freeman and Herron, 1998, p. 197). As such, sex in evolution involves two key components: **crossover** and **mate selection**. This subsection discusses the prevalence of crossover and mate selection in the context of GA.

Crossover

In biology, there are four main explanations proposed for the importance of crossover (Holland, 2000):

- Crossover provides long (random) jumps in the space of possibilities, thus providing a way off of local maxima.
- Crossover repairs mutational damage by sequestering deleterious mutations in some offspring while leaving other offspring free of them.
- Crossover provides persistent variation that enables organisms to escape adaptive targeting by viruses, bacteria, and parasites.
- Crossover recombines building blocks.

In GA research, crossover has received extensive exploration since the seminal work of John Holland in the 1960s. The idea behind crossover was to take sub-parts from individuals that supply different sub-solutions and combine them to improve solutions. Among several versions of crossover operators, single-point crossover is the simplest form: it was defined (Holland, 1975) as taking two strings representing parents and exchanging the set of attributes that follow a randomly chosen position to

form offspring. Other forms of crossover have been investigated, such as two-point, n-point crossover, and uniform crossover (Syswerda, 1989; Spears and De Jong, 1991a). In contrast with these problem-independent operators, another research line was conducted to investigate self-adaptive crossover mechanisms (Schaffer and Morishima, 1987; Spears, 1995). Riopka and Bock (2000) also implemented intelligent recombination through the learning of GAs' individuals. On the other hand, Jones (1995) investigated crossover from an entirely different perspective: it is in essence a “macro-mutation” operator that simply performs large jumps in the search space. In addition, Chen (1999) and Watson and Pollack (2000) have noted two different properties of crossover: preservation of similarity and combination of differences.

Mate Selection

Another relevant issue of the role of sex in evolution is mate selection, or mating strategies. In biology, mating strategies are generally classified as random mating and non-random mating. Random mating is referred to as *panmixia*, where each individual member of a population has an equal chance (probability) of mating with every other individual of the opposite sex. In this case, the probability of mating is not determined by genotype; each individual mates without preference. If any degree of preferential matings between individuals exists and results in particular genotypes mating more frequently than would be expected at random, it constitutes non-random mating.

In the context of GA, random mating can be implemented in simple GAs where individuals are selected for mating according to their fitnesses relative to the average fitness of population, such as fitness-proportionate selection. In this case, individuals are not allowed to actively determine their mating partners because mating preference is not implemented.

On the other hand, non-random mating is closely related to speciation—formation

of reproductively isolated groups of organisms. The result of reproductively isolation is that individuals of a population are not able to mate with individuals of another reproductive isolated population. In GA research, speciation mechanisms were conceived of mainly based on two classes of speciation, namely, allopatric speciation (Mayr, 1942) and sympatric speciation (Dobzhansky, 1937). Allopatric speciation imposes direct or indirect mate restriction on mating through geographic barriers. Canonical examples of direct restricting mating via geographic barriers are fine-grained parallel GAs (PGAs), where each individual is allocated at each deme and crossovers occur only between individuals that are near one another geographically. This is a form of local mating.

As opposed to the local mating algorithms, coarse-grained PGAs indirectly restrict mating in which the population is explicitly divided into smaller subpopulations. Each subpopulation is isolated from the others in the sense that it evolves independently with occasional migrations of individuals from one subpopulation to another. The resulting mating can only take place within geographically separate groups, except the migrating individuals. (A nice review regarding the existing PGA models can be found in (Cantú-Paz, 1997).)

Sympatric speciation, on the other hand, restricts mating by explicit rules, rather than by geographic barriers. For example, Eshelman and Schaffer (1991) implemented a method to prevent incest: they disallowed sufficiently similar individuals (in terms of the Hamming distance between their genotypes) to mate, aiming to keep the population as diverse as possible. On the other hand, Deb and Goldberg (1989) restricted mating between phenotypically similar individuals in order to facilitate speciation to solve multimodal function optimization problems. Other mating mechanisms such as tag-added mating schemes (Spears, 1994), template-added mating schemes (Booker, 1985) and sexual selection (Todd and Miller, 1991) have the capability to generate speciation, as well. In addition, the Distributed Breeder Genetic Algorithms invented

by Mühlenbein and Schlierkamp-Voosen (1993) were designed to study artificial selection similar to that used by human breeders. In their models, mating is based on rational selection performed by human breeders, rather than by natural selection.

The central theme of this thesis is to systematically investigate mate selection based on similarity tests that belong to the category of sympatric speciation. In particular, by allowing individuals to actively select mates, fitnesses of candidate mating partners are dynamically re-scaled by the individual who chooses them. This would introduce another source of selection pressure, in addition to the selection pressure arising from the environment. Therefore, we can expect that these two sources complicate the actual probability of an individual being selected for reproduction.

Before delving fully into the thesis, the rest of this chapter discusses the goal of this thesis and provides an outline for the remaining chapters.

1.3 Goal and Outline

I have briefly discussed in the preceding section that the process of information exchange among a GA's population individuals involves two key components: crossover and mate selection. This thesis concentrates on the investigation of effects of **mate selection** in GAs. The prevalence of mate selection in biology is widely recognized, yet a systematic investigation of this subject in GA research is still lacking. The goal of this thesis is to propose a framework that facilitates exploration of mate selection in GAs in order to (1) gain a deeper understanding of how GAs work, (2) how to design more robust GAs, and (3) shed more light on why mate selection matters in biology.

Chapter 2 discusses, in depth, my motivation for conducting research on mate selection. Starting from an example in the context of GA, I indicate potential problems that occur in a simple implementation of GA. Then some biological background of mate selection is described. Afterwards, I present a literature review of prior work

that is relevant to mating strategies, where I focus on the distinction between similarity and dissimilarity-based mate selection schemes.

In Chapter 3, I propose a framework for investigating mate selection schemes in the context of GA. The major point is that by allowing individuals to select mates, fitnesses of candidate mates are dynamically re-scaled by the individual who chooses them. This results in a system in which population members' fitnesses depend on other individuals, rather than being determined only by the environment. I demonstrate that some biological phenomena, such as hitchhiking and the founder effect, can be explained in GA's context. These phenomena, in general, have significant as well as negative effects on GA's search power, and the results show that proper mating choices can reduce these effects. I then proceed to examine performance of GAs with other mate selection schemes. The results obtained illustrate that mating preferences indeed help discriminate population individuals and better utilize building blocks already discovered for exploration of the search space, which in turn improve the GA's performance. In addition, I conduct experiments to investigate effects of two important factors that affect GA's search power: deception and population size.

In Chapter 4, I employ a Markov chain model to analyze GAs as a theoretical basis for the mate selection schemes, where I introduce an explicit way of describing how an individual's fitness depends on others' fitnesses. Although closed form analysis is difficult in general, useful insights can be obtained by means of both visual and computational exploration of the models. I characterize effects of various factors interacting with mate selection schemes, such as mutation, crossover, and difficulty of test functions. Then I show how different mate selection schemes enhance the GA's search power.

The study of Chapters 3 and 4 presents important properties of mate selection in the context of GA. Since the testbeds used in these two chapters are relatively simple, in Chapter 5, I continue the study based on more complicated test functions for larger

population sizes and string lengths. The investigation first concentrates on several original versions of the Royal Road functions (Mitchell, Forrest, and Holland, 1992), which form a class of idealized building-block-based test functions for comparing effects of different mate selection schemes. I then continue testing the mate selection schemes using the hyperplane-defined functions (Holland, 2000), which extend the complexity of fitness landscapes for further exploration of the GA's behavior.

With the results obtained for the building-block-based testbeds, I introduce an important hypothesis regarding general effects of the mating choices on the GA's search. Chapter 6 then presents an empirical validation based on more realistic, non-building-block-based problems, and the results indeed validate my hypothesis. These results are encouraging since it means that the ideas of mate selection proposed in this thesis can be applied to practical problems.

In Chapter 7, I discuss a more general setting in the context of multimodal function optimization, engineering and machine learning. Identifying multiple peaks and maintaining subpopulations of the search space are two central themes. An immune system model is employed to study these two problems. The experimental results then shed more light into how the mate selection schemes compare to traditional selection schemes.

Finally, Chapter 8 provides a summary of this thesis, and highlights its contributions to GA research. It discusses paths for future research, and draws overall conclusions from the research presented in this thesis.

CHAPTER 2

MOTIVATION

2.1 Introduction

A Simple GA (SGA) traditionally generates a random, initial population of candidate solutions (chromosomes). Selection for reproduction, crossover and mutation operate on the population over a certain number of generations until a stop criterion is reached. The probability of individuals being selected for reproduction is according to their fitness values: better solutions have larger probability to be chosen to cross with other solutions and generate offspring that share the genetic material from both parents. Mutations may occur with very low probability. If there is no specific restriction on how mating partners should be chosen, an individual mates with any other regardless of its parenthood or likeness. This type of mating scheme is referred to as random mating (Roughgarden, 1979; Russel, 1998).

Random mating is the simplest form of mating process used by GAs. The selection scheme acts on the population to pick two parents for producing offspring, each parent being *passively* assigned its mating partner. Such a simple implementation in fact overlooks a potential advantage in the GA's mating process—the design of robust GAs is contingent not only upon how they exchange information (the task of crossover), but also upon their selection of proper mates (the task of mate selection). It may be quite beneficial to seek a mate by active search, rather than by happenstance; once a good mate is found, the information exchange and combination may be facilitated

to improve performance substantially. Let us consider the following example:

Given a problem to be solved, if a bit string representation for candidate solutions is used, a simple GA works as follows (Mitchell, 1996):

1. Start with a randomly generated population of n l -bit chromosomes (candidate solutions to a problem).
2. Calculate the fitness $f(x)$ of each chromosome x in the population.
3. Repeat the following steps until n offspring have been created.
 - a. Select a pair of parent chromosomes from the current population, the probability of selection being an increasing function of fitness. Selection is done “with replacement,” meaning that the same chromosome can be selected more than once to become a parent.
 - b. With probability P_c (the “crossover probability” or “crossover rate”), cross over the pair at a randomly chosen point (chosen with uniform probability) to form two offspring. If no crossover takes place, form two offspring that are exact copies of their respective parents.
 - c. Mutate the two offspring at each locus with probability P_m (the “mutation probability” or “mutation rate”), and place the resulting chromosomes in the new population. If n is odd, one new population member can be discarded at random.
4. Replace the current population with the new population.
5. Go to step 2.

In this simple GA implementation, many selection schemes can be used to fulfill the selection process. For example, if we adopt the fitness-proportionate selection

scheme (Holland, 1975), a pair of parent chromosomes are selected from the population, the probability of selection being an increasing function of fitness. Therefore the first individual is *passively* assigned mating partners by fitness-proportionate selection. The consequence is that each individual is forced to mate with some partners that may not carry potential genetic material needed for further improvements. This often impedes the power of information exchange through individuals of GAs. The following example helps further elucidate this point:

Suppose the population is composed of bit-strings of length 8 and the relevant building blocks are 1111**** and ****1111 (* can be either 1 or 0), and each of them contributes fitness of 4 to the strings in the population. Then, for example, a string X , 11110000, is of fitness 4, and the optimal string is 11111111, whose fitness is 8. Now given string X , and two candidate mating partners, Y_1 (11110000) and Y_2 (00001111), under the mechanics of the simple GA above, Y_1 and Y_2 have the same probability to be chosen for mating since these two strings are of the same fitness. However, if we are concerned with finding the optimal string, clearly, string Y_2 is better than Y_1 because the mating between Y_2 and X is likely to generate the optimum, yet it is not the case if X mates with Y_1 . This implies that the fitness-proportionate selection scheme is not able to distinguish individuals of the same fitness, yet of quite different string structures.

The example above shows a possible deficiency of employing a simple selection scheme that does not permit individuals to *actively* determine their mates. Such an implementation of the selection process confers the only selection pressure in simple GAs, where population members' fitnesses are independent of others in the sense that their fitnesses are determined by the environment only, rather than by other individuals. However, allowing individuals themselves to actively choose mates introduces another source of selection pressure. The fitnesses of candidate mates are somewhat "re-defined" according to the degree of their satisfying mating preference

of the first individual. As a result, the fitness values of population members can be interdependent and coevolve with each other.

In the rest of this chapter, biological background of non-random mating will be discussed, and I will present a literature review of prior GA research work on this subject. Then starting from the next chapter, I will show the framework for investigating non-random mating schemes in the context of GA, and present the relevant theoretical and empirical analyses.

2.2 Biological Background

One of the reasons population genetics is a successful theory is that it is built upon a null model, the Hardy-Weinberg equilibrium principle (Freeman and Herron, 1998, p. 121). This null model predicts, under certain assumptions (i.e., population experiencing no selection, no mutation, no migration, no genetic drift and random mating), that across generations allele and genotype frequencies will not change. The Population will not evolve. Violations of these simple assumptions of the null model can result in change of allele frequencies. Population genetics thus identifies the mechanism of population evolution.

The assumption of the Hardy-Weinberg Law that members of opposite sexes in populations mate randomly ensures that the population equilibrium is not disturbed. This case is referred to as *panmixia* in which each individual member of a population has an equal chance of mating with every other individual of the opposite sex. The probability of mating is not determined by genotype; each individual mates without preference. However, any degree of preferential matings between individuals that results in particular genotypes mating more frequently than would be expected at random constitutes non-random mating. When non-random mating occurs, genotypes will not occur in frequencies predicted by the Hardy-Weinberg equilibrium principle.

Sexual selection is typical of non-random mating, where members of one sex show

a consistent preference for a particular *phenotype* of the opposite sex (Price, 1996; Freeman and Herron, 1998). It normally results in strongly skewed mating systems, because panmixis is lost and often a small number of individuals are responsible for much of the reproduction in a population.

Another representative non-random mating mechanism is assortative mating (Price, 1996; Freeman and Herron, 1998). Assortative mating means mating within subgroups of a population characterized by genetic similarities between mates. For example, organisms with similar genetically controlled heat requirements tend to become sexually active at the same time and mate with each other more frequently than with individuals with different heat requirements.

Assortative mating can be generally classified as four types: *positive assortative mating*, *negative assortative mating*, *inbreeding*, and *outbreeding*. Positive assortative mating indicates that individuals preferentially mate with their own phenotype so as to decrease heterozygotic frequency (increases homozygotic frequency). This is common in humans; for example, tall men and tall women tend to marry. Negative assortative mating, on the contrary, means that individuals avoid their own phenotype and the heterozygotic frequency is increased. For example, if tall persons select short persons to marry this would represent negative assortative mating.

The third type is *inbreeding* or, more commonly, *incest* (Roughgarden, 1979; Russell, 1998): close relatives have an increased probability of mating with each other, and unrelated individuals have a decreased probability of mating. *Outbreeding* refers to less mating between relatives than occurs in random mating. Inbreeding increases the proportion of homozygous individuals in a population at the expense of heterozygotes, thereby resulting in decrease of the genetic diversity in the population. Outbreeding, on the contrary, increases that same diversity (Roughgarden, 1979). In case of inbreeding, genetic disorders are often recessive and, oftentimes, increase in frequency with increase in homozygous recessives. *Inbreeding depression* is the reduc-

tion in mean fitness of a population from inbreeding resulting in homozygotes with deleterious or lethal recessive alleles. This is a significant problem in zoos and captive breeding programs, as well as for endangered species such as the California condor.

Biological background of non-random mating provides significant insights for deeper understanding of the role of sex in evolution. In prior GA research, several models have been proposed that implement the idea of non-random mating. I will present a literature review of representative work on non-random mating in the context of GA in the next section.

2.3 Non-Random Mating in Prior GA Research

In prior GA research, non-random mating (restricted mating) schemes have been proposed for achieving different goals. For example, the objectives may be to directly promote population diversity, to prevent the formation of lethal solutions, or to induce speciation.

Diversity, as implemented in the context of GA, plays at least two key roles: (1) the formation and maintenance of diverse building blocks on the way to a single, satisfying solution, and (2) the formation and maintenance of diverse multiple solutions. The two roles are not entirely independent. In searching the path to a single solution, the GA may encounter multiple solutions. Similarly, maintaining multiple solutions may be applicable, in some form, to exploitation of beneficial population diversity to construct a single desired solution.

Promoting population diversity for improving GA's performance of locating a single desired solution (e.g., the global optimum or the best-so-far individual in the population) is a common practice. Booker (1987) proposed crossover among reduced surrogates—nonmatching alleles of the strings being crossed. If the population individuals differ in more than one bit, crossover is guaranteed to produce offspring different from their parents. The idea is to restrict mating between individuals that are

too similar and to prevent fixation within classes rather than between classes. In (Eshelman and Schaffer, 1991) *incest prevention* is another non-random mating scheme, where mating is allowed only between dissimilar individuals; i.e., if strings' Hamming distances exceed a certain threshold, they are permitted to mate. (Craighurst and Martin, 1995) also proposed a method for incest prevention in which recombination between individuals with a certain degree of shared parenthood is not permitted. The authors defined an incest prevention degree, which designates how far back in the family tree the GA must look in order to prevent recombination between two related individuals. Fernandes, Tavares, Munteanu, and Rosa (2001) proposed the *negative assortative mating* GA that chooses one parent (the first parent) and a set A of individuals by regular selection schemes (e.g., fitness proportionate selection). Then the second parent will be the one, belonging to set A , that has the maximum Hamming distance to the first parent. If several second parents are of the same Hamming distance, the one with higher fitness is selected. The GAs with these “dissimilarity-based” mate selection schemes all demonstrated improved performance when the goal is to locate a single desired solution.

Lethal solutions are the low-fitness offspring that may be generated from crosses between multiple conflicting solutions. For example, consider a function with two equal peaks: $f(x) = (x - (\frac{1}{2}))^2$, where 0 and 1 are the two optima. In many encodings, 00...0 and 11...1 may represent these two optima. The crossing from solutions at these two optima is rather likely to create useless hybrids that degrade certain online performance of GA. Resolving this problem usually involves a certain degree of restriction on mating, because the formation of lethal solutions can be avoided. Deb and Goldberg (1989) restricted mating by allowing an individual to search for a mate within a distance of σ_{mating} . If such an individual can be found, mating is performed. Otherwise, a random individual is chosen. They showed that restricting mating to similar individuals produces more consistent results across multiple runs,

and improves average population fitness.

The third objective related to non-random mating is speciation. A species, according to the biological definition, is a class of organisms that are capable of interbreeding among themselves, but that typically do not breed with individuals outside their class (Cook, 1991). This notion of *reproductive isolation* has led some GA researchers to attempt to induce speciation by preventing mating among dissimilar individuals.

Booker (1982, 1985) restricted crossovers to occur between functionally similar individuals. The individuals were rules in a Holland-style classifier system, in which the notion of “functionally similar” has a natural interpretation as the set of classifiers that simultaneously match the message list. Only classifiers that match (or partially match) the same message are allowed to cross. It was shown that this approach works well for forming sub-populations in the context of classifier systems, but was not examined in the more general GA context. In addition, his additional use of a sharing scheme in selection raises the possibility that the combination of sharing and mating restriction, or sharing alone, rather than mating restriction alone, is responsible for the success.

Booker (1982) and Goldberg (1989) explored various approaches in which a mating tag is attached to each individual. This tag must match another individual in some number of loci before a cross is permitted. Many variations exist, including one-way matching, two-way matching, and partial matching. More advanced methods add a template to each individual, and matches must occur between tags and templates instead of between tags and individuals. Tags and templates evolve, along with the rest of a string. Deb applied evolving species tags and templates to restrict mating in multimodal function optimization (Deb and Goldberg, 1989). Two individuals cross if their tags and templates match both ways; if the matching is only one way, they cross with probability 0.5.

Spears (1994) employed a k-bit tag attached to each population individual, which

effectively divides the population into 2^k subpopulations. He then disallowed mating between individuals with different tags. Tags are never modified by mutation and crossover, yet selection is allowed to operate across subpopulations. The results he obtained showed that on two sinusoidal test functions, each having five peaks of equal height, the algorithm maintains a few of the peaks. However, if peaks are of different heights, all subpopulations eventually converge to the highest peak.

Todd and Miller (1991) used mating-preference tag to induce sexual selection of individuals and demonstrated that sympatric speciation can be formed in their GA's framework. The tag decodes to a real number on a given interval that indicates an individual's preferred mating distance. The partial probability that a given individual mates with another individual is determined by a triangular function. If two individuals' partial probabilities are calculated, the probability that they actually mate is the product of their individual, partial probabilities.

With a specific selection-for-reproduction scheme, Todd and Miller showed that, on a flat fitness function, over the course of the run, the resulting population is dynamically divided into several clusters of phenotypically similar individuals: some subpopulations merge and some split into sub-clusters. However, on non-flat fitness functions, selection pressure and noise become dominant, and the GA does not generally achieve speciation.

Mahfoud (1995) proposed a scheme that restricts mating to take place only within a species; if two individuals are from different species, the crossing is not permitted, and the individuals proceed to the mutation stage. His implementation used a phenotypic distance threshold of 0.1 to decide whether two individuals belong to the same species. It turned out that runs on a sinusoidal testbed that has five peaks, spaced at intervals of 0.2, without mating restriction, the GA fully converges by generation 40 to a single global optimum from the five possibilities. (All 100 individuals become identical.) With mating restriction, due to the reduced number of crosses, the GA

fully converges by generation 30 to a single global optimum. Mating restriction, in this case, accelerates convergence to a single peak, rather than distributing population on several peaks—speciation does not occur through this intraspecies mating restriction.

In conclusion, the prior existing work reveals that dissimilarity-based mating restriction seemingly facilitates the search for a single, satisfactory solution. Crosses between species appear to be desirable for potential jumps to higher peaks in the search space. However, to avoid producing lethal offspring, it would require that matings occur only between individuals of similar characteristics.

Speciation is a much more complicated issue. If selection pressure dominates the evolution process, similarity-based mating restriction alone generally does not guarantee effective speciation. The result Mahfoud obtained based on phenotypic similarity is a clear example. Several aforementioned similarity-based mating restrictions, including Spears’ tag-based restriction and Todd and Miller’s sexual selection (on non-flat fitness functions), are not successful in formation of species, either. To facilitate speciation would require additional strategies to control selection pressure. For instance, Deb and Goldberg (1989) employed the fitness sharing scheme to regulate fitnesses of population members based on the information available for niches in question. The results showed that their approach indeed promotes speciation and prevents convergence on a single peak. When similarity-based mating restriction was applied, the resulting speciation becomes more consistent and stable.

We may put together the observations above and propose two claims as follows:

- Dissimilarity-based mating selection accelerates exploration of the search space.
- Similarity-based mating selection enhances the capability of speciation mechanism for formation and maintenance of subpopulations.

It indicates, aiming at *rapid* formation and *stable* maintenance of a single solu-

tion or multiple solutions, that there exists a delicate balance between mating like and unlike individuals. Both interspecies and intraspecies crosses may be beneficial, depending on the problems at hand, the algorithms used, and the objectives.

Given all the results obtained so far in the context of GA, a systematic investigation on similarity and dissimilarity-based mate selection is still lacking. In this thesis, I intend to propose a framework that allows us to conduct comprehensive analyses to validate (or invalidate) the claims above and deepen the understanding for GA's behavior. The following chapters present such analyses.

CHAPTER 3

METHODOLOGIES AND GENERAL ANALYSIS

3.1 Introduction

Traditional GA models focus on problems in which each population member's fitness is independent of other population members—their fitnesses are evaluated by an exogenously imposed objective function. In nature organisms may interact and coevolve with other organisms in an environment. The fitnesses of organisms are thus evaluated not only by the environment but also by the individuals with whom they interact.

Several authors have studied GA models in which the fitness evaluation of individuals depends on other individuals in the same population or in other populations. Some of the results reported show that such models yield better performance than traditional optimization models (Hillis, 1992; Rosin and Belew, 1997; Vafaie and De Jong, 1996; Potter et al., 1995).

In this thesis, I introduce simple models that implement mate selection in the context of GA, based on the idea of assortative mating in biology. Allowing individuals to actively search for mates is an approach to model interdependent fitnesses of population members. For instance, an individual may first choose a set of candidate mating partners. Then the probability of these candidates being selected as the actual mate may depend on the degree of their satisfying the first individual's mating preference. This method can be regarded as re-defining the fitnesses of these candidates based on

the degree of satisfying mating preference, and the probability of being selected is an increasing function of the newly defined fitnesses. As a result, through endogenous interactions in the population, the realized fitness of an individual depends on both the environment and how many others would consider it a good mate.

In the following parts of this chapter, I first use the traditional fitness proportionate selection scheme to introduce two special cases of mating choices to facilitate a theoretical analysis based on the Schema theorem. Although they represent two opposite extremes, it turns out that such an analysis can be applied to the investigation for some well known phenomena in biology and in the context of GA, such as hitchhiking and the founder effect. These two phenomena have been identified as important factors that impede the GA's search power. The experimental results will show that the dissimilar mating choice effectively suppresses these two phenomena.

Afterwards, I extend the range of mate selection to include two intermediate cases that combine similarity test and fitness-proportionate selection to help discriminate candidate mates. Therein it will be more clearly seen that the individuals' probabilities of being selected for mating are complicated by the environments and mate preferences. I then study the GA's performance in terms of "rate of improvement" and "creation of lethal hybrids." The results show that the dissimilar mating scheme furthers the GA's exploration of the search space and yields a better best-so-far performance, yet at the expense of generating more lethal hybrids that degrade the population's fitness.

Then I present some results based on a microscopic diversity measure to consolidate our understanding gained from the investigation of hitchhiking and the founder effect, where it is shown that the dissimilar mating preferences retain more genetic variation in the population.

Thereafter, I adopt tournament selection as the background selection scheme since the experiments I conducted show that the tournament-based selection schemes tend

to outperform the proportion-based selection schemes. I will present the relevant experimental results which show the superior performance of the dissimilar mating schemes in terms of improving the best-so-far solutions.

In the final part of this chapter, I investigate the effects of population size on the GA's search power. The results show that for limited population sizes, the dissimilarity-based mate selection schemes can improve the GA's best-so-far performance; however, this performance advantage will be suppressed when population size is large enough. Thus, we learn that in many practical problems where population size is limited the dissimilarity-based mate selection schemes are better strategies for the GA to improve the best-so-far performance.

3.2 Mate Selection Schemes

Many approaches to implement the mechanism of mate selection are possible. For example, the first individual may choose a set of members in the population, and then select the one who most matches its preference. If we adopt the Hamming distance as the similarity metric, then, in case of similar mating, the first individual may select as the actual mate the one whose Hamming distance to him is the smallest. In case of dissimilar mating, the one whose Hamming distance is the largest will be selected. Such mate selection schemes are deterministic in the sense that the individual who is finally being selected is the most similar (or dissimilar) to the first individual. It is quite natural to implement a probabilistic mate selection strategy. For example, the probability of candidates being selected as the actual mate may be proportional to the magnitude of their Hamming distances.

In this thesis, I replace “positive assortative mating” used in biology with “similar mating.” Likewise, “negative assortative mating” is replaced with “dissimilar mating.” Then several mate selection schemes are proposed to facilitate the investigation on the effects of mate choices.

I first examine two extreme cases: in case of similar mating, the population member that is the most similar to the first individual is chosen as the mate; in case of dissimilar mating, the most dissimilar individual is chosen.

The similarity measure between two individuals (bit strings) used here is Hamming distance—the number of locations at which corresponding bits differ.* This mate selection scheme works as follows:

During each mating event, a fitness-proportionate selection is run to pick out the first individual. Then the Hamming distances of all population members to the first individual are calculated. The actual mate of the first individual is chosen according to the following two different schemes:

Maximum Similar Mating (MSM): The population member whose Hamming distance is the smallest is selected for mating.

Maximum Dissimilar Mating (MDM): The population member whose Hamming distance is the largest is selected for mating.

Notice that in the mate selection schemes above if several members are of the same maximum (or minimum) Hamming distance to the first individual, then one of them is randomly selected. The computational cost involved in a generation is $O(N^2)$ similarity comparisons, where N is the population size.

These two extreme cases simplify the analysis of mate selection based on the Schema Theorem (Holland, 1975). This is the subject of the next section, where I perform a simple theoretical analysis and present empirical results to enhance the investigation.

*There are several ways to define the similarity between individuals (see the literature review presented in Section 2.3). In this thesis, I focus on the Hamming distance, which is sufficient for the study of the mate selection schemes proposed here.

3.3 Schema Theorem Analysis

3.3.1 Review of The Schema Theorem

A common assumption in the traditional GA theory is that, at a very general level of description, GAs work by discovering, emphasizing, and recombining good “building blocks” of solutions in a parallel fashion. The idea behind this assumption is that good solutions tend to be composed of promising building blocks.

The notion of building blocks was formalized by Holland (1975) through so-called schemata—A schema is a set of bit strings that can be described by a template composed of ones, zeros, and asterisks, the asterisks representing wild cards. For example, the schema $H = 0 * * * 1$ is the representation of a set of all 5-bit strings which begin with 0 and end with 1.

The investigation on the increase and decrease of the number of schema instances over generations makes clear how GAs process sub-solutions to improve the search quality. Holland’s schema theorem, which describes such a dynamic behavior, can be derived as follows:

Let $m(H, t)$ be the number of instances of schema H present in the population at generation t . We calculate the expected number of instances of H at the next generation, or $E(m(H, t+1))$, in terms of $m(H, t)$. By fitness-proportionate selection, the simple GA (as described in Chapter 2) assigns a string a probability of selection directly proportional to fitness. Thus the expected number of offspring of a string x is equal to $f(x)/\bar{f}(t)$, where $f(x)$ is the fitness of x and $\bar{f}(t)$ is the average population fitness at time t . It follows that H can expect to be selected $m(H, t) \cdot (f(H, t)/\bar{f}(t))$ times, where $f(H, t)$ is the average fitness of those strings in the population that are instances of H at time t . The probability that single-point crossover destroys a schema is precisely the probability that the crossover point falls within the schema’s

defining positions. The probability that H survives crossover is:

$$S_c(H) \geq 1 - p_c \left(\frac{d(H)}{l-1} \right),$$

where p_c is the crossover probability, $d(H)$ is the defining length of H (i.e., the distance between its outermost defined bits) and l is the length of bit strings. Since a disrupted schema may regain its composition if two similar strings cross with each other, this survival probability is an inequality.

The probability that H survives mutation is $(1 - p_m)^{o(H)}$, where p_m is the mutation probability and $o(H)$ is the order of H (i.e., the number of defined bits in H). The product of the expected number of selections and the survival probabilities yields the schema theorem:

$$E(m(H, t+1)) \geq m(H, t) \cdot (f(H, t) / \bar{f}(t)) \cdot [1 - p_c \left(\frac{d(H)}{l-1} \right)] \cdot [(1 - p_m)^{o(H)}]. \quad (3.1)$$

The schema theorem describes the growth of a schema over generations. It is often interpreted as implying that short, low-order, above-average (in fitness) schemata grow exponentially over time, while below-average schemata decay at a similar rate. Therefore, the simple GA, prior to significant convergence, allocates an exponentially increasing number of trials to promising schemata or building blocks.

Notice that the schema theorem is often predictive of schema growth and describes the disruptive aspects of crossover and mutation; it is not used to account for how new schemata are discovered. New schemata can be constructed by crossover and mutation to facilitate the exploration of the search space. In general, as a GA progresses, population members are more and more alike, so that schemata which are disrupted by crossover tend to be regained immediately. On the other hand, as long as the population undergoes significant convergence, the power of crossover for recombining building blocks is hindered because of the loss of population diversity.

In the next section, I use the schema theorem to investigate how the mate selection schemes influence schema dynamics.

3.3.2 Interpretation of the Effects of Mate Selection By The Schema Theorem

In Section 3.2, two mate selection schemes are proposed: the maximum similar mating, and the maximum dissimilar mating. In the simple GA proposed in Chapter 2, selection is done *with replacement*, meaning that the same individual can be selected more than once to become a mate. This includes the likelihood of mating with the exact same copy of oneself. Since in case of the maximum similar mating an individual chooses the population member who is the most similar to himself, this would guarantee that the first individual always chooses the identical copy of himself for mating.

Thus the probability that schema H survives crossover is:

$$S_c(H) = 1,$$

since a disrupted schema regains its composition immediately after crossover.

Let $E_{MSM}(m(H, t))$ denote the expected number of instances of H at the t th generation based on the maximum similar mating. Then the schema theorem becomes:

$$E_{MSM}(m(H, t + 1)) = m(H, t) \cdot (f(H, t)/\bar{f}(t)) \cdot [(1 - p_m)^{o(H)}]. \quad (3.2)$$

On the other hand, in case of the maximum dissimilar mating an individual chooses the population member whose Hamming distance is the largest. This would give the crossover-surviving probability of schema H a larger likelihood to approach the lower bound $1 - p_c(\frac{d(H)}{l-1})$. Let $E_{MDM}(m(H, t))$ represent the expected number of instances of H at the t th generation based on the maximum dissimilar mating. Then the schema theorem becomes:

$$E_{MDM}(m(H, t + 1)) \geq m(H, t) \cdot (f(H, t)/\bar{f}(t)) \cdot [1 - p_c(\frac{d(H)}{l-1})] \cdot [(1 - p_m)^{o(H)}]. \quad (3.3)$$

The ratio of the expected growth rate of schema H for these two mate selection schemes is given by

$$1 \leq \frac{E_{MSM}(m(H, t + 1))}{E_{MDM}(m(H, t + 1))} \leq \frac{1}{1 - p_c \frac{d(H)}{l-1}}. \quad (3.4)$$

Equation 3.4 shows that the expected schema-growth rate for the maximum similar mating is greater than that for the maximum dissimilar mating.

The rest of this chapter presents empirical results to further compare the similar and dissimilar mating schemes in the context of simple GAs.

3.4 Empirical Analysis

The two cases proposed so far facilitate the theoretical analysis. Although they represent two opposite extremes, it turns out that such an analysis can be applied to the investigation for some well known phenomena in the context of GA—hitchhiking and the founder effect, which have been identified as important factors that affect the GA’s search power. This is the subject of the next two subsections.

Then I will introduce two intermediate cases of mate selection to more clearly show that how the individuals’ probabilities of being selected for mating are complicated by the environments and mating choices. Based on these schemes, I will study the GA’s performance in terms of best-so-far performance, which is a traditional performance metric employed in many practical applications. Thereafter I present some results based on a microscopic diversity measure to consolidate our understanding gained from the investigation of diversity in hitchhiking and the founder effect.

Then I adopt tournament selection as the background selection scheme since the experiments I conducted show that the tournament-based selection scheme tends to outperform the proportion-based selection scheme. I will present the relevant experimental results to show that the dissimilar mating schemes can improve the GA’s best-so-far performance.

In the final part of this chapter, I investigate the effects of population size on the GA’s search power, where we can see how the difference between the mate selection

schemes' power is suppressed when the population sizes are large enough.

In short, the most important lesson we will learn is that dissimilarity-based mate selection is more beneficial in improving the GA's search power.

3.4.1 Hitchhiking

In population genetics “hitchhiking” is a well-known phenomenon that occurs when some newly discovered allele (or sets of alleles) offers great fitness advantages. As that allele spreads quickly through the population, the closely linked alleles (though they may make no contribution to the fitness) could propagate to the next generation by hitchhiking on that allele. The rapid occupancy of those non-relevant alleles thus greatly reduces exploration of alternatives at those loci. They either drown out other already-discovered alleles that are advantageous, or leave no room for not-yet-discovered beneficial alleles.

In GA research, hitchhiking has been identified as a major problem that limits implicit parallelism by reducing the sampling frequency of various building blocks (Das, and Whitley, 1991; Mitchell, Forrest, and Holland, 1992). Forrest et al. (1993) found that if some intermediate stepping stones are much fitter than the primitive components, hitchhiking generates more severe problems that greatly hamper the discovery of some necessary schemata.

To see the hitchhiking phenomenon in the context of GA, let us consider an example function—a small “Royal Road (RR)” function (Forrest and Mitchell, 1993), in which four consecutive building blocks of five ones each are defined. Table 3.1 is the schematic of this function.

This function involves a set of schemata $S = \{s_1, \dots, s_4\}$ and the fitness of a bit string x is defined as

$$F(x) = \sum_{s \in S} c_s \sigma_s(x),$$

where each c_s is a value assigned to the schema s as defined in the table; $\sigma_s(x)$ is

Table 3.1: Small royal road function S_1 .

$s_1 = 11111*****;$ $c_1 = 10$
 $s_2 = *****11111*****;$ $c_2 = 10$
 $s_3 = *****11111*****;$ $c_3 = 10$
 $s_4 = *****11111*****;$ $c_4 = 10$

defined as 1 if x is an instance of s and 0 otherwise. In this function, the fitness of the global optimum string (20 1's) is $10 \times 4 = 40$.

I select this small Royal Road function as a testbed because it belongs to a class of building-block-based functions, in which improvements in the RR domain depend entirely on the discovery and exploitation of building blocks. This would serve as an idealized testbed for us to observe (1) how mate selection facilitates distinguishing individuals that carry necessary building blocks for further improvements (see the example illustrated in Chapter 2), and then (2) how crossover brings these building blocks residing on separate strings into combination on a single string. (The second goal above has been extensively investigated by Mitchell, et al. (1992) and Forrest et al. (1993). This thesis focuses on the study of the first goal.)

We can observe hitchhiking directly by plotting the densities (percentage of the population that are instances) of the relevant schemata over time for the GAs with the maximum similar mating and the maximum dissimilar mating. The experiments performed are based on one-point crossover rate 1, and population size 20 over 200 generations.[†]

To give hitchhiking its easiest test, I first turn off mutation operator, since mutation may destroy hitchhiker(s). Figure 3.1 is a typical run that shows such density dynamics for the GA based on the maximum similar mating. Schema s_3 is found at

[†]The experimental results obtained show that, for this small problem, the small population size (20) serves well for distinguishing the effects of different mating preferences. Therefore, I will use this population size throughout most of this chapter, except that effects of population size will be investigated in the final subsection of this chapter. In Chapter 5, I will present more experimental results for test problems of larger string lengths and population sizes.

the first generation and quickly spread through the population. In the mean time, schema s_4 was rapidly suppressed by the hitchhikers adjacent to s_3 . A closer examination shows that these hitchhikers are 00101. Since mutation is turned off and mating partners are simply the copies of individuals that select them, the population is quickly filled with the clones of the individuals that carry s_3 and its hitchhikers, and finally converges to a single genotype.

In Figure 3.2, the GA was run with mutation being turned on at probability 0.005 and the same random seed. The results show that once s_3 is discovered, its density in the population rapidly rises, and the density of s_4 simultaneously drops to zero in the first several generations. However, with this nonzero mutation rate, s_4 regains its appearance around generation 130, being drowned out again, and then shows up before generation 140.

In addition, we see that a distinct blip in the density of s_2 around generation 50, showing that the decrease of the density of s_3 allows s_2 to appear in the population. But while s_3 rises quickly again, s_2 dies out. It is rediscovered around generation 75, and drowned out again. Until around generation 180, there is the third blip that shows up, still exhibiting the similar situation as the first two blips.

The results of the GAs with mutation being turned on and turned off clearly show that when matings only occur between exact identical individuals, mutation is the only way to get away from hitchhiking.

Now I examine the GA runs based on the maximum dissimilar mating. Figure 3.3 shows the density plots for a typical GA run with zero mutation rates. The appearances and disappearances of blips are more dramatic than Figure 3.1. Unlike Figure 3.1, s_3 has no apparent edge over s_2 and s_4 : the densities of the three schemata never exceed 0.6, leaving enough space for adjacent schemata to reproduce in the population and keep alive for a longer while.

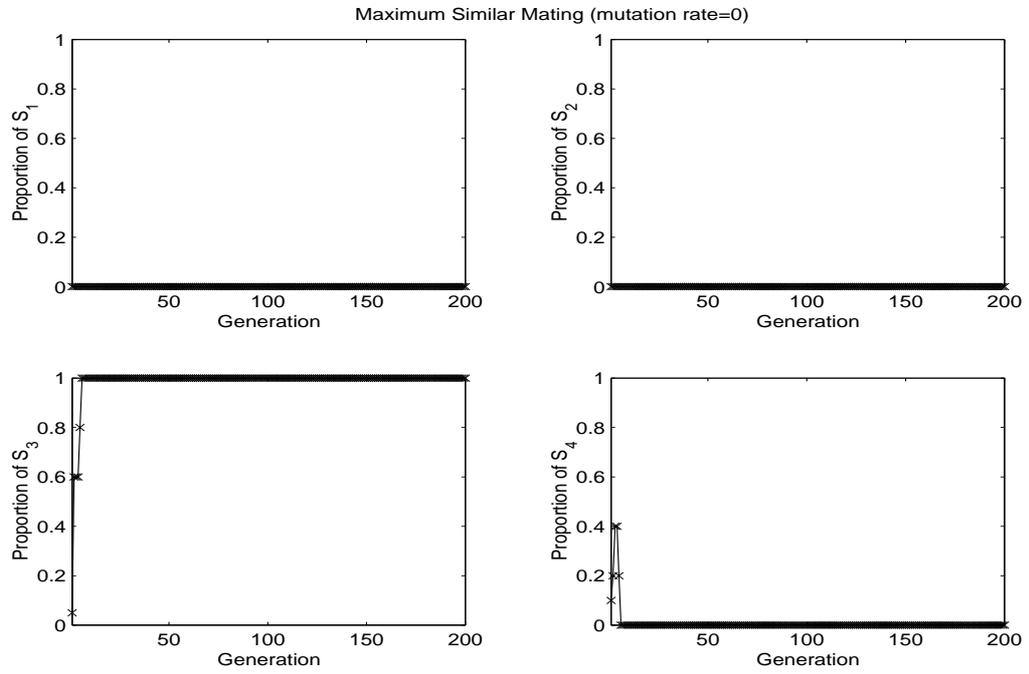


Figure 3.1: Schema dynamics for observing hitchhiking (Maximum Similar Mating).

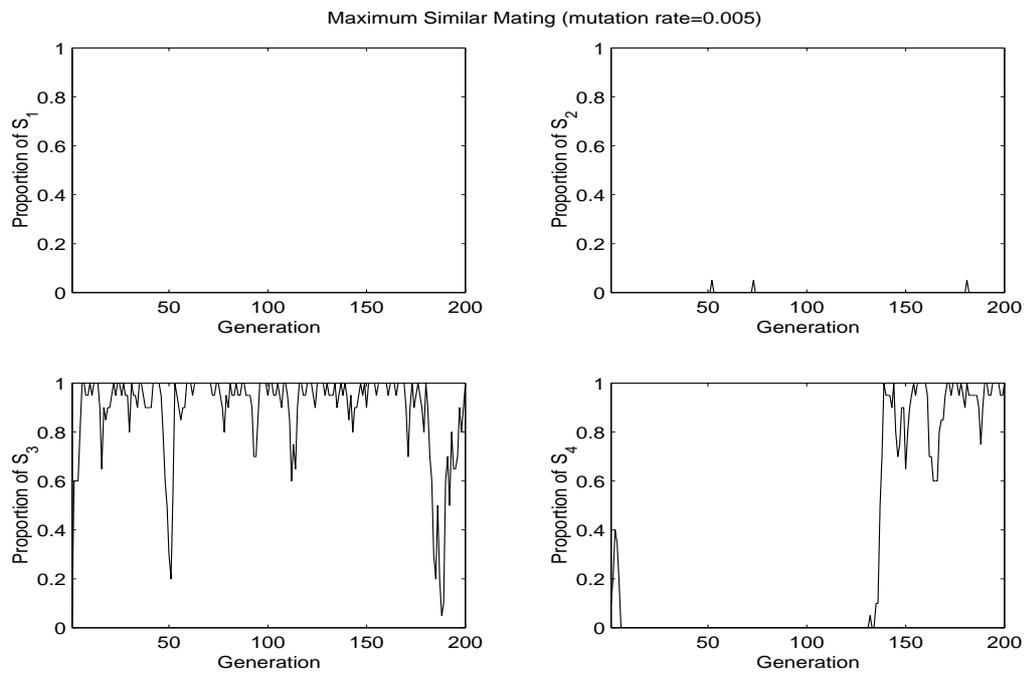


Figure 3.2: Schema dynamics for observing hitchhiking (Maximum Similar Mating).

In addition, the results for the maximum dissimilar mating GAs with mutation rate 0 and 0.005 (Figure 3.4) do not exhibit the clear difference which occurred between the maximum similar mating GAs with mutation off and on.[‡]

The difference between the experimental results obtained for the maximum similar mating and the maximum dissimilar mating is a consequence of different schema growth rates. I show that, in Section 3.2, the schema growth rate for the maximum similar mating is greater than that for the maximum dissimilar mating. As a result, once a schema is discovered, the maximum similar mating guarantees that, with a larger likelihood, instances of that schema take over the entire population in a shorter time than the maximum dissimilar mating. This often entails non-relevant hitchhiking alleles that impede further exploration of alternatives at the hitchhiking loci. Then the density of one or more of the disjoint schemata is seen to reduce significantly. The greater convergence rate resulting from the similar mating scheme therefore can negate progress that the population has made towards good schemata that overlap the hitchhiking bits.

The most likely positions for hitchhikers are those close to the defined positions of good schemata, since they are less likely to be separated from those defined positions under crossover (Forrest and Mitchell, 1993). This implies that the defining bits of those schemata in the highly fit strings, along with the *nearby* bits, quickly come to occupy most of the population. In other words, the nearby bits “hitchhiking” to prominent schemata partially or totally suppress the appearance of disjoint building blocks, leaving few variants at those loci. This leads to the loss of diversity in the vicinity of the better schemata. As a consequence, some of the necessary building

[‡]As I shall show in some of the rest of this chapter, the maximum dissimilar mating scheme, along with crossover turned on, is rather disruptive; thus, although all the schemata have chance to appear, as shown in Figure 3.4, their frequencies cannot approach one. For moderate mating preferences, such as the tournament-based mating preferences used in the later part of this chapter, these four schemata have chance to simultaneously approach to one when the searching process goes on.

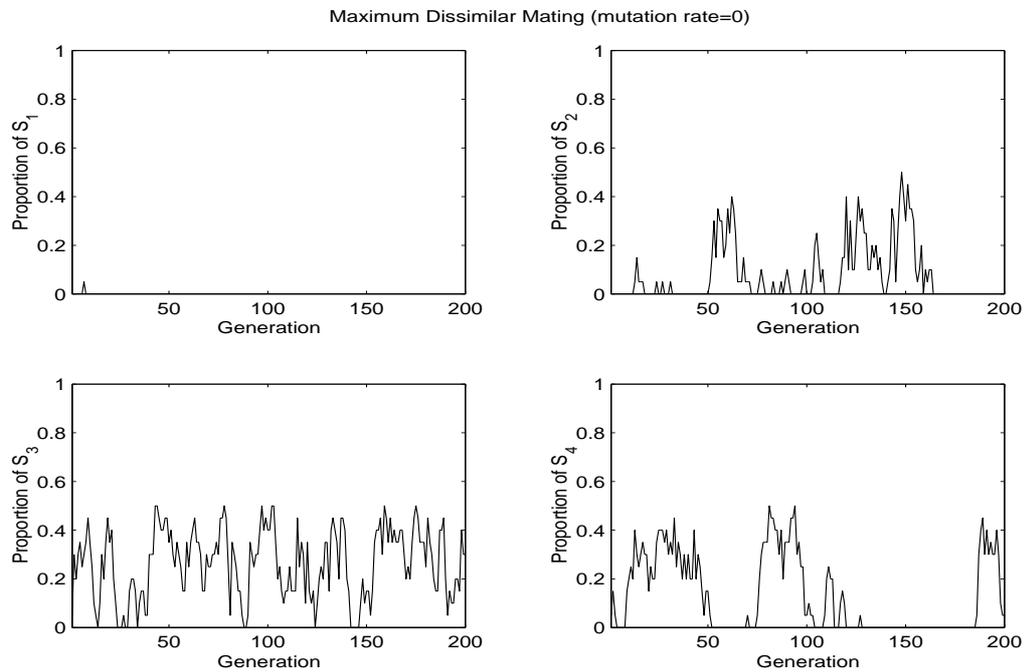


Figure 3.3: Schema dynamics for observing hitchhiking (Maximum Dissimilar Mating).

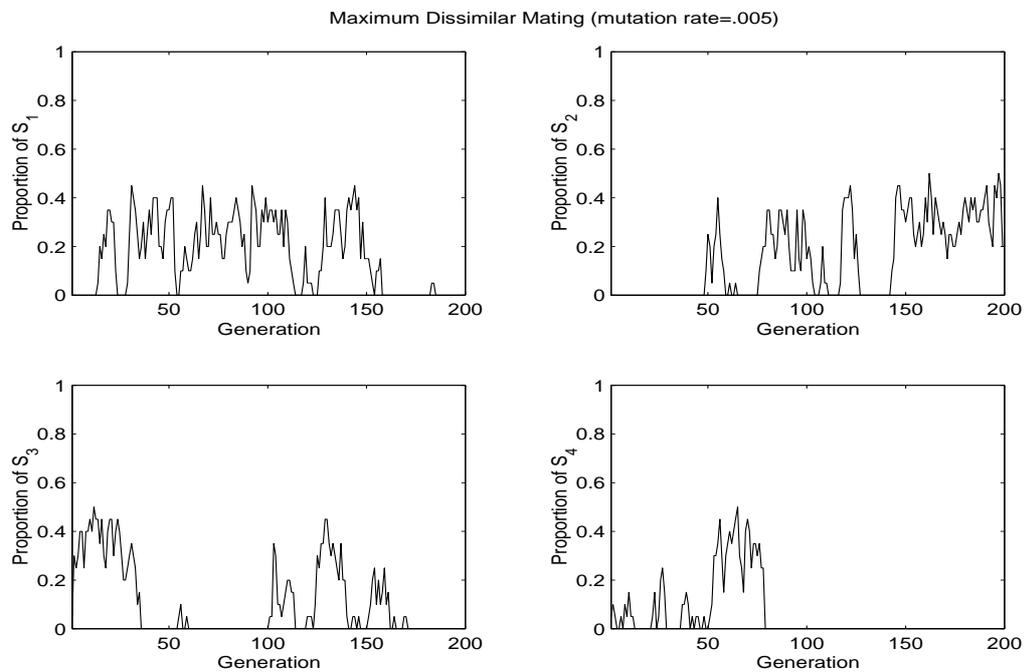


Figure 3.4: Schema dynamics for observing hitchhiking (Maximum Dissimilar Mating).

Table 3.2: Incompatible small royal road function IS_1 .

$$\begin{aligned}
 s_1 &= 11111*****; c_1 = 10 \\
 s_2 &= 00000*****; c_2 = 5 \\
 s_3 &= *****11111*****; c_3 = 10 \\
 s_4 &= *****00000*****; c_4 = 5 \\
 s_5 &= *****11111*****; c_5 = 10 \\
 s_6 &= *****00000*****; c_6 = 5 \\
 s_7 &= *****11111; c_7 = 10 \\
 s_8 &= *****00000; c_8 = 5
 \end{aligned}$$

blocks for crossover to combine to gain performance advancements are lost or unlikely to be discovered, and the GA's search power is greatly hampered.

3.4.2 Founder Effect

In GA research, a much more important constraint on exploration than hitchhiking is the founder effect (Holland, 2000). In presence of incompatible schemata, the first discovered of the incompatible schemata comes to establish a large portion of the population, and constrain the future evolutionary avenue. Consequently, the founder schema effectively precludes the testing of the other incompatible schema. Further improvements stem from the founder, making it progressively less likely that the other schema will influence the search process.

The fitness landscape of the simple royal road function used in the last section consists of four consecutive building blocks. The combination of these building blocks constructs the only path for the GA to improve the search. Let us consider a variant of the simple royal road function that consists of incompatible schemata as shown in Table 3.2.

This function involves mutually exclusive alleles at each schema, and there are 16 (2^4) alternatives for search to improve solution quality. The fitness of the global optimum (20 1's) is 40, and that of the other local optimum (20 0's) is 20.

The incompatibility of schemata on this function is designed for testing the founder effect—I would expect each 5-bit block to be soon occupied by a founder schema, unless the founder schema is destroyed by crossover or mutation.

I examine the founder effect directly by plotting the density of each schema over generations for the GAs with the maximum similar mating and the maximum dissimilar mating. The experiments performed are based on one-point crossover rate 1, and population size 20 over 200 generations.

Analogous to the analysis for the hitchhiking problem, I first turn off mutation in order to give the founder effect its clearest examination, since mutation may destroy founder schemata. Figure 3.5 is a typical run that shows the schema density dynamics for the maximum similar mating GA. Schemata s_1 and s_4 are discovered at the first generation and quickly spread through the population. A closer examination shows that each of s_1 and s_4 has an instance at the first generation. Then because maximally similar individuals are always chosen for mating, which in turns enhances the degree of premature convergence, the instances of s_1 and s_4 quickly take over the whole population. Due to the incompatibility, s_2 and s_3 are prohibited from being tested. In the mean time, along with the growth of schema s_4 hitchhikers 10110 rapidly propagate to the whole population and block the testing of schema s_5 or s_6 (adjacent to s_4 on the right).

These results demonstrate that for this typical run s_1 and s_4 are the first discovered of the two incompatible schemata which rapidly found their dynasties. As a consequence, further improvements are constrained by s_1 and s_4 , and the GA can only devote itself to 4 possible alternatives for further improvements. (For other runs different schemata may take over the whole population. That is, due to sampling errors, either of the incompatible schemata is likely to dominate the population.)

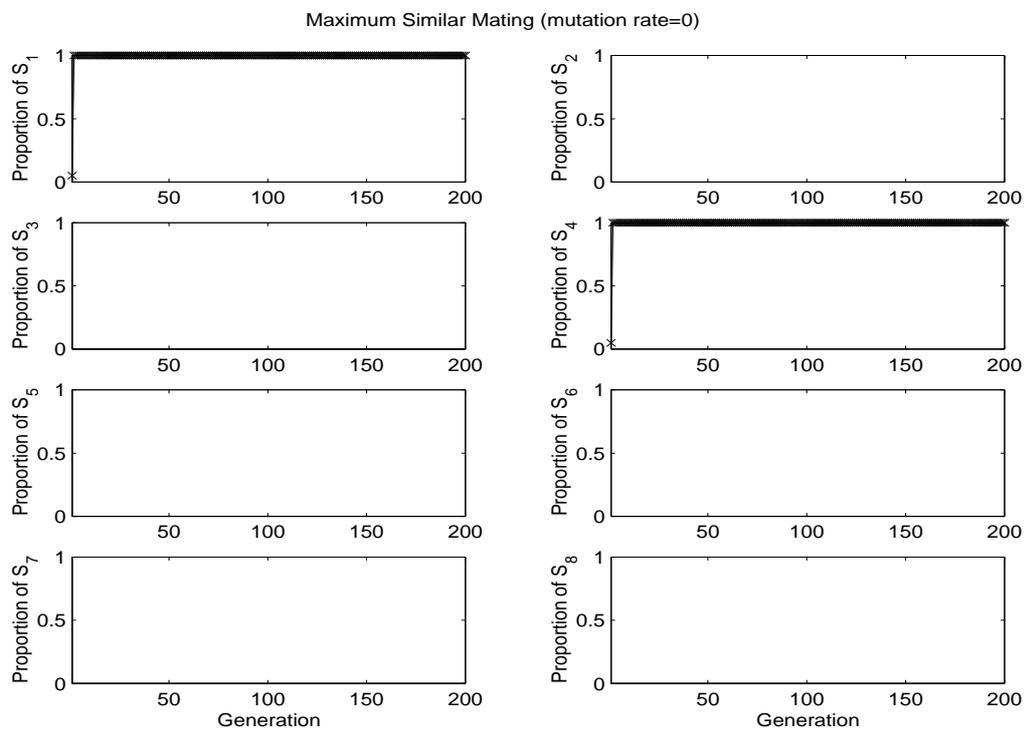


Figure 3.5: Schema dynamics for observing the founder effect (Maximum Similar Mating).

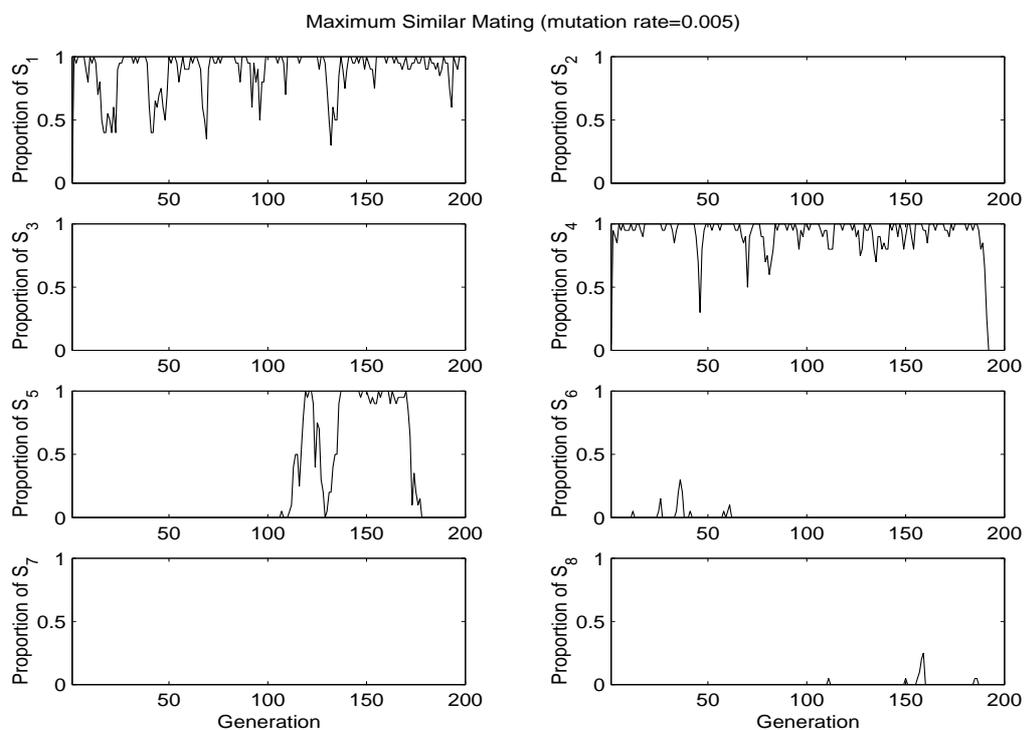


Figure 3.6: Schema dynamics for observing the founder effect (Maximum Similar Mating).

In Figure 3.6, I use the same random seed to run the GA with mutation being turned on at rate 0.005. It exhibits quite different results than the GA with zero mutation rate. s_1 and s_4 now leave some space for the incompatible schemata to show up in the population, yet within two hundred generations, s_2 and s_3 does not appear at all. On the contrary, s_5 does move away from hitchhiking and regain its appearance after generation 100. Since the hitchhikers on the right of s_4 are 10110, only two mutations are needed to discover s_5 , but this is not the case for the founder effect. To get away from the founder effect, it would require five simultaneous mutations together to recover an incompatible schema. Hence the difficulty imposed by the founder effect is more severe than hitchhiking in this case. In addition, hitchhiking is still another problem that affects the search for s_2 and s_3 . Along with the founder effect, these two schemata are thus more difficult to be discovered than s_5 (or s_6).

For the maximum dissimilar mating, the density plots for a typical GA run with mutation being turned off is shown in Figure 3.7. Unlike Figure 3.5, no schema effectively founds a dynasty; as can be seen, the maximum density of each schema is around 0.5, making it less likely that either of incompatible schemata precludes the testing of other incompatible schema (note that, due to sampling error, s_1 , s_4 , and s_5 still do not show up in the population). This situation is made more obvious in the results shown in Figure 3.8, which are obtained for the GA run with mutation rate 0.005 under the same random seed. It is now clear that no schema has an apparent edge over its counterpart; each schema gains considerable proportion in the population.[§]

The distinction between the two selection schemes comes from the fact that the maximum similar mating ensures that individuals always mate with their identical copies. In other words, they prevent themselves from mating with other individuals and thus preclude greater allelic diversity carried by other individuals. On the

[§]Again, as mentioned in the previous subsection, the maximum dissimilar mating scheme, along with crossover, is rather disruptive; thus, although either of the incompatible schemata has chance to appear, its frequency cannot approach one. For moderate mating preferences, such as the tournament-based mating preferences used in the later part of this chapter, the schemata can approach to one when the searching process goes on.

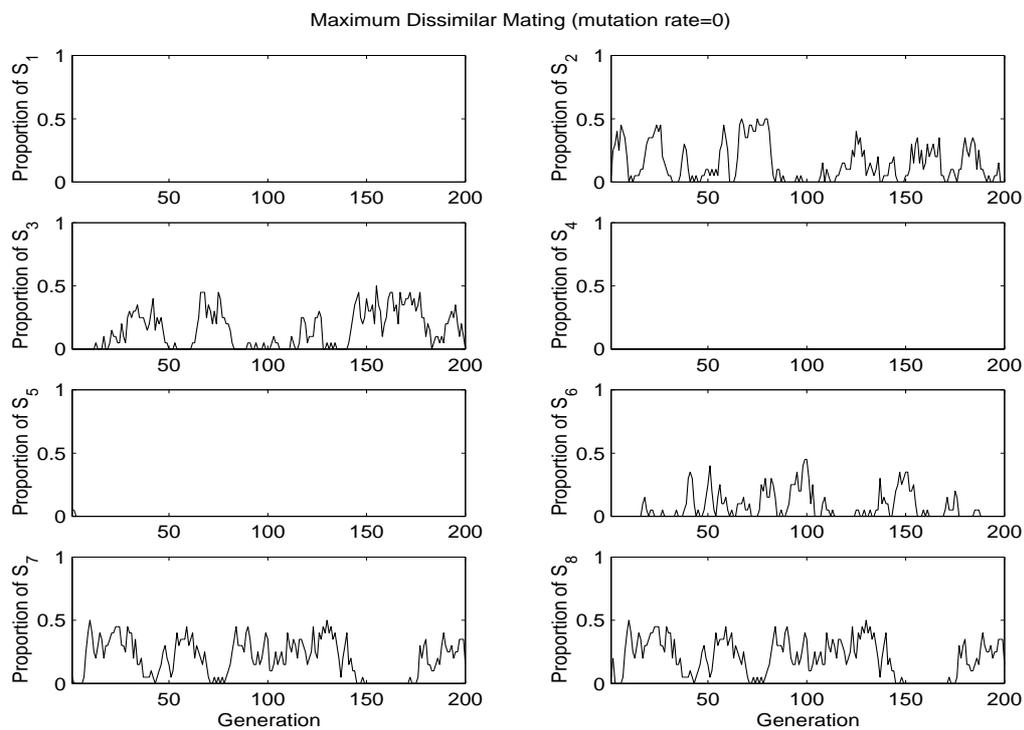


Figure 3.7: Schema dynamics for observing the founder effect (Maximum Dissimilar Mating).

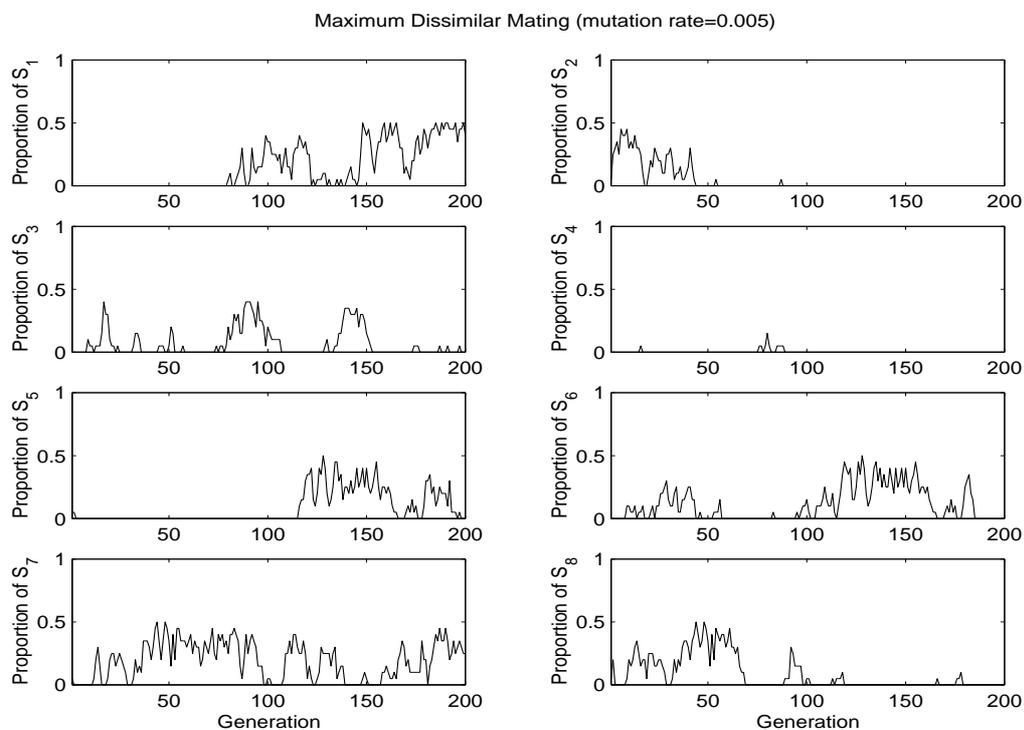


Figure 3.8: Schema dynamics for observing the founder effect (Maximum Dissimilar Mating).

contrary, the maximum dissimilar mating allows individuals to exchange genetic material with dissimilar mates, thereby recruiting more distinct alleles to escape from the founder effect and hitchhiking.

3.4.3 Rate of Improvement and Creation of Lethal Hybrids

How rapid is evolutionary change, and what determines the rates, patterns, and causes of change, or lack thereof? Answers to these questions can tell us much about the evolutionary process. The study of evolutionary rate in the GA context usually involves defining a performance measure that embodies the idea of rate of improvement, so that its change over time can be monitored for investigation.

In many practical problems, a traditional performance metric is the “best-so-far” curve that plots the fitness of the best individual that has been seen thus far by generation n . On the other hand, with an idea borrowed from control systems—*rise time*, which can be defined as the time required to reach some predetermined threshold, we can compare how long different GAs would have to run on average before a given metric first encounters the threshold. For example, in the simple royal road function S_1 , the threshold can be defined as fitness value 30 (3/4 of the maximum fitness) if the we are concerned with the best-so-far. We may also choose a threshold value at 10 if the mean population fitness is the chosen performance metric.[¶]

Since the two mate selection schemes discussed so far are two extreme cases where only the maximally similar or dissimilar population member is chosen as the mate, I introduce another two intermediate mate selection schemes before studying the rate of improvement. That is, during each mating event, the first individual is still

[¶]There are other performance measures available, of course. For example, one may be interested in the relative time taken by an algorithm to perform computations other than fitness evaluations. See the performance indexes used in the Second International Contest on Evolutionary Optimization held in the IEEE-ICEC 97 conference (ICEO, 1997), where the two performance indexes used, the best value reached (BV) and the Expected Number of Evaluations per Success (ENES), correspond to the best-so-far and the rise time performance measures I used, respectively. Which metric should be used depends on the problem domain, the algorithm used and the computation resources. See Section 6.1 for a more detailed discussion on this subject.

picked by fitness-proportionate selection, and the Hamming distances of all population members to this individual are calculated. Then the mate for the first individual is chosen according to the following two schemes:

Proportional Similar Mating (PSM): The probabilities of population members being selected are *reversely* proportional to their Hamming distances.

Proportional Dissimilar Mating (PDM): The probabilities of population members being selected are proportional to their Hamming distances.

Notice that since individuals' Hamming distances may be zero, to avoid dividing by zero in the proportional similar mating I offset all the Hamming distances by 1 in this thesis. (Other offset values are possible, of course.)

The experiments performed here are still based on one-point crossover rate 1, mutation rate 0.005, and population size 20 over 50 runs. Figure 3.9 shows the averaged best-so-far curves on the test function S_1 for the four mate selection strategies.^{||}

The threshold I choose for both test functions (S_1 and IS_1) is 25. We see the maximum dissimilar mating results in better improvement than the other three, because its rise time to the threshold (around 130 generations) is the least. On the contrary, the maximum similar mating yields the worst result. The two intermediate cases exhibit the trend that the dissimilar mating is advantageous for further improvements of the best-so-far performance.

The similar result can be observed in Figure 3.10 where the averaged best-so-far curves (over 50 runs) for the test function IS_1 is shown.

With the observation that the dissimilar mating outperforms the similar mating in the best-so-far curves, it seems reasonable to ask if the whole population converges

^{||}The vertical bars overlaying the metric curves throughout this thesis represent the 95-percent confidence intervals calculated from Student's t -statistic (Miller, 1986). In case of the averaged best-so-fars, the 95-percent confidence intervals represent the intervals within which the actual mean best-so-fars would reside with probability 0.95.

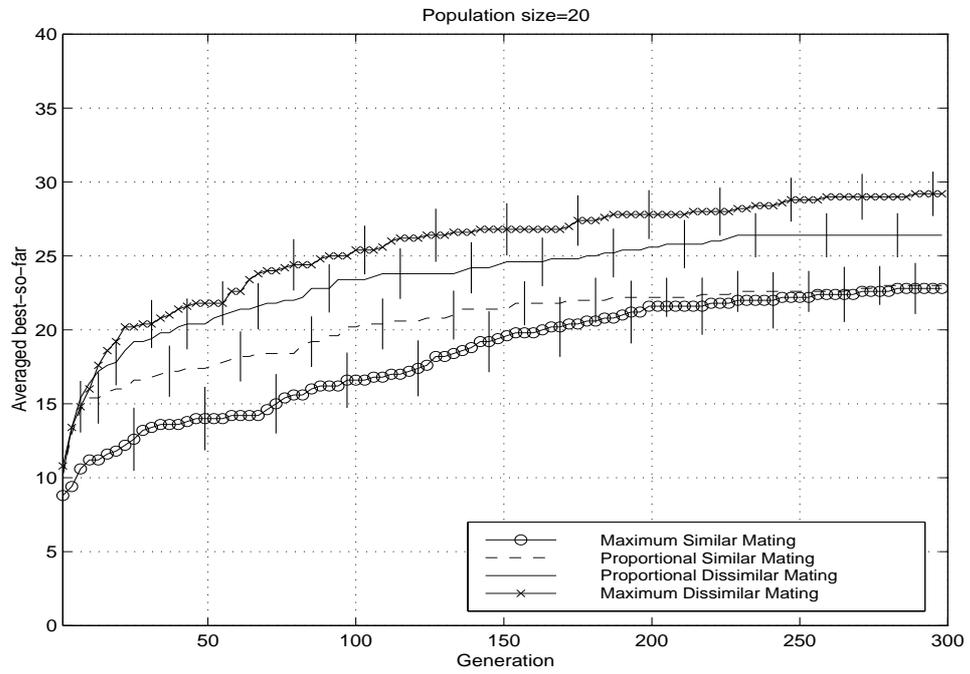


Figure 3.9: Best-so-far performance on S_1 .

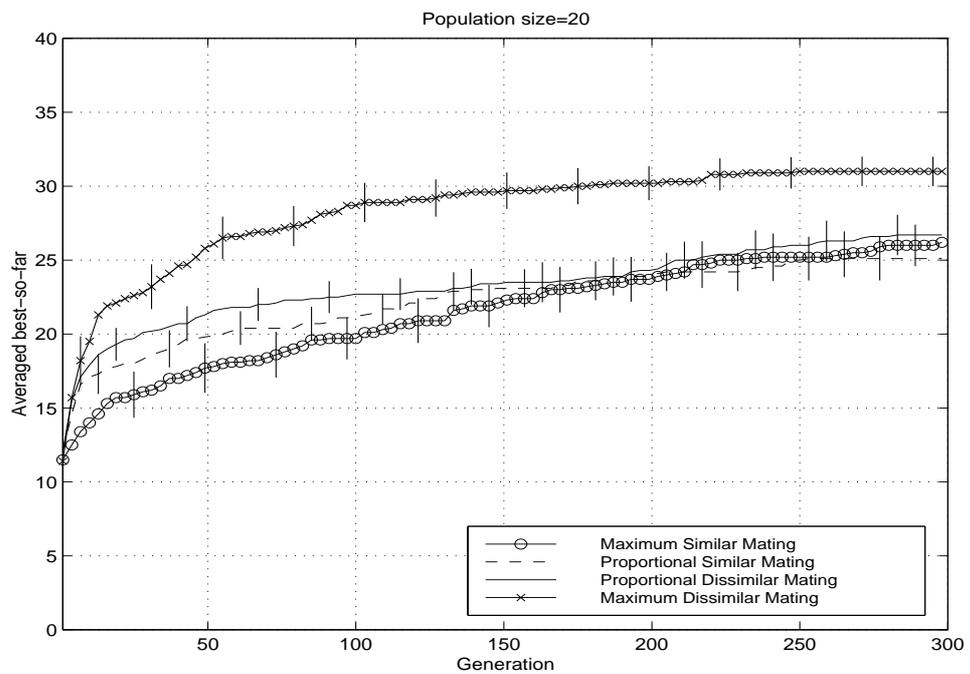


Figure 3.10: Best-so-far performance on IS_1 .

to a higher fitness stage under the dissimilar mating. It turns out that the results are interesting anomalies.

In Figure 3.11 and Figure 3.12 we see that the similar mating yields higher mean population fitness than the dissimilar mating. Most surprisingly, the mean population fitness obtained for the maximum dissimilar mating is relatively small and does not seem to improve at all.

We have learned that by suppressing hitchhiking and the founder effect, the maximum dissimilar mating retains more genetic variation in the population. The further exploration of the search space for the GA thus yields a better best-so-far performance, yet at the expense of the mean population fitness. Namely, although the maximum dissimilar mating GA is engaged in searching and constructing better best-so-far individuals, this mate selection would seem to create more lethal hybrids that degrade the population's fitness.

The following examination supports this claim. As a simple example, I suppose the population is composed of bit-strings of length 4 and the relevant building blocks that are able to contribute are 11** and **11 (the optimal string is thus 1111). Consider lethal offspring that are easily removed from the population, say individuals of fitness 0. Then, for example, given a string X , 1100, and two candidate mating partners, Y_1 (1100) and Y_2 (0011), the maximum similar mating will require X to select Y_1 as the mate. Since X and Y_1 are identical, their mating generates neither the optimum nor lethal hybrids. In case of the maximum dissimilar mating, however, Y_2 will be chosen as the mate and there exists probability value $\frac{1}{3}$ for generating the optimum and $\frac{2}{3}$ for lethal offspring after crossover.

The empirical results in Figure 3.13 validate the claim made above, from which we can see the significant proportions (around between 0.4 and 0.5) of lethal offspring generated from the maximum dissimilar mating. This indicates that for the maximum dissimilar mating about 8 to 10 individuals (population size 20) at each generation are

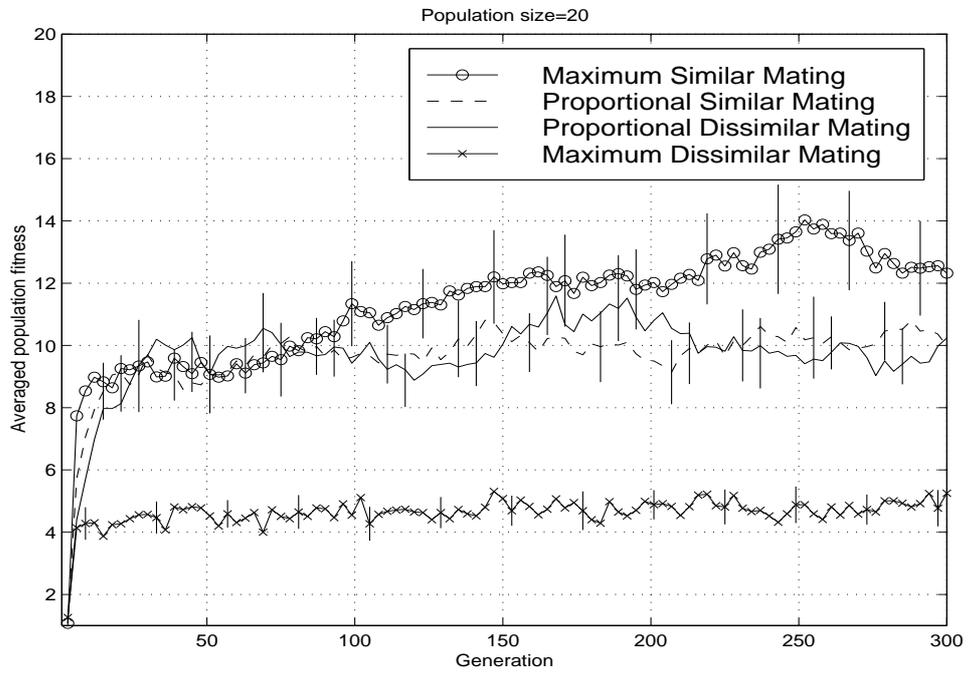


Figure 3.11: Averaged population fitness on S_1 .

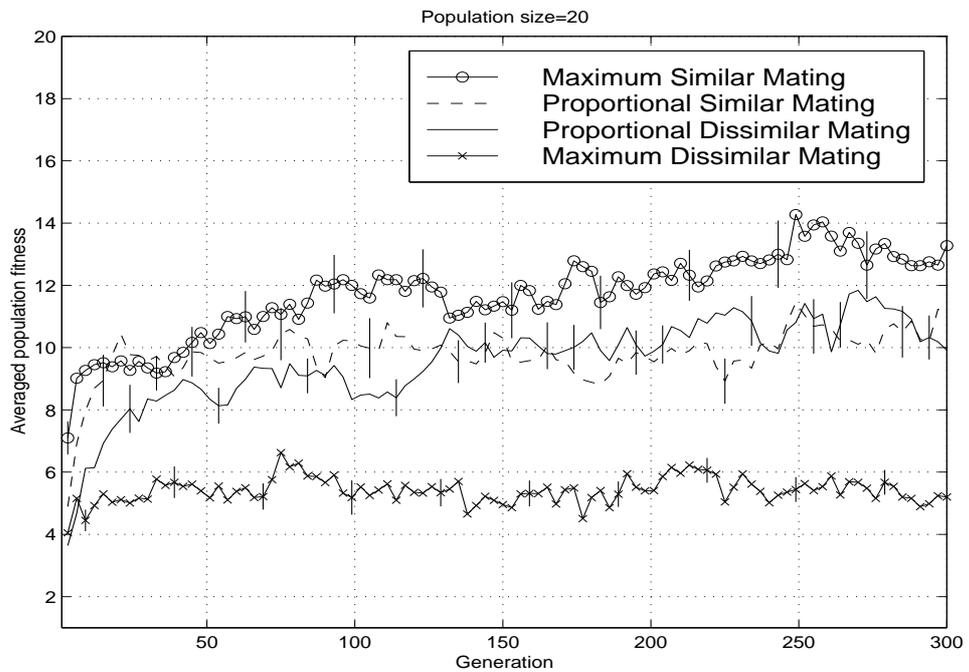


Figure 3.12: Averaged population fitness on IS_1 .

lethal and rapidly disappear from the population. Similar matings, on the contrary, prevent the creation of lethal offspring. In particular, matings between maximally similar individuals preserve the proportions of lethal offspring at the level lower than 0.1 after a few generations, meaning that on average less than 2 strings are of fitness value 0 (in fact, these lethal strings are generated from mutation, because crossover does not have effects on the maximum similar mating scheme).

We can further compare the difference between the maximum similar mating (MSM) and the maximum dissimilar mating (MDM). Figure 3.14 shows the results over 50 runs for averaged mean population fitness (i.e., each run's mean population fitness is collected and then averaged over 50 runs), averaged maximum fitness (i.e., the maximum fitness in the population for each run is collected and averaged over 50 runs), and averaged minimum fitness (i.e., the minimum fitness in the population for each run is collected and averaged over 50 runs) at each generation. It is interesting to note that the difference between the averaged maximum and minimum fitness for the MDM is greater than that for the MSM, indicating that the MDM indeed generates larger fitness variation while achieving better best-so-far performance.

All the results above show that the maximum dissimilar mating mechanism retains relatively large population diversity that facilitates the search for improving best-so-far, at the expense of creating more lethal hybrids. In the next subsection, I present an analysis on population diversity that helps enhance our understanding of the GAs with different mate selection schemes.

3.4.4 Diversity

In simple genetic algorithms of finite population size, use of the “survival of the fittest” principle generates high selection pressure towards higher fitness individuals. The number of offspring of these individuals increases over time, and lower fitness individuals are gradually discarded from the population. According to the schema

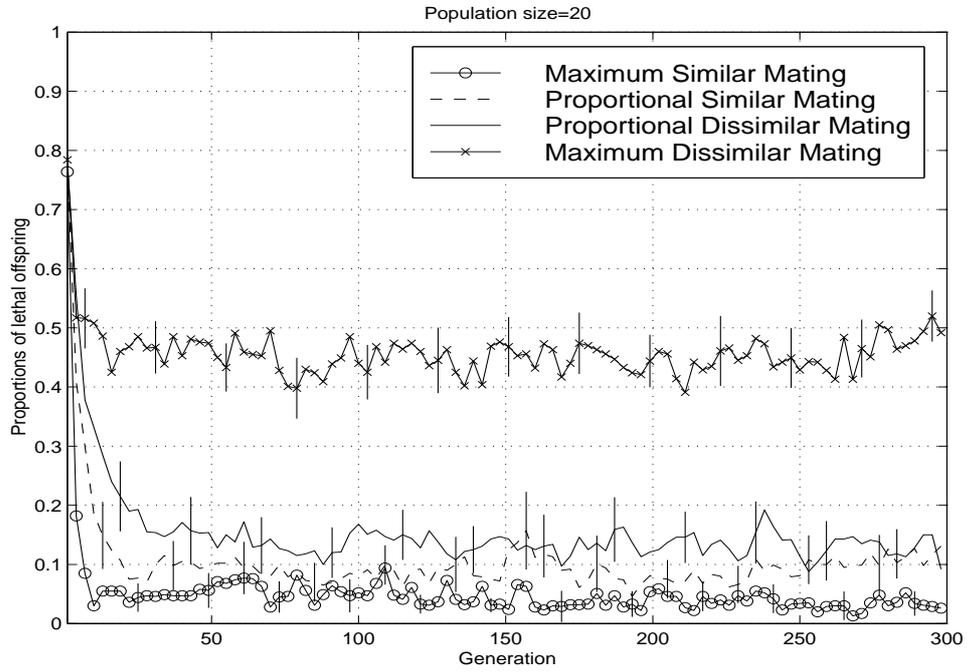


Figure 3.13: Proportions of lethal offspring on IS_1 .

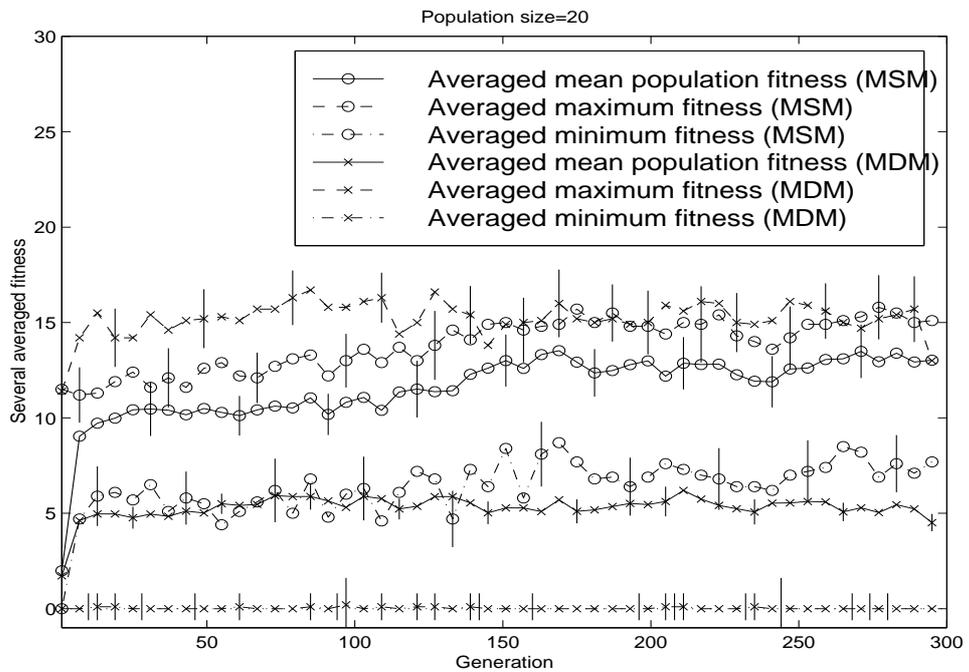


Figure 3.14: Several averaged fitness on IS_1 .

theorem, the increase of the offspring number of highly-fit individuals results in exponentially increasing numbers of trials to potential building blocks in the search space. The process of emphasizing building blocks brings about what is known as *convergence* in the GA literature.

For some optimization problems, convergence may be an advantage. But if selection pressure is high enough that the degree of convergence is too strong, GAs may suffer so-called *premature convergence*—the individuals prematurely converge on undesirable sub-optima. In this case, the population diversity is quickly lost and GA’s exploration of the search space is greatly limited.

In many other GA applications, they would require individuals to be distributed over several solutions. For example, an original motivation for developing niching methods was to promote diversity in the traditional GA (Mahfoud, 1995). In this approach, maintaining the population diversity means maintaining as many optima (or sub-optima) as possible. As a second example, in a classifier system (Holland, Holyoak, Nisbett, and Thagard, 1986) we are concerned with searching for a collective set of rules that performs well in a task environment, each rule playing a unique and complementary role. Thus, the system needs to evolve a set of rules that are specialized to various tasks (or niches) rather than producing a homogeneous (converged) population of similar rules. Analogously, the computational model of the immune system (Forrest and Perelson, 1991) requires a population of antibodies to evolve to cover a set of antigens. If the antibody population is sufficiently large, it clearly makes sense to evolve antibodies that are specialized to recognize different classes of antigens instead of evolving one generalist antibody that weakly matches all antigens. For these more ecological environments, the objective usually involves promoting diversity and covering multiple solutions, instead of convergence on a single solution.

Therefore, maintaining population diversity is crucial for the long term success

of many evolutionary systems. If diversity is properly utilized, the population would be able to adapt quickly to changes in the environment. It allows the population to continue searching for productive niches, and avoid becoming trapped in the basins of some local optima. In case of multimodal function optimization, diversity should be utilized in search for locating a single best-so-far solution (or the global optimum), or multiple solutions, or maintaining stable subpopulations. These three goals are not mutually exclusive—techniques that try to reach the global optimum, or multiple solutions in a multimodal function often encounter many local optima during the course of search. Likewise, techniques that try to maintain subpopulations are likely to perform a useful exploration in the search space and are also likely to identify the global optimum if some local optima constitute the way to the global optimum. In general, a meaningful GA exploration of the search space should be goal-oriented.

Thus far I have studied the GA's search for a single solution in terms of the best-so-far performance. In Chapters 4, 5 and 6, I will continue the study on this direction; and in Chapter 7, the other two goals will be studied. Before doing so, let us first inspect allelic diversity for the GAs with different mate selection schemes used in the preceding sections.

Allele Diversity

The schema density plots used in the preceding sections serve well to study how different mate selection schemes affect the GA's search process. The observation is that the dissimilar mating schemes would maintain more genetic variation than the similar mating schemes, yet we have not yet zeroed in on a firm examination on how that is so. In this subsection, I adopt a microscopic genetic diversity measure that would clearly help examine the dynamics of allelic frequencies for the different mate selections.

Diversity measures based on allele frequencies are common in both genetic algo-

rithms and biological genetics. With a single genome based on a binary alphabet, the most common notion of “fully diverse” is a single goal containing 50% zeros and 50% ones.

Let i be an arbitrary locus from 1 to l (string length). To measure diversity at the i th locus, a simple bitwise diversity metric can be defined as follows (Mahfoud, 1995):

$$D_i = 1 - 2|0.5 - p_i|, \quad (3.5)$$

where p_i is the proportion of 1s at locus i in the current generation. Thus the maximum of D_i is 1 when there are equal proportions of 1s and 0s at locus i ; and the minimum is 0 when all individuals' locus i is fixed to either 1 or 0.

Collins and Jefferson (1991) took the bitwise diversity metric (3.5) at each locus, and used the average over all loci as a combined diversity measure for the population. Therefore we have the allelic diversity measure for the population:

$$D = \frac{\sum_{i=1}^l D_i}{l}. \quad (3.6)$$

D has a value of 1 when the proportion of 1s at each locus is 0.5 and 0 when all of the loci are fixed to either 0 or 1. Effectively it measures how close the allele frequencies are to a random population (1 being closest).

Figure 3.15 and Figure 3.16 show the allelic diversity for the four mate selection schemes on the test functions S_1 and IS_1 . We can see that the allelic diversity for the maximum dissimilar mating is relatively close to 1 (random population), indicating that the GA with this mate selection actually wanders around the search space to collect existing information present in the population and locate useful genetic material.

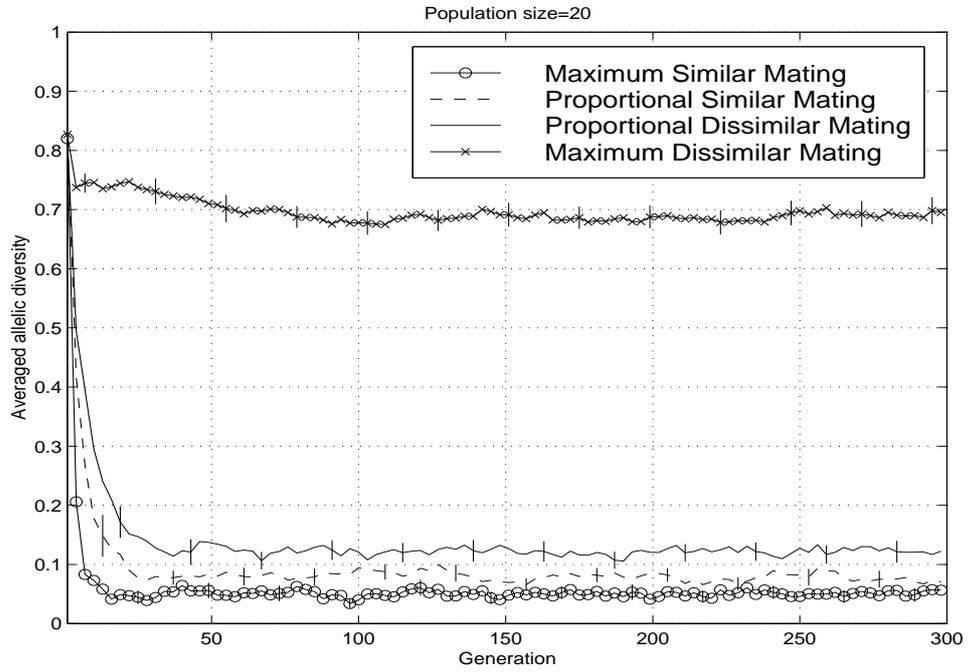


Figure 3.15: Averaged allelic diversity on S_1 .

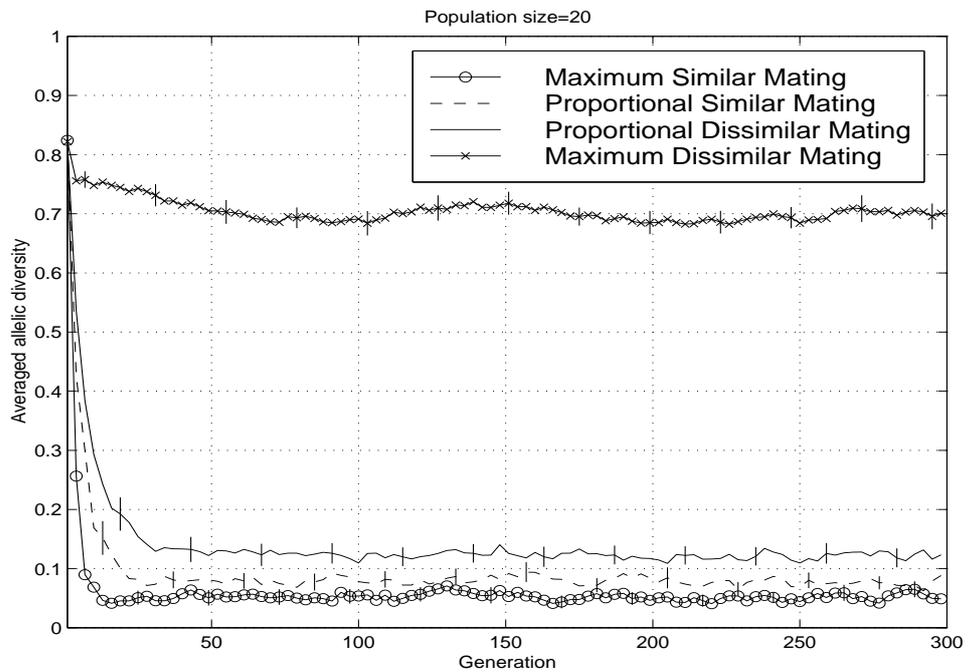


Figure 3.16: Averaged allelic diversity on IS_1 .

This is an example that GA efficiently utilizes diversity to further improvements. It is worth mentioning that not all cases with allelic diversity being close to 1 would guarantee a meaningful search; apparently without selection and variation a random population keeps having the largest allelic diversity value 1, but no meaningful search will be existing in such a population. This is consistent with the claim made by Mahfoud (1995): “... , this microscopic perspective can be misleading. Biases in the search space such as optima at Hamming cliffs or such as complementary dual optima may allow diversity at all bit-positions, but typically do not maintain a variety of subsolutions.” For example, the test function IS_1 has dual optima and if we were to locate the global optimum, it is more beneficial to maintain the building blocks that constitute the global optimum (i.e., s_1 , s_3 , s_5 , and s_7), rather than allowing diversity at all the complementary dual schemata.**

3.4.5 Tournament Mate Selection

The background selection scheme adopted so far has focused on fitness-proportionate selection. In the beginning of this subsection I intend to adopt another selection scheme and compare the difference between the two background selection schemes.

For lower computation cost, I use a binary tournament selection (Goldberg and Deb, 1991) to pick two parents for each mating—with probability one the fitter of the two randomly sampled individuals is chosen.

Figure 3.17 is the averaged best-so-far performance on IS_1 for the GAs with fitness-proportionate selection and tournament selection. The results show that the tournament selection scheme consistently outperforms the fitness-proportionate selection scheme. This is no surprising because it has been shown that tournament

**I will revisit the diversity subject by using a Markov model in Section 4.3.1.

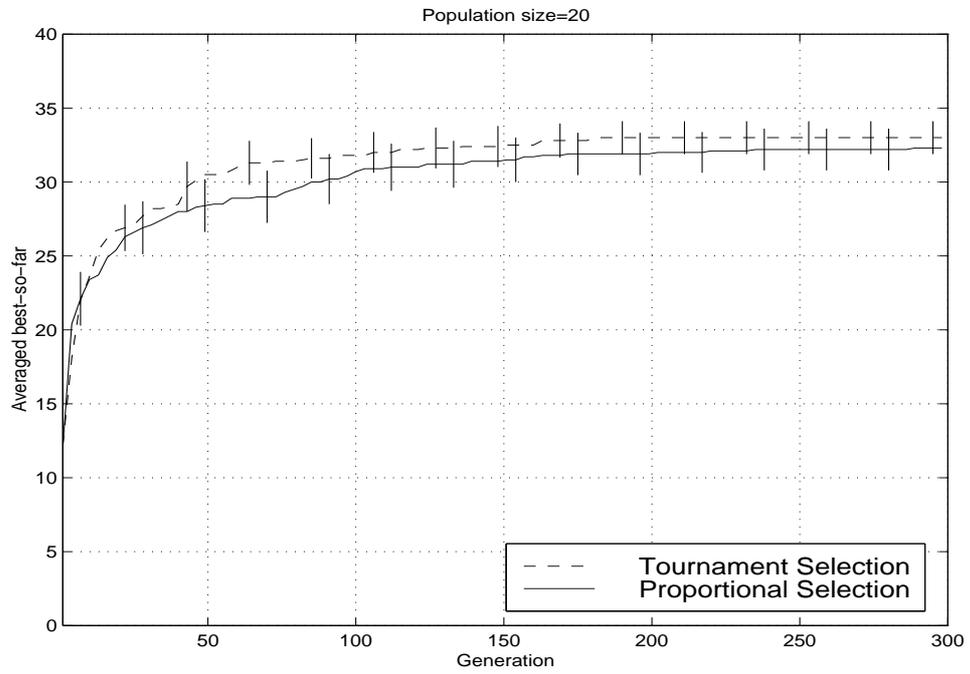


Figure 3.17: Best-so-far performance on IS_1 .

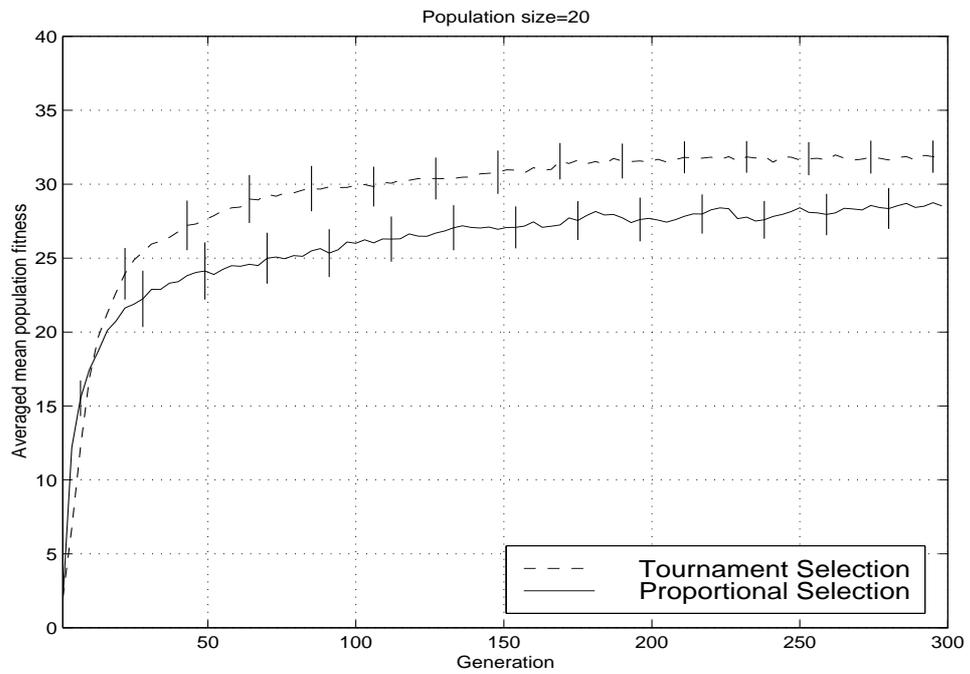


Figure 3.18: Averaged population fitness on IS_1 .

selection is capable of preventing too-quick convergence that normally occurs to the fitness-proportionate method. In Figure 3.18 we also see that the tournament selection maintains a better mean population fitness than the proportional selection. Unlike the maximum dissimilar mating in which the best-so-far performance is improved at the expense of the mean population fitness, the tournament selection scheme consistently has an edge over the proportional selection on these two goals. Therefore, in the rest of this chapter, the tournament selection is adopted as the background for conducting experiments based on several mate selection schemes.

The mate selection schemes proposed thus far involve calculating the Hamming distances for all the population individuals, resulting in $O(N^2)$ similarity comparisons. This is not computationally efficient. As a matter of fact, in natural world individuals seldom look around all the population and then choose the best mate; it is more frequent that an individual only picks a subset of population, performing some kind of comparison and sorting, and finally decides to mate with the most satisfying population member. The mate selection strategies based on a subset of the population is a natural way to save computational cost. This inspires us to adopt the following mating strategies:

During each mating event, a binary tournament selection—with probability one the fitter of the two randomly sampled individuals is chosen—is run to pick out the first individual, then choosing the mate according to the following schemes:

Tournament Selection (TS): Run the binary tournament selection again to choose the mate.

Tournament Dissimilar Mating (TDM): Run the binary tournament selection two more times to choose two candidate partners; then the one more dissimilar to the first individual is selected for mating.

Tournament Similar Mating (TSM): Run the binary tournament selection two

more times to choose two candidate partners; then the one more similar to the first individual is selected for mating.

Random Dissimilar Mating (RDM): Randomly choose two candidate partners; then the one more dissimilar to the first individual is selected for mating.

Random Similar Mating (RSM): Randomly choose two candidate partners; then the one more similar to the first individual is selected for mating.

I still use the Hamming distance as the similarity metric. Notice that in the mate selection schemes above if the two candidates are of the same Hamming distance to the first individual, then one of them is randomly selected.

In the five approaches above, the first individual is always sampled by the regular tournament selection. For the tournament dissimilar and similar mating, there are two ways to affect an individual's probability of being selected. The first comes from the explicit fitness evaluation provided by a given test function. The second is from the coevolution of population members, each individual preferring other individuals that possess certain characteristics. The two sources complicate the actual probability of an individual being selected. I expect that tournament selection contributes more selection pressure toward highly-fit individuals, and the mate preference refines the searching for mates. As for the random dissimilar and similar mating, the selection pressure is reduced by removing the tournament selection acting upon the candidate mates. The only source that affects the mate selection probability is precisely the mating preference, which exerts a selection pressure on the population based on genotype.

For the GAs with these five tournament-based mate selections, the experiments are conducted for 50 runs, based on population size 20, mutation rate 0.005, and one-point crossover rate 1. Figure 3.19 shows the experimental results for the test function S_1 . Analogous to the results obtained in the foregoing subsections, dissimilar mating

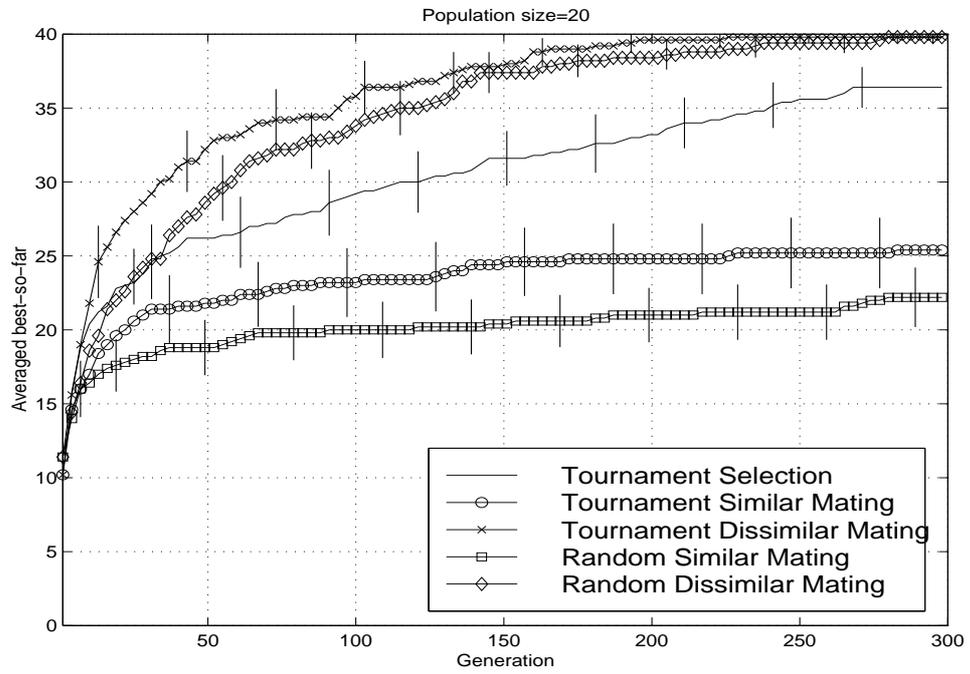


Figure 3.19: Best-so-far performance on S_1 .

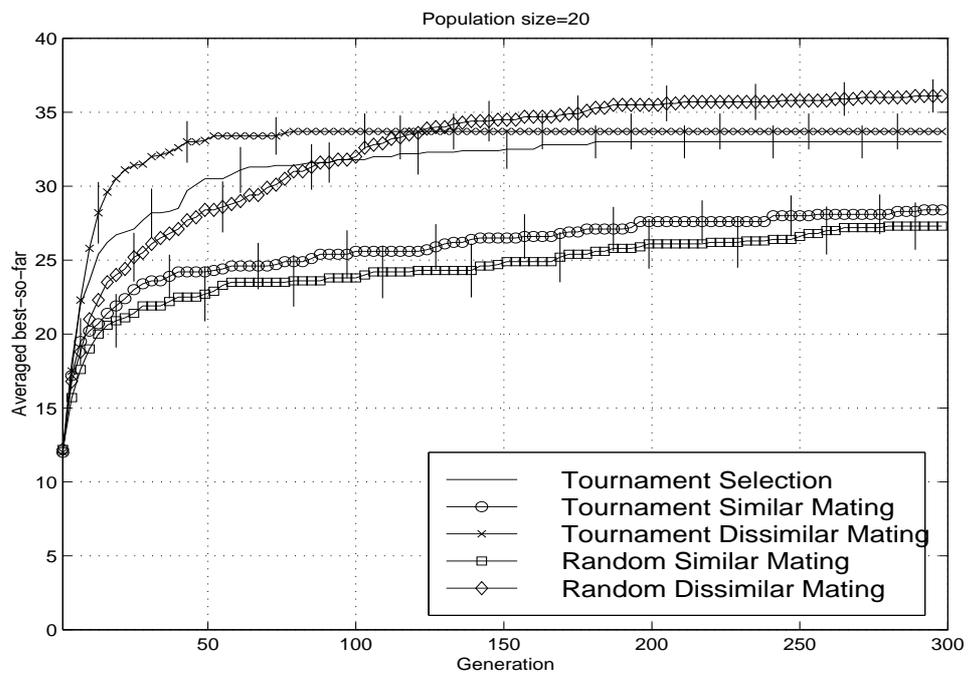


Figure 3.20: Best-so-far performance on IS_1 .

outperforms others in terms of the best-so-far performance. In the tournament dissimilar mating, when tournament selection is incorporated for selecting the second individual, it obviously confers larger selection pressure towards highly-fit individuals and yields better results than the random dissimilar mating. (Though Figure 3.19 seems to show that the tournament dissimilar mating outperforms the random dissimilar mating only over early generations, the discrepancy will be clearer if we compare the mean function evaluations for attaining the global optimum (20 1's) for these two schemes: the tournament dissimilar mating spent an average of 4783 evaluations to reach the optimum, and for the random dissimilar mating it is 5515.) However, Figure 3.20 shows a different situation: the best-so-far for the tournament dissimilar mating rises quickly and stops before 100 generations. On the contrary, the best-so-far for the random dissimilar mating exhibits gradual improvements, exceeding the performance of the tournament dissimilar mating around generation 120, and never stops before generation 300.

We can further inspect the averaged population fitness. Recall that in the maximum dissimilar mating case the mean population fitness is at a much lower level than the other similar mating schemes. Figure 3.21 shows that the random dissimilar mating, while achieving a much better best-so-far performance on IS_1 , still maintains the mean population fitness at the same level as those of the two similar mating schemes. Figure 3.22 is the experimental results that correspond to the proportions of the resulting lethal offspring (strings of fitness value 0) generated from these mating schemes. Before generation 100, the lethal proportions of all five schemes have substantially dropped down to zero, even in the random dissimilar mating case. This shows that the random dissimilar mating can retain reasonable mean population fitness, while gaining much better advancements of the best-so-far.

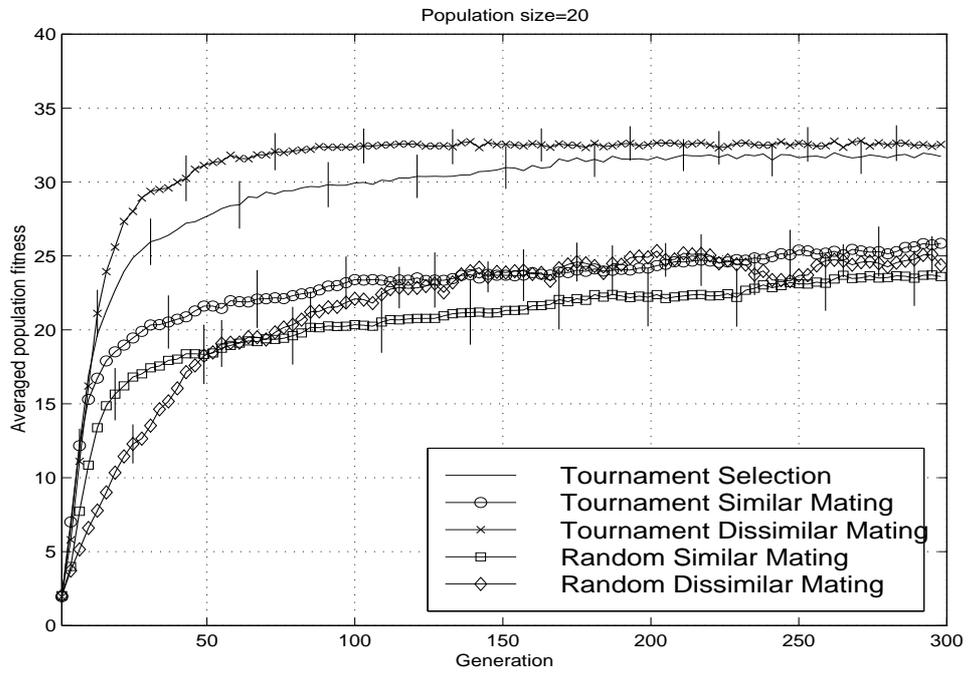


Figure 3.21: Averaged population fitness on IS_1 .

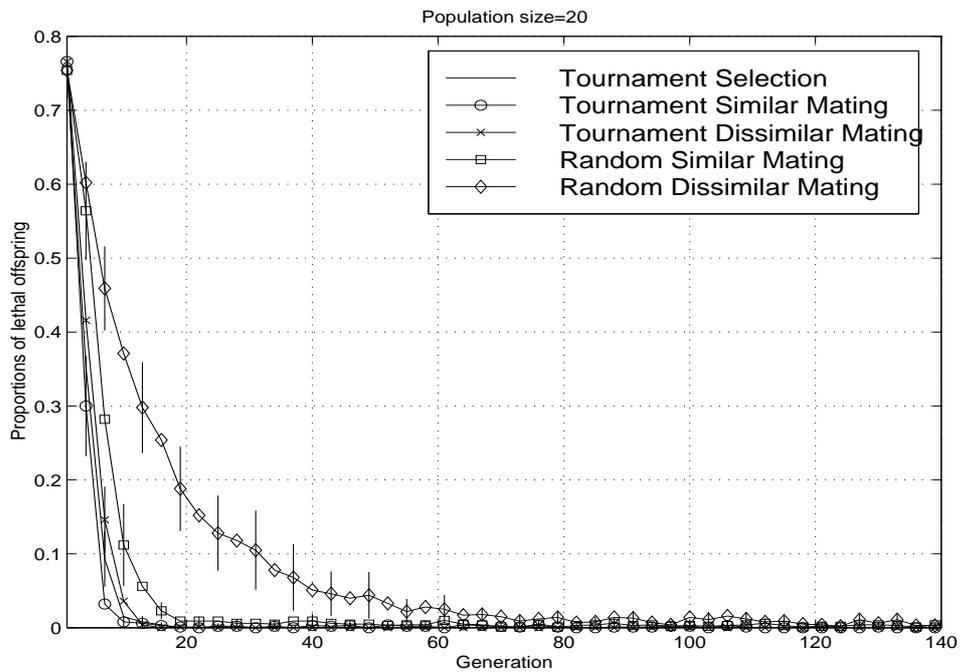


Figure 3.22: Proportions of lethal offspring on IS_1 .

In Section 3.4.1, I have discussed that in function S_1 hitchhiking is a problem for GA's search that worsens as the selection pressure becomes larger. Though the tournament dissimilar mating involves a higher selection pressure than the random dissimilar mating, its preference for dissimilar mates overwhelms the hitchhiking problem and yields better results. On the other hand, in function IS_1 where the founder effect becomes dominant (see Section 3.4.2), the higher selection pressure resulting from the tournament dissimilar mating facilitates the founding of incompatible schemata. When schemata that are not components of the global optimum were founded, the overall performance is greatly impeded. This accounts for the worse result in the tournament dissimilar mating, compared to that of the random dissimilar mating. This phenomenon will become more salient when I present the experimental results on a deceptive test function in the next subsection.

In addition, though RDM removes the selection pressure toward higher-fitness individuals while selecting mating partners to reduce the degree of premature convergence, this in turn increases the likelihood of useless hybrids being generated by lower-fitness parents. This may account for why in Figure 3.22 the proportion of lethal offspring generated by RDM is larger than that generated by TDM in the earlier generations. Justifying this conjecture would need more analysis, and this direction will be deferred for future work.

3.4.6 Effects of Deceptiveness

The problems of bounded deception designed by Goldberg et al. (1989) were used to investigate the performance of GAs on deceptive functions in which low-order, highly-fit schemata mislead GAs away from global optima and toward the complement of the global optimum. One example that exhibits this kind of deception is an order-5 deceptive function as defined in Table 3.3.

The fitness contribution of each block increases as the number of zeros in the function rises. On a similar testbed, 3-bit fully deceptive function, Goldberg et al. (1989) calculated the average fitness of schema and showed that GAs are likely to be

Table 3.3: A Deceptive, Order-5 Problem.

bit	value	bit	value
11111	8	11000	3
00000	5	11100	2
10000	4	11110	1

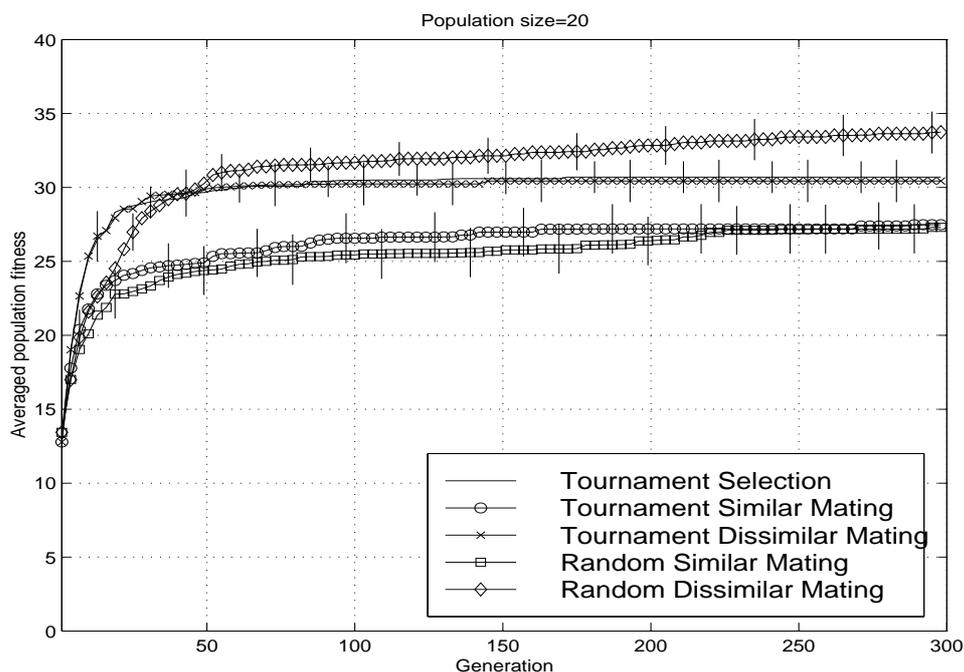


Figure 3.23: Best-so-far performance on D_1 .

led toward the complement of the global optimum.

To demonstrate the effects of the deception on the search power of GAs, I designed a 20-bit deceptive function D_1 , which is composed of four consecutive blocks of the 5-bit deceptive function. The deceptive function D_1 is an extension of the test function IS_1 —the founder effect may occur when a particular EA acts on it. Furthermore, the deceptiveness embedded in this function may gradually mislead GA away from the global optimum. Therefore, I expect that this 20-bit deceptive function imposes more difficulty on GAs than IS_1 .

The results of the GAs with the five tournament-based mate selection schemes are presented in Figure 3.23, where all the parameters are the same as those used in the

previous subsection. Compared with Figure 3.20, it is clearly that the performance of the tournament dissimilar mating is worsened, since its best-so-far curve falls below that of the regular tournament selection when generations are large enough. The GA with the random dissimilar mating still gains gradual improvements and never stops before generation 300.

As I discussed in the preceding subsection, the tournament dissimilar mating involves a higher selection pressure than the random dissimilar mating, which could be beneficial in improving the best-so-far performance on a non-deceptive testbed. However, on a deceptive function where the founder effect may become dominant, the higher selection pressure can in turn enhance the convergence on schemata that are not components of the global optimum, and the GA's search process is greatly impeded. On the other hand, since RDM removes the selection pressure toward higher-fitness individuals while selecting mating partners, this in turn reduces the degree of premature convergence. Thus the GA's population can further explore the search space to continue improving the best-so-far performance.

These phenomena will be seen again in Chapters 5 and 6.

3.4.7 Effects of Population Size

As population size is a source of diversity for GAs, I expect that population sizes have important effects on GA's behavior. In this subsection, I present some experiments for the best-so-far performance of the GAs based on different population sizes.

The experiments are conducted over 30 runs based on one-point crossover rate 1, mutation rate 0.005, with population size 20, 50, 80 and 200 for each case. For each run, I use the maximum of 10000 function evaluations as a baseline for comparing the effects of the mate selection schemes. The results are shown in Figure 3.24.

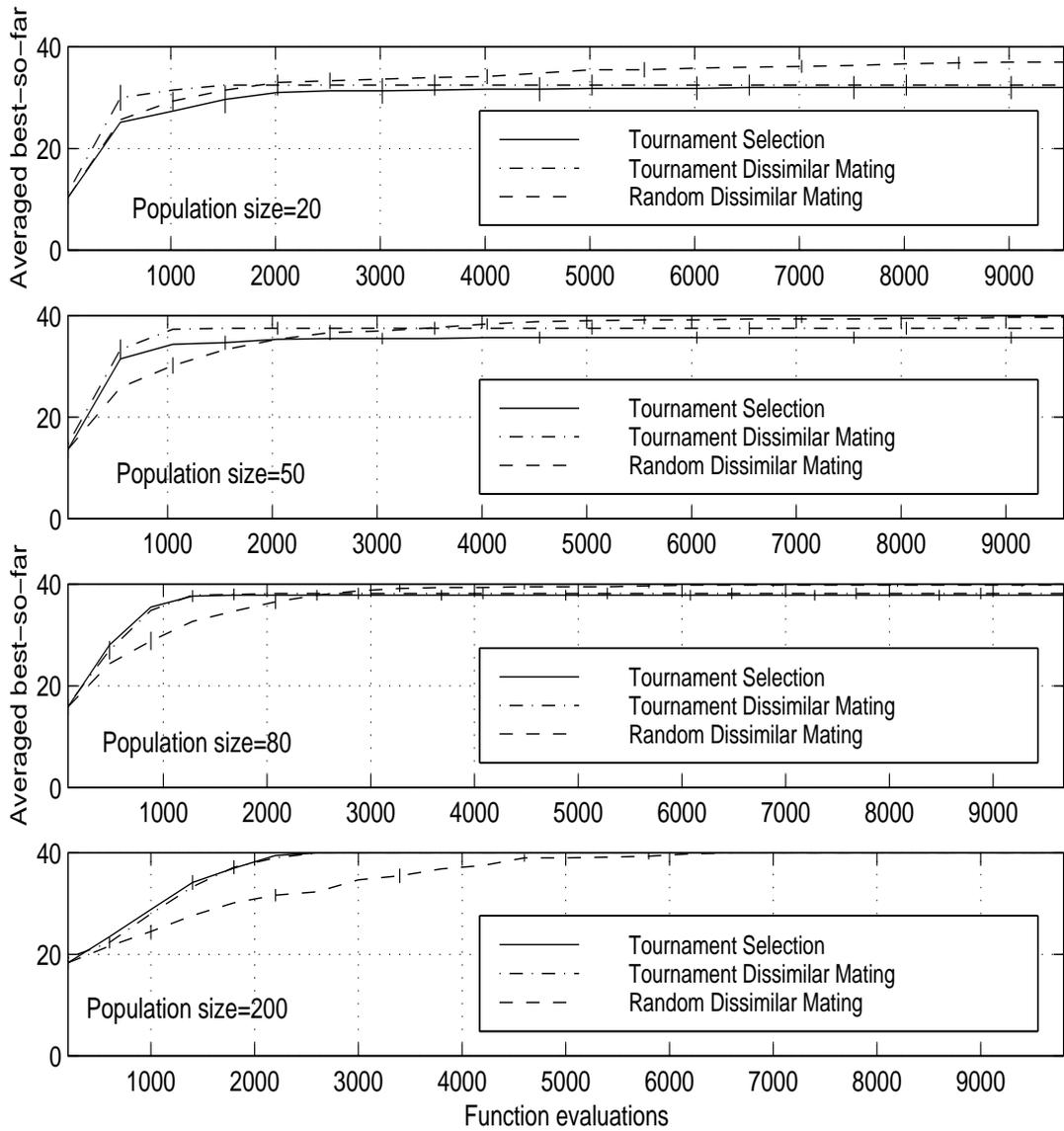


Figure 3.24: Averaged Best-so-far on IS_1 for various population sizes.

The first three plots (population size 20, 50, and 80) show that the random dissimilar mating GA achieves higher averaged best-so-far than the other two, although it has a slower rate of improvement. Again, the possible reason is that the random dissimilar mating does not employ the selection pressure toward higher-fitness individuals while selecting mating partners, which in turn slows down the GA's searching process.

For small population sizes (20 and 50), the trends in these results show that the GA with the tournament dissimilar mating outperforms the GA with the traditional tournament selection. In addition, in case of population size 200, these three GAs all reach the global optimum. This indicates that larger population sizes can suppress the difference between the best-so-far values attained by the three GAs.

These results show that population size is an important factor that affects the GA's performance. If the time allowed for an experiment is limited, such as maximum function evaluations, then using the tournament dissimilar mating would be more beneficial. On the other hand, if the time for experiments is not severely limited, one may want to use the GA with the random dissimilar mating scheme to achieve higher best-so-far values.

In Chapter 5 and 6 I will use harder problems to further investigate the effects of the three mate selection schemes, where I use moderate population sizes. As will be seen, those results are consistent with the results obtained above.

3.5 Summary and Discussions

In this chapter the framework proposed for investigating similarity and dissimilarity-based mate selection schemes allows us to conduct a systematic analysis to deepen our understanding of GAs. I first started with mating preference for the maximally similar or dissimilar individuals, and showed that, in the context of GA, hitchhiking and the founder effect can be explained in terms of the schema analyses, and these two phenomena can be suppressed by dissimilar mating choices. Afterwards I introduced

more complicated means for calculating individuals' probabilities of being selected via combining similarity test and fitness-proportionate selection. Allowing population members to discriminate candidate mates introduces another source of selection pressure, in addition to the selection pressure arising from the environment. The resulting GA hence forms a more complex system in which individuals' fitnesses depend on both the environment and other population members.

The very essence of good GA design is retention of diversity, furthering exploration, while exploiting building blocks already discovered. The results I obtained demonstrate that proper mating preferences are shown to indeed facilitate discriminating individuals in the population, retain genetic diversity, and better utilize building blocks already discovered for exploration of the search space. The advantage of using dissimilarity-based mating preferences is reflected by the corresponding GA's improved best-so-far performance.

Note that in studying the creation of lethal hybrids and the diversity issue, the most surprising is that the results in Figure 3.13, 3.15 and 3.16 show a relatively large difference between MDM and PDM, compared with the difference between the PDM, PSM and MSM, since one may expect that the difference between MDM and PDM, and the difference between PDM and PSM are at similar levels. The reason could be that MDM is a relatively "disruptive" mating strategy which simply selects the most dissimilar mating partners to collect different information for increasing a single individual's fitness, yet at the expense of generating more useless hybrids. To test this hypothesis would need more analysis, which will be deferred for future work.

After studying the four mating schemes based on fitness proportionate selection, I then adopted tournament selection as the background selection scheme to proceed for other analyses, due to its general superior performance to fitness proportionate selection. The relevant results again show that the dissimilar mating schemes are more beneficial than the traditional selection and similar mating schemes.

I also conducted studies of the effects of population size on the GA's search power. The results show that if the time allowed for an experiment is limited, then using the tournament dissimilar mating is likely to facilitate the GA's search process. On the other hand, if the time for experiments is not severely limited, we can use the random dissimilar mating scheme to improve the GA's best-so-far performance.

In conclusion, the analyses reveal that individuals of more distinct characteristics (in terms of the Hamming distance) collectively facilitate the search for a single, better solution. The problem it would cause is lethal hybrids: matings between dissimilar individuals give crossover more opportunity to disrupt existing building blocks, leading to decrease of the mean population fitness. Therefore, to avoid producing lethal offspring, it may require that matings occur only between individuals of similar characteristics.

In the next chapter, I will use a Markov chain model to analyze GA for several mate selection strategies. Although the Markov chain approach involves enormous numbers of states that generate great computation burden, and I have to concentrate on problems of limited state space size, it turns out that this model can provide useful insights—the behavior of an GA on small (computationally tractable) problems can actually be observed in larger problems. I will use the Markov chain analysis to enhance the understanding of the GAs studied in this chapter, and use the insights obtained to further investigation in the following chapters.

CHAPTER 4

MARKOV CHAIN ANALYSIS

4.1 Introduction

A large number of systems arising in practice have the property that the present state completely determines the future; i.e., the past states have no influences on the future. This property is called the *Markov property*, and systems satisfying this property are called *Markov chains* for discrete time cases, or *Markov processes* for continuous time cases. For example, let the system be observed at the discrete moments of time $n = 0, 1, 2, \dots$, and let X_n denote the state of the systems at time n . Then X_n has the Markov property if

$$P(X_{n+1} = x_{n+1} | X_0 = x_0, \dots, X_n = x_n) = P(X_{n+1} = x_{n+1} | X_n = x_n)$$

for every choice of the nonnegative integer n and the numbers x_0, \dots, x_{n+1} , each in the state space of the system.

A discrete-time system that satisfies the Markov property and is composed of N states can be completely described by the $N \times N$ state transition matrix Q , which gives the probability of transitioning from state i at time n to state j at time $n+1$:

$$Q(i, j) \equiv p_{i,j} \equiv P(X_{n+1} = j | X_n = i)$$

The $p_{i,j}$ values define the “one-step” probability transition matrix Q , since it describes the probability of transitioning from state to state in one time step. The

transient behavior of the system is obtained from the “ k -step” probability transition values, which are obtained from the k th power of Q :

$$Q^k(i, j) \equiv p_{i,j}^{(k)} \equiv P(X_{n+k} = j | X_n = i)$$

Since a “state” of simple genetic algorithms can be defined by a particular population, and the composition of the population at the next step is entirely determined by the present population, it has been quite natural to model simple GAs as Markov chains (De Jong 1975; Goldberg and Segrest 1987). One can then imagine a state space consisting of all possible populations and examine the characteristics of the population trajectories a GA produces over time.

There are several Markov models that were derived assuming infinite population and involve characterizing steady state behavior (Davis and Principe, 1991; Suzuki, 1993; Rudolph, 1994). On the contrary, the model developed by Nix and Vose (1992) was based on finite population size. De Jong, Spears, and Gordon (1994) applied this finite-population model to investigate GA-based function optimization (GAFO), in which they performed transient Markov chain analysis to calculate the mean waiting times. The Nix and Vose model and the GAFO theory of De Jong et. al. serve as the basis for the Markov chain analysis of mate selection in this chapter.

In the following sections, I briefly summarize the Nix and Vose Markov model. Then I show how the mate selection schemes are incorporated in this model, and use them for simple visualization analysis. Afterwards, a brief summary of the GAFO theory of De Jong et. al. is presented, following some computational explorations for understanding effects of different mate selections.

4.2 The Nix and Vose Markov Model

Nix and Vose (1992) introduced a Markov chain model that is intended to represent a simple, finite-population GA based on a standard binary representation, fitness-

proportional selection scheme, standard mutation, and one-point crossover operators.

The simple GA model they considered moves from one generation to the next as follows:

- (1) Obtain two parents by proportional selection.
- (2) Mutate (mutation implies change) the parents with rate μ .
- (3) Produce the (mutated) parents' child by one-point crossover with rate χ .
- (4) Put one child into the next generation.
- (5) If the next generation contains less than the population size, go to step 1.

The search space is the set of all length l strings and $r = 2^l$ is the total number of possible strings. If n is the population size, then the number of possible populations, N , corresponding to the number of possible states is:

$$N = \binom{n + 2^l - 1}{2^l - 1}.$$

The possible populations are described by the matrix Z , which is an $N \times r$ matrix.

* The i th row $\phi_i = \langle z_{i,0}, \dots, z_{i,r-1} \rangle$ of Z is the incidence vector for the i th population. Thus $z_{i,y}$ is the number of occurrences of string y in the i th population, where y is the integer representation of the binary string. For example, suppose $l = 2$ and $n = 2$; then $r = 4$, $N = 10$ and the Z matrix is shown in Table 4.1:

With these definitions, Nix and Vose derived the formula for calculating exact state transition probabilities $Q_{i,j}$:

$$Q_{i,j} = n! \prod_{y=0}^{r-1} \frac{(M[\frac{F\phi_i}{|F\phi_i|}]_y)^{z_{j,y}}}{z_{j,y}!}, \quad (4.1)$$

where F is determined from the fitness function, and M depends on the mutation and crossover operators.

Considerable insights into transient behavior can be obtained computationally by computing and analyzing Q^k directly. Unfortunately, the size of the Q matrix for

*For programming convenience I transpose the Z matrix of Nix and Vose (1992), as indicated by De Jong et al. (1994).

Table 4.1: The Z matrix ($n=2$ and $l=2$).

State	00	01	10	11
P1	0	0	0	2
P2	0	0	1	1
P3	0	0	2	0
P4	0	1	0	1
P5	0	1	1	0
P6	0	2	0	0
P7	1	0	0	1
P8	1	0	1	0
P9	1	1	0	0
P10	2	0	0	0

typical GAFO applications is computationally unmanageable since the number of states N grows rapidly with population size n and string length l . For example, a GA with population size 10 and bit-string length 6 has about 6.2×10^{11} states.

4.3 Incorporation of Mate Selection in the Nix and Vose Markov Model

The Nix and Vose Markov model consists of two key operators: F and M , where F relates to selection, and M relates to mutation and crossover. Since the mate selection schemes only affect the GAs' selection processes, altering the operator F is the major objective in this chapter. In addition, since the Nix and Vose model was developed using fitness proportionate selection, I will concentrate on modifying this model based on the first four mate selection schemes proposed in the foregoing chapter—the maximum similar mating, the proportional similar mating, the proportional dissimilar mating, and the maximum dissimilar mating. The tournament-based mate selections will not be explored in this thesis, but will be deferred for future work.

In the formal model of Vose and Liepins (1991), each string in the search space is specified by the corresponding integer equivalent between 0 and 2^l-1 . The population

at generation t is represented by two vectors, $\vec{p}(t)$ and $\vec{s}(t)$, each of length 2^l , where $\vec{p}(t)$ specifies the composition of the population at generation t , and $\vec{s}(t)$ represents strings' probabilities of being selected. Let F be a two-dimensional matrix such that $F_{i,j} = 0$ for $i \neq j$, and $F_{i,i} = f(i)$, where $f(i)$ is the fitness of string i . Then under proportional selection it yields

$$\vec{s}(t) = \frac{F\vec{p}(t)}{\sum_{i=0}^{2^l-1} F_{i,i}p_i(t)}, \quad (4.2)$$

where $p_i(t)$ denotes the i th component of $\vec{p}(t)$, which is the proportion of the population at generation t consisting of string i . Therefore the i th component of $\vec{s}(t)$ is the probability that string i will be selected as a parent.

For example, if $l=2$ and the population consists of two copies of 11 and one copy of 00 and 01, then

$$\vec{p}(t) = (0, 0.25, 0.25, 0.5).$$

If the fitness is equal to the number of ones in the string, then

$$\vec{s}(t) = (0, 0.1667, 0.1667, 0.6667).$$

Given these preliminaries, one can compute the expected proportion of string k at generation $t+1$:

$$E(p_k(t+1)) = \sum_{i,j} s_i(t)s_j(t)r_{i,j}(k), \quad (4.3)$$

where $r_{i,j}(k)$ is the probability that string k will be produced by a recombination event between string i and string j , given that i and j are selected to mate.

Then Vose and Liepins derived the operator M to encapsulate the calculations regarding crossover and mutation that account for $r_{i,j}(k)$. Finally, Nix and Vose (1993) followed these results to construct the finite-population model as described in Equation 4.1.

To understand how the mate selection schemes are integrated with the Nix and Vose Markov model, let us take a closer look at Equation 4.3, in which both the individuals selected for mating—string i and string j —are based on fitness-proportionate

selection. Recall that the maximum similar mating scheme proposed in Chapter 3 picks up the first individual, which plays the role of string i in Equation 4.3, by fitness-proportionate selection. Then the population member who is the most similar to the first one (in terms of the Hamming distance) is chosen as the second individual, which appears as string j in Equation 4.3. Hence the way of calculating $s_j(t)$ in Equation 4.3 is the only part that needs to be reconsidered.

Since the maximally similar individual must be chosen as the second string, it turns out that the fitness matrix F needs to be re-defined so that only the string that is maximally similar to the the first string can possess non-zero fitness values. If several strings are of the same minimum Hamming distance to the first string, these strings are assigned the same fitness values.

Plugging the new fitness matrix F into Equation 4.2 yields the new selection probability $s_j(t)$ of string j for Equation 4.3. The resulting Markov model is exactly the GA model with the maximum similar mating scheme.

The implementation above can be summarized in Table 4.2.

As for the maximum dissimilar mating, the first string will select as its mate the string whose Hamming distance is the maximum. Thus it only needs to replace “minimum” by “maximum” in Table 4.2, and this implementation is shown in Table 4.3.

We can proceed to derive the Markov model for the proportional similar mating in the same way. Recall that in the proportional similar mating, the first string is picked by fitness-proportionate selection, and the probabilities of strings being selected as the first string’s mate are *reversely* proportional to their Hamming distances. Since strings’ Hamming distances may be zero, I offset all the Hamming distances by 1 to avoid the case of dividing by zero. The implementation is shown in Table 4.4.

As for the proportional dissimilar mating, since the probabilities of population members being selected as the mate are proportional to their Hamming distances, it is easy to obtain the corresponding implementation as show in Table 4.5.

Table 4.2: Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the maximum similar mating.

1. Let $F_{i,j}=0$ for all i and j .
2. For i from 1 to r , do:
 - a. For j from 1 to r , compute the Hamming distance of string j to string i , if $z_{m,j} \neq 0$, where $z_{m,j}$ is the number of occurrences of string j in the m th population.
 - b. $F_{j,j} = 1$, if string j is the only one that possesses the minimum Hamming distance;
or $F_{j,j} = \frac{1}{u}$, if u is the number of strings whose Hamming distances are the minimum.

Table 4.3: Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the maximum dissimilar mating.

1. Let $F_{i,j}=0$ for all i and j .
2. For i from 1 to r , do:
 - a. For j from 1 to r , compute the Hamming distance of string j to string i , if $z_{m,j} \neq 0$, where $z_{m,j}$ is the number of occurrences of string j in the m th population.
 - b. $F_{j,j} = 1$, if string j is the only one that possesses the maximum Hamming distance;
or $F_{j,j} = \frac{1}{u}$, if u is the number of strings whose Hamming distances are the maximum.

Table 4.4: Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the proportional similar mating.

1. Let $F_{i,j}=0$ for all i and j .
2. For i from 1 to r , do:
 - a. For j from 1 to r , compute the Hamming distance of string j to string i , if $z_{m,j} \neq 0$, where $z_{m,j}$ is the number of occurrences of string j in the m th population.
 - b. $F_{j,j} = \frac{1}{D_j+1}$, where D_j represents the Hamming distance of string j .

Table 4.5: Procedure of calculating the second individual's probability of being selected in Equation 4.3 for the proportional dissimilar mating.

1. Let $F_{i,j}=0$ for all i and j .
2. For i from 1 to r , do:
 - a. For j from 1 to r , compute the Hamming distance of string j to string i , if $z_{m,j} \neq 0$, where $z_{m,j}$ is the number of occurrences of string j in the m th population.
 - b. $F_{j,j} = D_j$, where D_j represents the Hamming distance of string j .

4.3.1 Visualizing Markov Models and Diversity

Having modified the Markov model for the mate selection schemes, I can compute the state transition matrix for each case. Let Q_{MSM} , Q_{PSM} , Q_{PDM} , and Q_{MDM} correspond to the state transition matrices for the maximum similar mating, the proportional similar mating, the proportional dissimilar mating, and the maximum dissimilar mating, respectively.

Visualizing Q^k provides us with the insight regarding the effects that different mate selections have on the state transition matrix Q . (For example, see Horn, Goldberg and Deb (1994), and De Jong, Spears, and Gordon (1994) for additional evidence concerning the usefulness of this approach.) This can be implemented through plotting Q^k as an image, the gray level of coordinate (i,j) reflecting the probability that the GA will move from state i to state j in k steps. The density of blackness at each (i,j) indicates the corresponding k -step transition probability, where the blacker a point, the higher the corresponding probability value.

As an example, I use a fitness function with $f(y) = integer(y) + 1$, where $integer(y)$ returns the integer equivalent of the bit string y . The string length is 2, population size is 3 (thus the number of total states is 20), crossover rate is 1, and mutation rate is 0.01.

Figure 4.1 shows the results for various Q_m^k , where m represents MSM, PSM, PDM and MDM. Let us first inspect the case for one-step transition, i.e., $k = 1$. A clear, visible diagonal line from Q_{MSM}^1 indicates that significant changes in the population in one generation are very unlikely. As we scan the images from left to right, we see that this situation can be altered by gradually allowing matings to take place between relatively dissimilar individuals, generating more diffuse state transition matrices. The net result is that dissimilarity-based matings can make larger changes more easily.

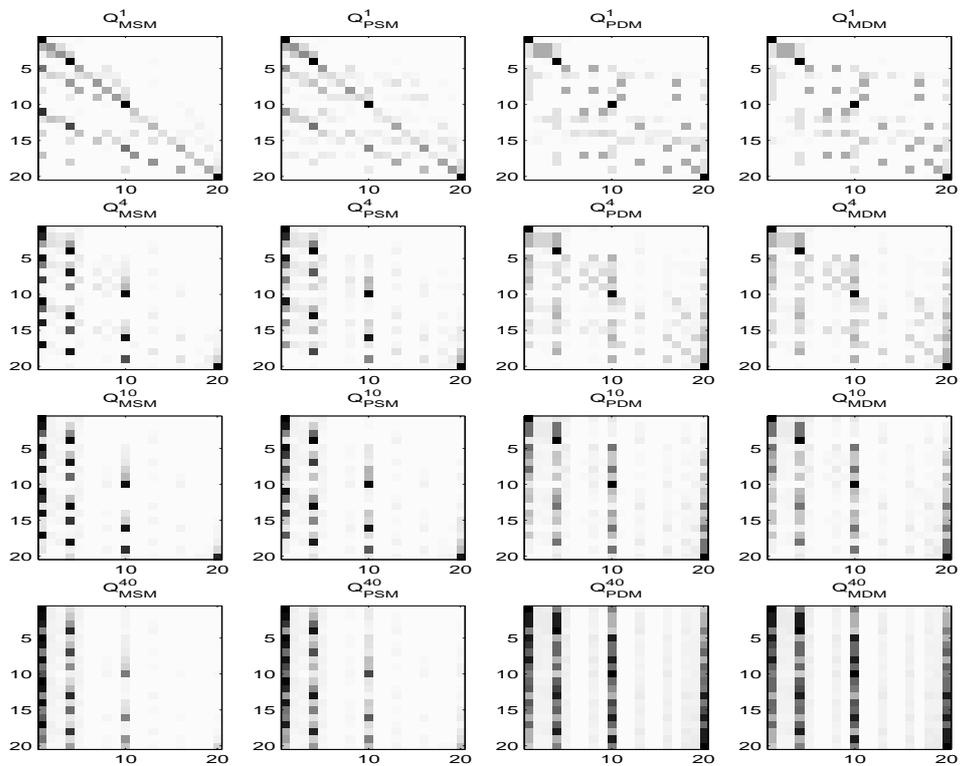


Figure 4.1: Visualization of Q^k ($l=2, n=3$).

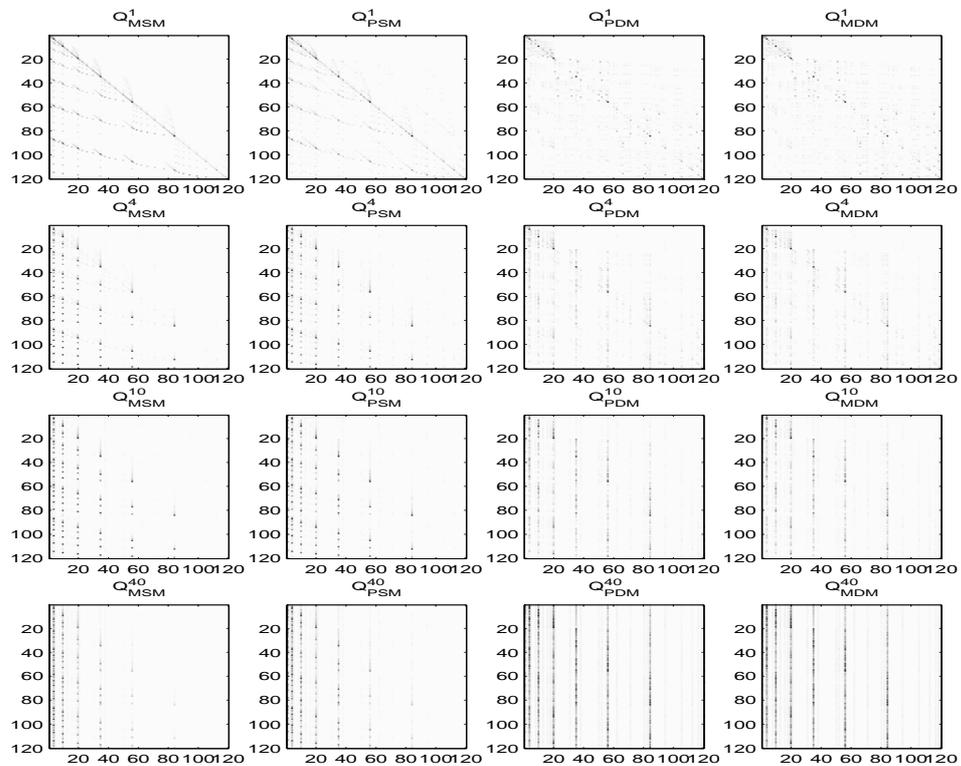


Figure 4.2: Visualization of Q^k ($l=3, n=3$).

As time proceeds, we see that the changes in the probability distribution are already evident in Q_m^4 and even more evident in Q_m^{10} and Q_m^{40} . The emerging vertical lines represent the particular populations at which the steady state distribution will accumulate most of its probability mass (i.e., the populations most likely to be observed when the GA settles into its dynamic equilibrium).

As one more example, Figure 4.2 shows the results for the case of string length 3, population size 3 (the number of total states is 120). In this figure, we can even more clearly see emergent vertical lines.

Upon a closer examination I find that dissimilar matings generally allow the GA to jump to more states than similar matings. For example, while the GA is in dynamic equilibrium, if one hopes to observe, at the next time step, to how many states the GA can jump from state 1 with probability greater than a threshold, say $\frac{1}{N} = .0082$ for the case of $l = 3$ and $n = 3$ (N is the number of total states), then it is 11 states for the maximum similar mating, and 22 states for the maximum dissimilar mating. Figure 4.3 shows more results for the numbers of the states that are of transition probability over threshold $\frac{1}{N}$ while the GA is in dynamic equilibrium and stays previously at state i ($i = 1, \dots, N$). (The top and the bottom plots correspond to $l = 2, n = 3$ and $l = 3, n = 3$, respectively.) All these results show that the GA with the maximum dissimilar mating can jump to more states than the GA with the maximum similar mating. That is, dissimilarity-based matings can make more state changes.

We can use the Markov model to further examine if dissimilarity-based mate selections would maintain larger population diversity. Recall that, in the preceding section, the matrix Z represents possible populations, which is an $N \times r$ matrix. (N is the number of possible populations and r is the total number of possible strings.) The i th row $\phi_i = \langle z_{i,0}, \dots, z_{i,r-1} \rangle$ of Z is the incidence vector for the i th population. $z_{i,y}$ is the number of occurrences of string y in the i th population (y is the integer

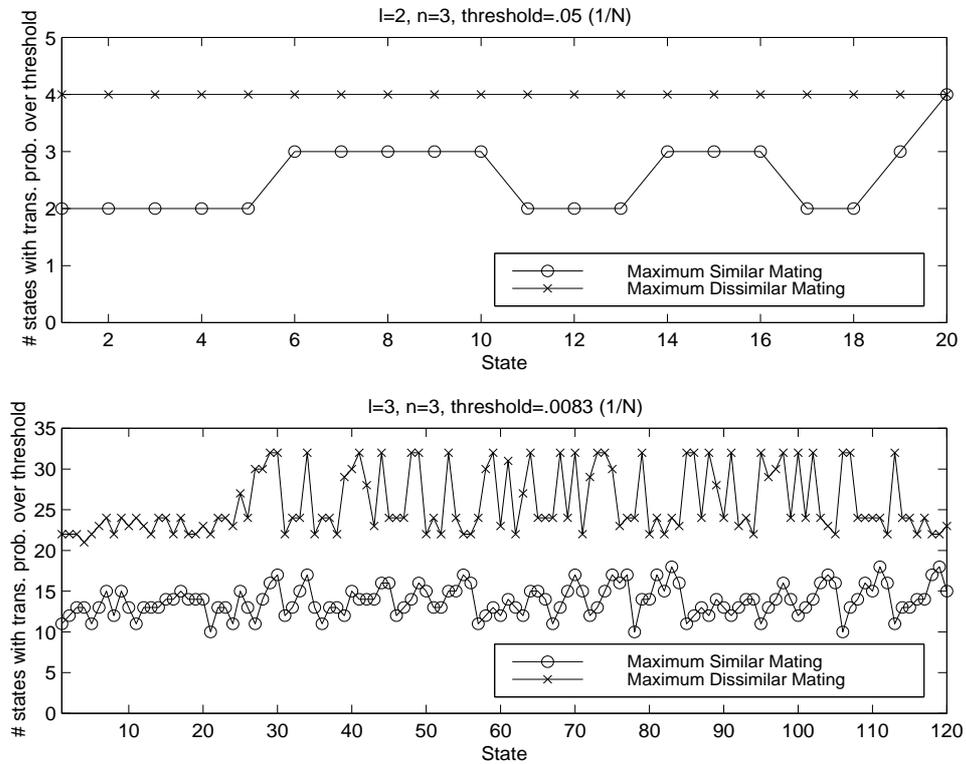


Figure 4.3: Number of states with transition probability over the threshold $\frac{1}{N}$.

representation of the binary string).

Let b denote the row vector of strings' binary representations, in which the i th component corresponds to string i 's binary representation ($i = 0, \dots, r - 1$). Then $b_{i,j}$ represents the bit value of string i 's locus j ($j = 1, \dots, l$). For instance, if binary strings are of length 2, then $b = \langle 00, 01, 10, 11 \rangle$, and $b_{0,1} = 0$, $b_{1,2} = 1$, $b_{3,1} = 1$ and so forth. To compute the proportion of 1s at each locus in the i th population, we need to extend the definition of vector inner product to include products of scalars and bit strings.

For example, $\phi_2 = \langle 0, 0, 1, 1 \rangle$ represents state P2 of Table 4.1, and its inner product with $b = \langle 00, 01, 10, 11 \rangle$ is:

$$\begin{aligned} \sum_{i=0}^3 z_{2,i} \cdot b_i &= 0 * (00) + 0 * (01) + 1 * (10) + 1 * (11) \\ &= (10) + (11). \end{aligned}$$

The result above shows that the number of 1s at the first locus (the right bit of strings) is 1, and that at the second locus (the left bit of strings) is 2. Dividing the number of 1s at each locus by population size 2 yields:

$$p_1 = 0.5 \quad \text{and} \quad p_2 = 1$$

where p_i represents the proportion of 1s at locus i .

Thus the inner product of ϕ_i and b , dividing by population size n , represents the proportion of 1s at each locus in the i th population.

By Equation (3.5) we can calculate the bitwise diversity $D_j, j = 1, \dots, l$. Then the average over all loci by Equation (3.6) yields the allelic diversity for the i th population, denoted as DIV_i . (I adopt DIV_i to avoid conflicting with the notations in Equations (3.5) and (3.6).)

Let DIV denote the column vector of populations' averaged allelic diversity, where DIV_i is the averaged allelic diversity corresponding to the i th population ($i = 1, \dots, N$). Then the expected allelic diversity at generation k is:

$$E(DIV) = (iniP \cdot Q^k) \cdot DIV, \tag{4.4}$$

where $iniP$ is a row vector whose i th component represents the probability of the GA being in state i at generation 0 (the initial generation), and thus $iniP \cdot Q^k$ is states' probability distribution at generation k . In this chapter, I consider randomly initialized GAs. Thus the probability of a GA being in state i at time 0, denoted as $P(X_0 = i)$, is:

$$P(X_0 = i) = \frac{n!}{z_{i,0}! \cdots z_{i,r-1}!} \left(\frac{1}{r}\right)^n.$$

Since there are r possible strings, each string has a probability of r^{-1} of occurring. There are n strings in the population, and the multinomial distribution takes into account the different ways the strings can be inserted into the population to generate a unique state (De Jong, Spears, and Gordon, 1994).

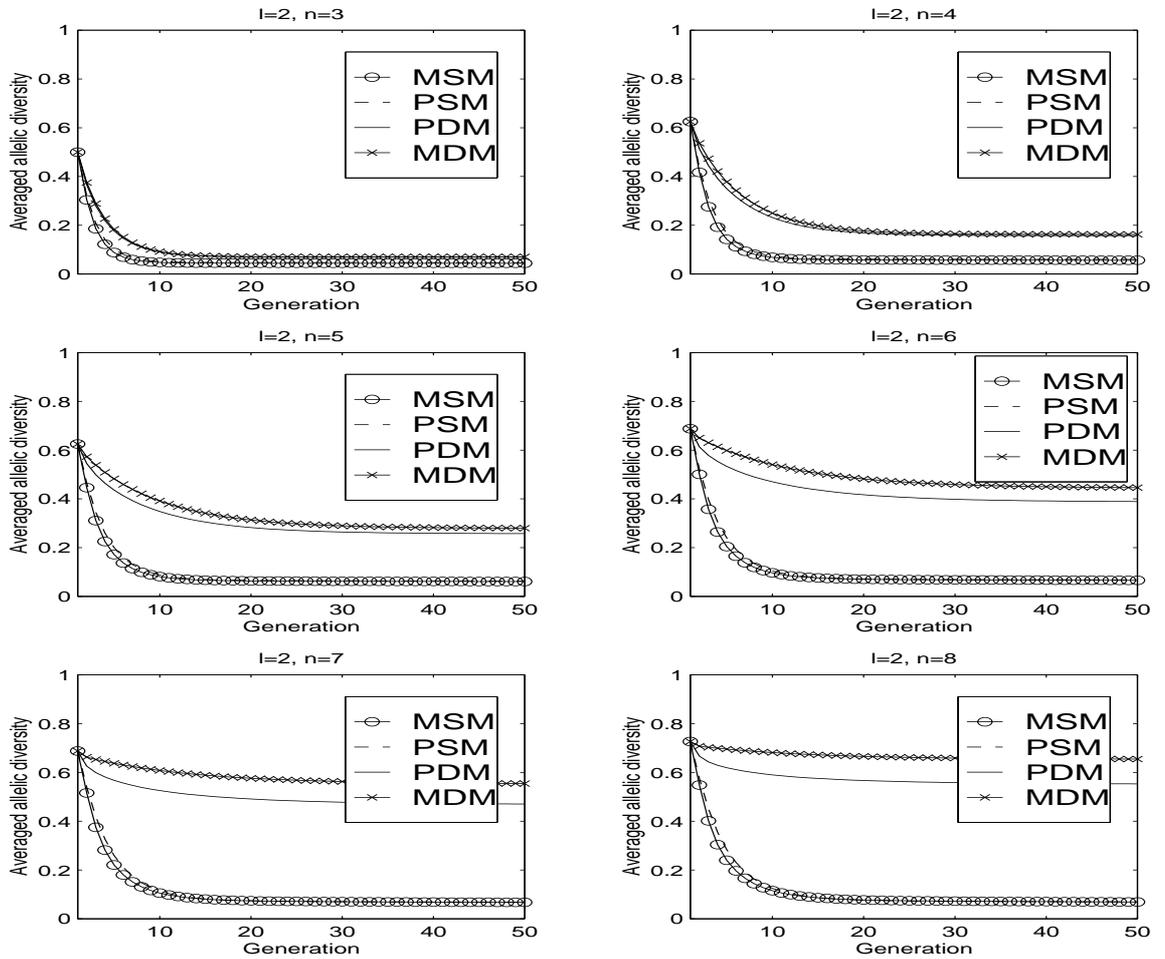


Figure 4.4: Averaged allelic diversity.

By Equation (4.4), we can compute the exact average allelic diversity for any cases. For illustrations, I still use the fitness function with $f(y) = \text{integer}(y) + 1$, where $\text{integer}(y)$ returns the integer equivalent of the bit string y . Figure 4.4 shows the results for string length 2 and several different population sizes ($n = 3, \dots, 8$), based on crossover rate 1 and mutation rate 0.01. The results show that the averaged allelic diversity of dissimilarity-based mate preferences is larger than that of similarity-based mate preferences. In particular, we can see that the difference between the dissimilarity and similarity-based mate selections enlarges as population size increases.

4.4 The GAFO Theory

In the foregoing chapter I was concerned with GA's search power in terms of the best-so-far performance. Such a performance metric is a form that a GA practitioner who is interested in function optimization would generally care about. For simple test functions the optimum is easily to locate and one may use the expected time (generations) to first encounter the optimum as a metric to compare different GAs' search power. In this section I use the GAFO (genetic algorithm function optimization) idea developed by De Jong et al. (1994) to examine the effects of different mate selection schemes on GA's performance.

4.4.1 Expected Waiting Time Analysis

De Jong et al. extended the Nix and Vose Markov chain analysis to provide answers for the expected waiting time until an event of interest is first observed. The observation is that the state transition matrix Q can be used to compute "mean first passage times" for going from state i to state j . If one is interested in knowing how long the GA would have to run on average before first reaching a set J of states, given that the process is currently in state i , then the expected waiting time is:

$$EWT(J) = \sum_{i \notin J} P(X_0 = i) m_{i,J},$$

where $m_{i,J}$ denotes the mean first passage time from state i to any of the states in set J , and i is not in J ; $m_{i,J}$ can be computed from the system of simultaneous equations

$$m_{i,J} = \sum_{j \in J} Q_{i,j} + \sum_{k \notin J} Q_{i,k} (1 + m_{k,J}).$$

If one defines J to be the set of states containing at least one copy of the optimum string, then $EWT(J)$ is the expected number of generations until the optimum is first encountered.

In the rest of this chapter I present some preliminary results to study effects of various factors on GA's EWT analysis. Characterizing these effects may provide

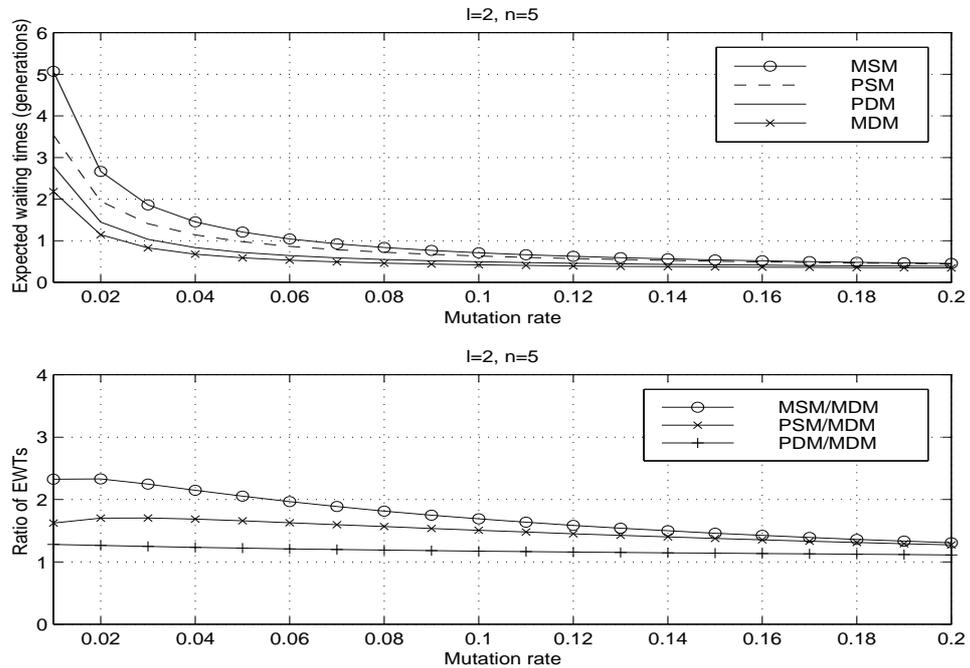


Figure 4.5: Interacting effects of mate selection and mutation.

useful insights and make predictions about how to improve GA's performance in the context of function optimization.

Interacting Effects of Mate Selection and Mutation

As an example, I again use the test function $f(y) = \text{integer}(y) + 1$, where $\text{integer}(y)$ returns the integer equivalent of the bit string y . Due to the computational limitation of the Nix and Vose model (as discussed in Section 4.2), I use the simplest possible case to proceed the investigation. In the following chapters, I will use more realistic population sizes and string lengths for further empirical study. In this example, the string length l is 2, population size n is 5, crossover rate is 1, and the optimum string is 11. For this case the first goal is to investigate the interacting effects of mate selection and mutation on the GA's EWTs to the optimum.

Figure 4.5 shows the results obtained for the maximum similar mating (MSM), the proportional similar mating (PSM), the proportional dissimilar mating (PDM), and the maximum dissimilar mating (MDM), based on mutation rates ranging from

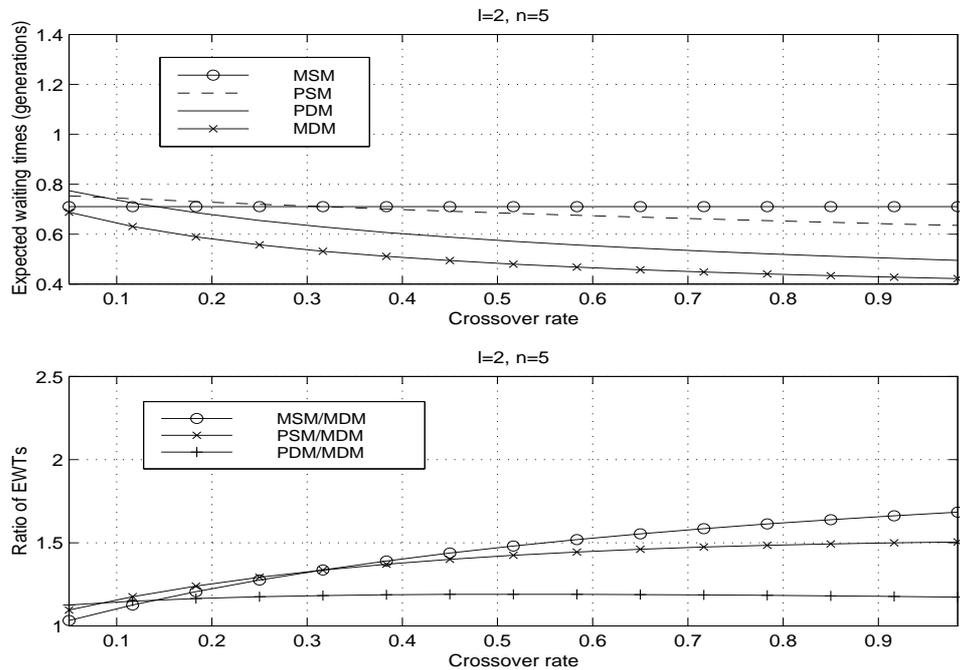


Figure 4.6: Interacting effects of mate selection and crossover.

0.01 to 0.2. The top plot is the exact EWTs and the bottom plot corresponds to the ratios of the EWTs from the MSM, the PSM, and the PDM to that from the MDM.

One can see that the maximum dissimilar mating generally has the lower expected waiting times than the other three. This agrees with the observations in the preceding chapter that dissimilar mating schemes generally outperform similar mating schemes if the goal is to find the best-so-far.

Note how dissimilar matings become increasingly important as mutation rate decreases (the larger the ratios, the better the relative performance of the maximum dissimilar mating). As mutation rate decreases, dissimilar mating would become a dominant factor that brings forth population diversity for further exploration of the search space.

Interacting Effects of Mate Selection and Crossover

One can also use these models to analyze the interacting effects of mate selection and crossover on EWTs. Figure 4.6 shows the results obtained for the four mate

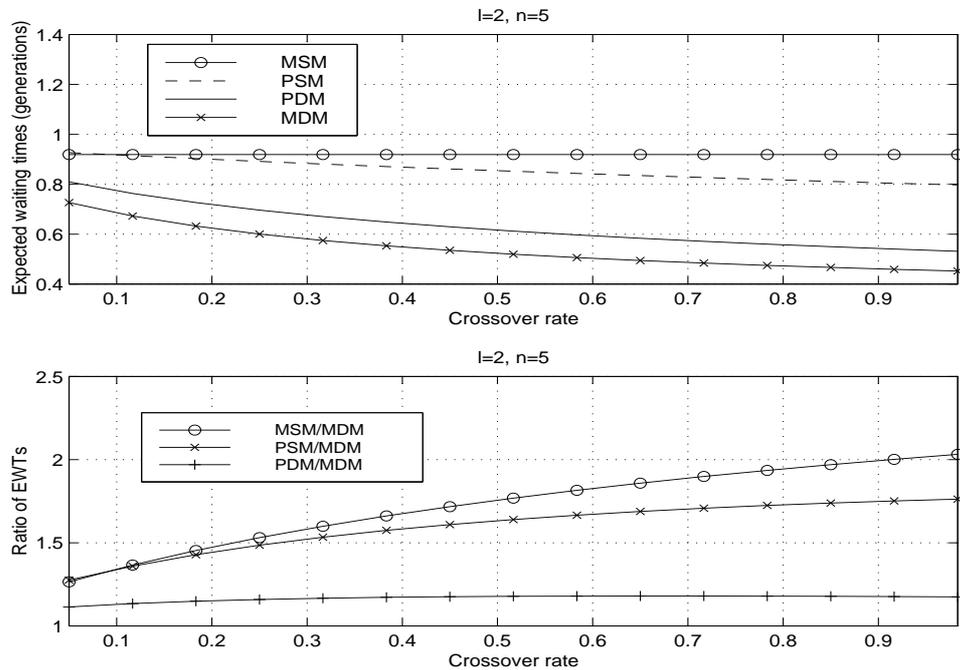


Figure 4.7: Effects of Scaling.

selection schemes, based on crossover rates ranging from 0.05 to 1. The top plot is for the exact EWTs and the bottom plot shows the ratios of the EWTs from the MSM, the PSM, and the PDM to that from the MDM.

One can see that the maximum dissimilar mating generally has the least expected waiting times than the other three. In particular, the dissimilar mating schemes demonstrate increasingly improved performance as crossover rate increases. This agrees with my intuition that both proper mate selection and crossover must operate together to enhance the power of simulating information exchange in GA's population.

Effects of Scaling

One can investigate how difficulties of test functions affect the GA's performance. It is well-known that proportional selection is sensitive to simple linear scaling of the fitness function (De Jong, Spears, and Gordon, 1994). For example, given the test function f and its variant, $g = f + 100$, one would expect that the GA is more incapable of locating the optimum of g , since the fitness of function g 's optimum is

relatively close to other fitness values.

Since MDM and PDM tend to choose dissimilar mating partners, essentially these two dissimilarity-based mate selections introduce additional effects of magnifying the difference between individuals. (See the simple example illustrated in Section 2.1.) But this is not the case for MSM and PSM. Therefore, if a GA is presented with the test function g , I predict that the performance difference between dissimilar mating (i.e., MDM and PDM) and similar mating (i.e., MSM and PSM) is larger than that with the test function f .

Figure 4.7 illustrates the EWT analysis for the four mate selection schemes based on the test function g . Compared with Figure 4.6, the performance discrepancy (the ratios of EWTs) between similar mating and dissimilar mating is enlarged, which confirms my hypothesis. These results show that the dissimilarity-based mate selection is beneficial in this more difficult function.

4.5 Summary and Discussions

This chapter presents my initial exploration of transient Markov chain analysis as a theoretical basis for the similarity and dissimilarity-based mate selection schemes.

Although closed form analysis is generally difficult, I gain useful insights by means of computational exploration of the transient behavior of the Markov models to characterize effects of various factors that affect GAs.

Visualizing the state transition matrix Q is a useful technique to see how the GA population becomes more diverse and diffuse in case of the dissimilar mate selections. I have also used the Markov model to investigate the effects of mate selection on population diversity. The results show that dissimilar mating generates larger population diversity than similar mating, which further supports the empirical results of population diversity obtained in the previous chapter.

As mutation rate decreases, the importance of mate selection in improving the

GA's performance becomes more salient. In addition, the study on the interacting effects of mate selection and crossover agrees with my intuition that both factors working together enhance the GA's search power. I have also demonstrated that the dissimilar mating schemes are beneficial in finding the optimum when test functions are relatively difficult.

A primary concern of Markov chain analysis is the scalability of the results. Due to the computational limitation of the Nix and Vose model, I will continue the study of mate selection through empirical methodologies in the following chapters, where more realistic population sizes and string lengths will be used.

There are a variety of directions worth exploring. For example, in addition to expected waiting times, the variance of the waiting times is also an important metric that can be derived from the mate selection Markov models. Since only the mating schemes based on fitness proportionate selection are studied in this chapter, in future work I also hope to develop a Markov model using tournament-based selection to further investigate the effects of mate selections based on tournament selection.

CHAPTER 5

BUILDING-BLOCK-BASED TEST FUNCTIONS

5.1 Introduction

In the preceding chapters, I have identified mate selection as an important factor that affects GA's search power. The results obtained show that dissimilarity-based mate preference facilitates the GA's exploration of the search space. Simply stated, the dissimilarity-based mate selections could get the GA away from hitchhiking and the founder effect, and further improve the GA's searching for best-so-far solutions.

In this chapter, I continue the study based on more complicated test functions for larger population sizes and string lengths. The investigation first concentrates on several original versions of the Royal Road functions, since these function are a class of building-block-based functions that serve as idealized testbeds for comparing effects of different mate selection schemes. I then continue testing the mate selection schemes on the hyperplane-defined functions, which extend the complexity of fitness landscapes so that the test functions may reflect more aspects of difficulties imposed on the GA's search power by real problems.

Based on the insights obtained from the preceding analysis, it is likely that dissimilar mating can improve the GA's search power in terms of locating best-so-far solutions. Thus I will concentrate on the comparison of the following mate selection schemes:

During each mating event, a binary tournament selection—with probability one the fitter of the two randomly sampled individuals is chosen—is run to pick out the first individual, then choosing the mate according to the following three different schemes:

Tournament Selection (TS): Run the binary tournament selection again to choose the mate.

Tournament Dissimilar Mating (TDM): Run the binary tournament selection two more times to choose two candidate partners; then the one more dissimilar to the first individual is selected for mating.

Random Dissimilar Mating (RDM): Randomly choose two candidate partners; then the one more dissimilar to the first individual is selected for mating.

I still use the Hamming distance as the similarity metric. Notice that in the mate selections above if the two candidates are of the same Hamming distance to the first individual, then one of them is randomly selected.

I show that the results obtained for small problems in Chapter 3 can be scaled to the original versions of the Royal Road (RR) functions, and to the more complicated hyperplane-defined functions. I will also show that on non-deceptive functions it is beneficial to first use the selection pressure from the externally imposed fitness function to select candidate mates, and then mate with the most dissimilar among those candidates (i.e., the tournament dissimilar scheme). On the other hand, on a deceptive test function, the results show that it is more beneficial that the dissimilar mate is selected from a number of candidates that are picked without the selection pressure from the externally imposed fitness function (i.e., the random dissimilar scheme).

In this chapter, the test functions first adopted are the Royal Road functions (Mitchell, Forrest, and Holland, 1992). I have indicated that the Royal Road functions

serve as idealized testbeds for comparing effects of different mate selection schemes. But it is clear that these test functions are inadequate in representing various fitness landscapes. For further testing these mate selections, I then adopt the hyperplane-defined functions (HDFs) (Holland, 2000), which increase the complexity of fitness landscapes such that the test functions are nonseparable, nonlinear, and nonsymmetric. These functions are hard to reverse engineer, but easy for analysis after the fact, as well.

5.2 Experiments on Royal Road Functions

In Chapter 3 I used several simple versions of the Royal Road functions for investigating basic properties of mate selections. These simple royal road functions facilitate the study for effects of distinct mating preferences on GAs. For example, the results showed that hitchhiking and the founder effect can be suppressed by means of dissimilarity-based mate selection strategies. In this chapter, I continue the investigation on effects of mate selections to see if the results obtained in Chapter 3 can be scaled to problems of larger string length.

The original idea of designing the Royal Road functions was to investigate in more detail the validity of the *Building Block Hypothesis*, and examine if GAs are a robust approach to all problems in which building blocks play a key role (Forrest and Mitchell, 1993). These functions are called Royal Road (RR) functions because they form a class of building-block-based functions in which improvements in the RR domain depend entirely on the discovery and exploitation of building blocks. Simply stated, the fitness landscapes of the Royal Road functions consist of two characteristics: the presence of short, low-order, highly-fit schemata and hierarchical structure which allow these small schemata to repeatedly construct more and more highly-fit schemata and eventually reach the global optimum.

As discussed in the first use of the simple Royal Road functions (in Section 3.4.1),

their building-block-based features serve as an idealized testbed for us to observe (1) how mate selection facilitates distinguishing individuals that carry necessary building blocks for further improvements, and then (2) how crossover brings these building blocks residing on separate strings into combination on a single string.

The second goal above was extensively investigated by Mitchell, et al. (1992) and Forrest et al. (1993). This thesis focuses on the investigation of the first goal. As the example illustrated in Section 2.1, necessary building blocks for improvements can be recognized by particular mating preferences; that is, the waiting times for identifying desirable schemata would be shorter if proper mate preferences are provided. Thus, I will focus on studying the following question: For a given landscape, what is the effect of mate selection on the waiting times for desirable schemata to be discovered? Answering this question in the context of the idealized Royal Road functions is a first step towards answering them for more general cases.

On the other hand, although the GA was expected to outperform mutation-based hill climbing search algorithms on the RR functions, the results obtained by Forrest et al. (1993) showed that the search power of GAs may be impeded by several factors. For example, if some intermediate stepping stones are much fitter than the primitive components, hitchhiking generates more severe problems that greatly hamper the discovery of some necessary schemata. Thus I will study if mate selection can remedy this problem, as well.

In short, based on the results obtained for small problems in Chapter 3, I gain insights that indicate dissimilarity-based mate selections could reduce the limitations imposed by hitchhiking and the founder effect on the GA's search power in terms of finding best-so-far solutions. In this section, the original versions of the Royal Road functions are used to study the questions mentioned above and examine if some results obtained in Chapter 3 can be scaled to these functions. The three functions used as testbeds in this section are: Royal Road R_1 , Royal Road R_2 , and

Table 5.1: Schematic of Royal Road R_1 .

$s_1 =$	11111111*****	$c_1 = 8$
$s_2 =$	*****11111111*****	$c_2 = 8$
$s_3 =$	*****11111111*****	$c_3 = 8$
$s_4 =$	*****11111111*****	$c_4 = 8$
$s_5 =$	*****11111111*****	$c_5 = 8$
$s_6 =$	*****11111111*****	$c_6 = 8$
$s_7 =$	*****11111111*****	$c_7 = 8$
$s_8 =$	*****11111111*****	$c_8 = 8$

the incompatible Royal Road IR_1 ,* where two of them are non-deceptive (R_1 and R_2), and the other is deceptive (IR_1). I will show that the tournament dissimilar mating scheme outperforms the other two on the non-deceptive functions, whereas the random dissimilar mating scheme is the best strategy that improves the GA's search performance on the deceptive function.

5.2.1 Royal Road R_1

Royal Road R_1 is the simplest of the Royal Road class test functions. Its fitness landscape is composed of eight consecutive building blocks of eight ones each. Table 5.1 shows the schematic of R_1 .

This function involves a set of schemata $S = \{s_1, \dots, s_8\}$ and the fitness of a bit string x is defined as

$$R_1(x) = \sum_{s \in S} c_s \sigma_s(x),$$

where each c_s is a value assigned to the schema s as defined in the table; $\sigma_s(x)$ is defined as 1 if x is an instance of s and 0 otherwise.

Thus, the fitness value $R_1(x)$ is the sum of the coefficients c_s corresponding to each given schema of which x is an instance. The fitness contribution from an intermediate stepping stone (such as the combination of s_1 and s_5) is a linear combination of the

* R_1 and R_2 are defined in (Mitchell, Forrest, and Holland, 1992); IR_1 is a larger version of the incompatible simple royal road function shown in Table 3.2.

Table 5.2: Mean function evaluations to the optimum or the number of runs out of 50 in which the optimum is found on R_1 .

Tournament Selection (TS)	43 runs found the optimum
Tournament Dissimilar Mating (TDM)	14647 (1017)
Random Dissimilar Mating (RDM)	19 runs found the optimum

fitness contribution of the lower-level components. For example, if x contains exactly two of the order-8 building blocks, $R_1(x) = 16$. Likewise, if x is the global optimum (64 1's), $R_1(111 \dots 1) = 64$. R_1 is similar to the “plateau” problem described by Schaffer and Eshelman (1991).

It is clear that Royal Road R_1 is a function in which the genes of building blocks are tightly linked and these building blocks are separable. It was therefore expected that GAs could perform quite well on such a fitness landscape because crossover can easily combine various building blocks residing on different strings (Mitchell, Forrest, and Holland, 1992).

The examination of the effects of mate selection on the Royal Road functions begins with comparing the time required to find the global optimum for GAs based on the three mating schemes: tournament selection (TS), the tournament dissimilar mating (TDM), and the random dissimilar mating (RDM).

For each of the three mating schemes, I conduct experiments of 50 runs based on one-point crossover rate 1, mutation rate 0.005, population size 128, and the maximum function evaluations in each run is 64000 (500 generations).

Table 5.2 summarizes the average function evaluations if the GA found the global optimum in all 50 runs. (The numbers in parentheses are the corresponding standard errors.[†]) If this not the case, the number of runs in which the GA found the global optimum is shown.

These results show that the tournament selection (TS) GA was outperformed by

[†]Standard error is calculated by: the standard deviation/ $\sqrt{\text{the number of runs}}$.

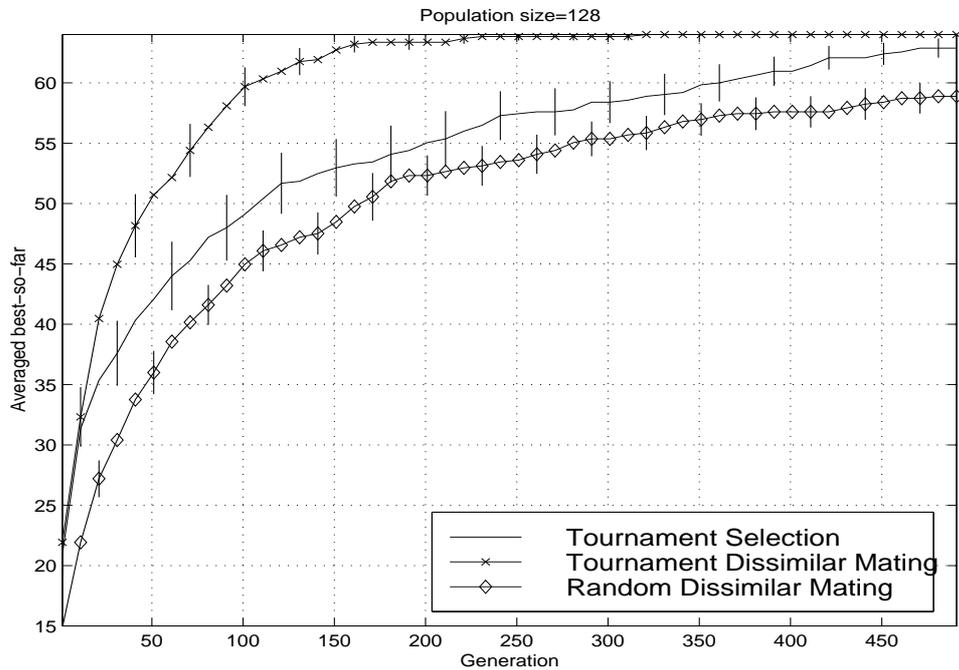


Figure 5.1: Best-so-far performance on R_1 .

the tournament dissimilar mating (TDM) GA; i.e., the additional dissimilarity-based mating preference considerably facilitates the GA’s discovery of the optimum. On the other hand, the random dissimilar mating (RDM) GA was outperformed by the other two GAs. This agrees with the results obtained in Section 3.4.6, where it was shown that the TDM GA outperformed the TS GA, and both of them outperformed the RDM GA on the non-deceptive testbed S_1 . (In the final part of this section, I will present the results for a deceptive test function in which RDM is more beneficial than the other two schemes.)

To further examine the performance difference, Figure 5.1 shows the best-so-far performance through generations, averaged over 50 runs. Between generation 110 and 120 the TDM GA usually has found the global optimum (see Table 5.2); and the fitness value of the averaged best-so-far is about 8 points larger than that obtained for the TS GA, a fitness value of a building block. This indicates that, compared with TS, TDM is able to significantly reduce the amount of time spent to effectively recognize the last building block for crossover to combine.

Table 5.3: The mean function evaluations of first appearance of a schema or schema combinations of increasing order for R_1 (over 50 runs).

	TS	TDM	RDM
Order 8	21 (3)	32 (4)	28 (4)
Order 16	591 (51)	571 (67)	856 (70)
Order 24	1574 (194)	1254 (102)	2391 (184)
Order 32	3267 (476)	2197 (177)	4483 (261)
Order 40	5617 (641)	3457 (262)	8220 (463)
Order 48	12620 (1581)	5586 (424)	15639(1057)
Order 56	20479 (1862)	8949 (589)	49 runs reached
Order 64	43 runs reached	14647 (1017)	19 runs reached

As a next step, we look more closely at the effect of mate selection on the GA's performance, considering the effects of different mate preferences on the waiting times for the various schemata defining the fitness function to be discovered. My intuition is that the dissimilarity-based test facilitates recognizing different string structures, which in turn reduces the waiting time for discovering a schema or schema combinations of increasing order.

Table 5.3 displays the average function evaluations over 50 runs at which the schema (or schema combinations) of a given order is first discovered by the GAs with and without mate selections, if the GA found the schemata of certain order in all 50 runs before the number of maximum evaluations (64000) is reached. (The numbers in parentheses are the corresponding standard errors.) If the GA did not find the given schema order in all 50 runs, the number of runs in which the GA found that schema order is shown. The results given in the table show that TDM significantly reduces the waiting time for discovering schemata at each order in the tree. Further inspection shows that the GA with RDM only reached order 64 in 19 runs during 64000 function evaluations, but the GA with TS can reach that order in 43 runs. This result shows how the TS GA found the optimum in 43 runs out of 50, and the RDM GA found the optimum in only 19 runs as shown in Table 5.2.

All the results so far indicate that on the non-deceptive function R_1 , TDM is a better strategy than TS, and TS is better than RDM. In other words, when an individual undergoes a mating event, it would be beneficial to first use the selection pressure from its environment (i.e., the externally imposed fitness function $R_1(x)$ in this case) to select candidate mates, and then mate with the most dissimilar among those candidates (this is what TDM does). On the other hand, if the dissimilar mate is selected from a number of candidates that are picked without the selection pressure from the environment (i.e., RDM in this case), then the GA will be outperformed by the GA without specific mate preference (i.e., TS in this case).

To understand why all this occurs, consider the nature of these mate selection strategies. Recall that the difference between TDM and TS is that TDM employs an additional strategy that facilitates distinguishing two strings that are of the same fitness, but of distinct string structures. This increases the likelihood of locating necessary, different schemata so as to speed up the discovery process for the optimum. On the other hand, RDM does not employ the selection pressure toward higher-fitness individuals while selecting the mating partners. Since R_1 is non-deceptive, the absence of such a selection pressure may in turn slow down the searching process for the optimum. The testing of this hypothesis is one of the central themes in the following parts of this thesis.

5.2.2 Royal Road R_2

The second Royal Road testbed used is the Royal Road function R_2 , where the fitness contributions of certain intermediate stepping stones are much higher. R_2 is illustrated in Figure 5.4. R_2 is calculated in the same way as R_1 : the fitness of a bit string x is the sum of the coefficients corresponding to each schema (s_1-s_{14}) of which it is an instance. For example, $R_2(1111111100\dots01111111) = 16$, since the string is an instance of both s_1 and s_8 , but $R_2(1111111111111100\dots0) = 32$, since the

Table 5.4: Schematic of Royal Road R_2 .

$s_1 = 11111111*****$; $c_1 = 8$
 $s_2 = *****11111111*****$; $c_2 = 8$
 $s_3 = *****11111111*****$; $c_3 = 8$
 $s_4 = *****11111111*****$; $c_4 = 8$
 $s_5 = *****11111111*****$; $c_5 = 8$
 $s_6 = *****11111111*****$; $c_6 = 8$
 $s_7 = *****11111111*****$; $c_7 = 8$
 $s_8 = *****11111111*****$; $c_8 = 8$
 $s_9 = 1111111111111111*****$; $c_9 = 16$
 $s_{10} = *****1111111111111111*****$; $c_{10} = 16$
 $s_{11} = *****1111111111111111*****$; $c_{11} = 16$
 $s_{12} = *****1111111111111111*****$; $c_{12} = 16$
 $s_{13} = 11111111111111111111111111111111*****$; $c_{13} = 32$
 $s_{14} = *****11111111111111111111111111111111*****$; $c_{14} = 32$

string is an instance of s_1 , s_2 , and s_9 . Thus, a string's fitness depends not only on the number of 8-bit schemata to which the string belongs, but also on their positions in the string. The fitness of the optimum string $11111111 \dots 1$ (64 1's) is 192, since the string is an instance of each schema in the tree.

The original motivation for designing Royal Road R_2 in (Forrest and Mitchell, 1993) was to examine if the step size of the intermediate stepping stones improves the GA's performance by means of allowing the fitness contributions of certain intermediate stepping stones to be much higher. However, Forrest et al. found that the much fitter intermediate stepping stones have adverse effect on the GA's search power, since hitchhiking is enhanced and induces more severe problems that greatly hamper the exploration of the search space.

My objective in this subsection is to examine if the dissimilarity-based mate selection can remedy this problem. To this end, I first conduct experiments for the GAs with TS, TDM and RDM, based on Royal Road R_2 . Then I compare the performance of the GAs on R_2 with that on R_1 to see if the dissimilarity-based mate preference can improve the GA's performance on R_2 .

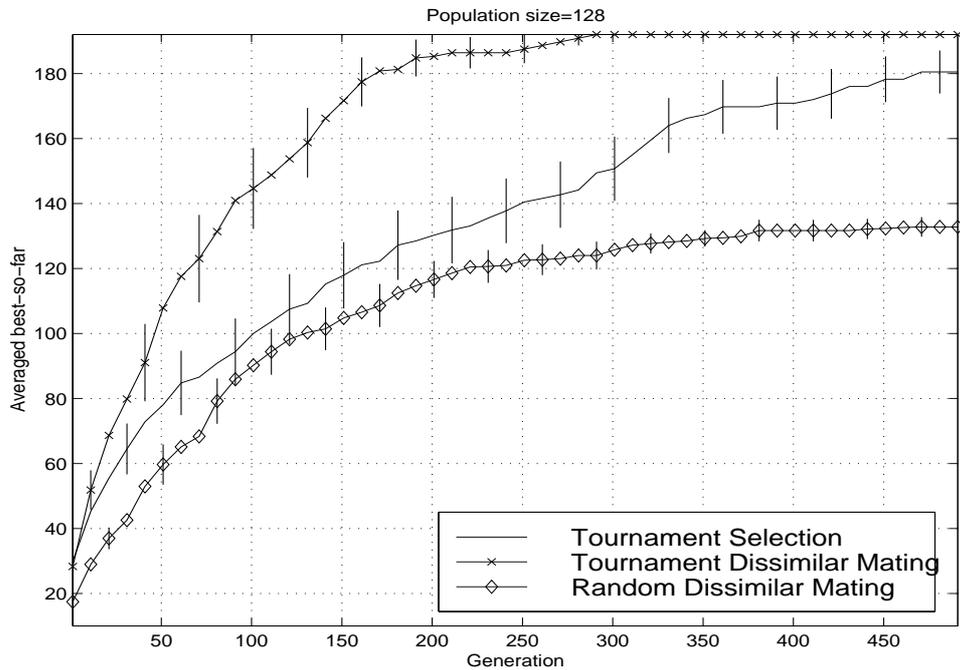


Figure 5.2: Best-so-far performance on R_2 .

For each of the three mate selection schemes, the experiments performed are based on one-point crossover rate 1, mutation rate 0.005, population size 128, and the number of maximum function evaluations allowed for each run is 64000 (equivalent to 500 generations).

Figure 5.2 shows the best-so-far performance through generations, averaged over 50 runs. As can be seen, the qualitative difference between the three mate selection schemes is similar to that in Figure 5.1, although the absolute performance difference is more obvious.

Based on the results obtained for R_1 and R_2 , it is clear that for these two non-deceptive functions, the combination of tournament selection and the dissimilar mating strategy (i.e., TDM) has better performance in terms of finding the best-so-far.

We can take a closer look at the effects of mate selection on the GA's performance, again considering the waiting times for the various schemata defining the fitness function to be discovered.

Table 5.5: The mean function evaluations of first appearance of a schema or schema combinations of increasing order for R_2 (over 50 runs).

	TS	TDM	RDM
Order 8	46 (9)	40 (9)	31 (4)
Order 16	759 (71)	827 (66)	1668 (177)
Order 24	1511 (109)	1452 (75)	3280 (227)
Order 32	3187 (528)	2657 (248)	8743 (826)
Order 40	7621 (1067)	4799 (491)	14073 (1142)
Order 48	14563 (1687)	7422 (606)	27044 (1769)
Order 56	48 runs reached	10815 (799)	29 runs reached
Order 64	40 runs reached	17515 (1180)	1 runs reached

Table 5.5 displays the average function evaluations over 50 runs at which the schema (or schema combinations) of a given order is first discovered by the GA, if the GA found the schemata of certain order in all 50 runs; if this is not the case, the number of runs in which the GA found that schema order is shown. The results given in the table show that, as expected, the TDM still significantly reduces the waiting time for discovering schemata at each order in the tree. Further inspection shows that the GA with RDM found order 56 in 29 runs, and order 64 in only one run before 64000 function evaluations are reached, but the GA with TS found order 56 in 48 runs and order 64 in 40 runs. This result accounts for the larger performance difference in Figure 5.2 than in Figure 5.1.

The next step I am interested in investigating is whether the dissimilarity-based mating strategy can remedy the more severe problem that hitchhiking imposes on R_2 than on R_1 . To do this, I would need to compare the performance of the GA on R_2 with its performance on R_1 for each mate selection scheme. By normalizing the best-so-fars (i.e., dividing these values by the value of the optimal string; e.g., for R_1 , the optimal fitness is 64, and for R_2 it is 192) and examining the ratio of the normalized best-so-far of R_2 to that of R_1 , we can compare different mating strategies. If the ratio approaches 1 for a given mate selection scheme, the degree of hitchhiking on

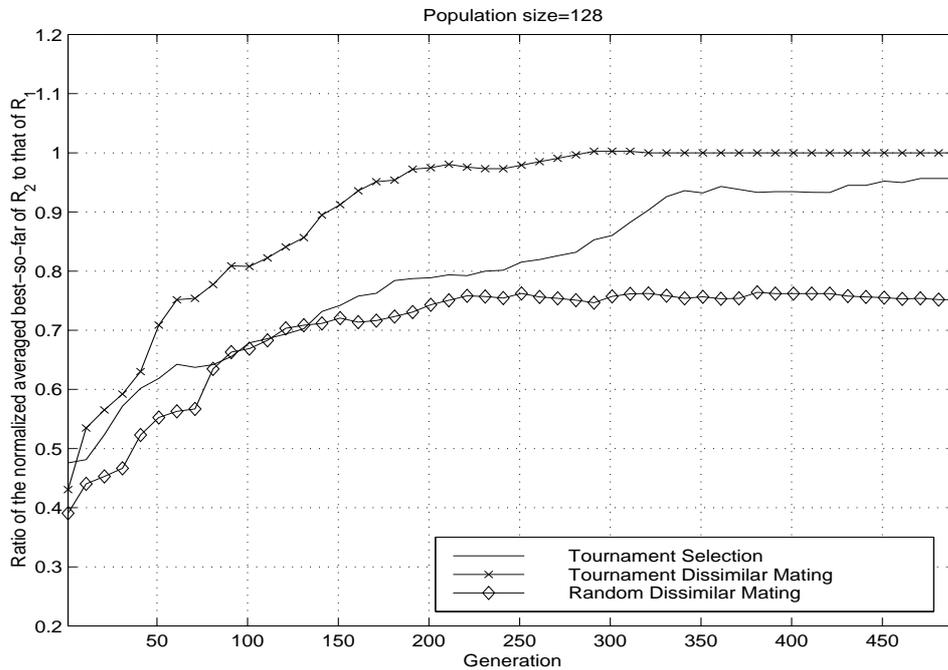


Figure 5.3: Ratio of the normalized averaged best-so-far of R_2 to that of R_1 .

R_2 is similar to that on R_1 . This would indicate that this mate selection effectively suppresses hitchhiking since the corresponding GA's performance on R_2 is not much different from on R_1 .

Figure 5.3 illustrates the result for the ratios obtained from Figure 5.2 and Figure 5.1. Since the GAs' initial populations for R_2 and R_1 are randomly generated, the mean individual fitness in the initial populations of the two cases are almost the same.[‡] But since they are normalized by different values (64 for R_1 and 192 for R_2), in the beginning generations the normalized best-so-far of R_2 is lower than that of R_1 . This results in the lower-than-one ratio curves as shown in the figure. As more generations go on, the ratio curve gradually picks up. In particular, the ratio curve corresponding to TDM eventually reaches one, but this situation does not occur to

[‡]The probability that a randomly generated string contains one of the bottom-level order-8 schemata is $8 \times \frac{1}{2^8} = \frac{1}{32}$. Since the initial population has 128 randomly generated individuals, there are on average $\frac{128}{32} = 4$ total instances of bottom-level schemata in the initial population. Also since the probability of order-16 schemata's presence in the initial population can be neglected, compared to that of order-8 schemata, the expected individual fitness for R_1 and R_2 are almost the same (i.e., $\frac{4 \times 8}{128} = .25$).

the other two schemes. These results show that TDM effectively suppresses hitchhiking on R_2 than the other two, and thus is the best strategy for the GA on such “non-deceptive” functions among the three schemes. (In the next subsection, I will show that RDM is more beneficial than TS and TDM on a “deceptive” test function.)

5.2.3 Incompatible Royal Road IR_1

As I have discussed in Section 3.4.2, a variant of the simple royal road function that consists of incompatible schemata (the test function IS_1) would generate the founder effect—in the presence of incompatible schemata, the first discovered of the incompatible schemata constrains the future evolutionary avenue and effectively precludes the testing of the other incompatible schema. Thus further improvements stem from the founder, making it progressively less likely that the other schema will influence the search process.

In this subsection, I continue to study the founder effect on the GAs’ performance, based on an incompatible Royal Road function, IR_1 . My objective is to examine if the dissimilarity-based mate selection can improve the GA’s search power. To this end, I first conduct experiments for the TS, TDM and RDM GAs. Then I examine the ratio of the normalized best-so-far of IR_1 to that of R_1 (as used in the preceding subsection) to see if the dissimilarity-based mate preference can reduce the founder effect.

Figure 5.6 is the schematic of a royal road function with incompatible schemata, IR_1 . IR_1 is calculated in the same way as R_1 : the fitness of a bit string x is the sum of the coefficients corresponding to each schema (s_1 – s_{14}) of which it is an instance. The fitness of the optimum string 11111111...1 (64 1’s) is 64, since the string is an instance of each schema in the tree. As can be seen, an incompatible schema may prevent its counterpart from being tested. For example, if s_2 comes to establish a large portion of the population, s_1 may be precluded for testing (although it is fitter

Table 5.6: Schematic of the incompatible Royal Road IR_1 .

```

s1 = 11111111*****; c1 = 8
s2 = 00000000*****; c2 = 5
s3 = *****11111111*****; c3 = 8
s4 = *****00000000*****; c4 = 5
s5 = *****11111111*****; c5 = 8
s6 = *****00000000*****; c6 = 5
s7 = *****11111111*****; c7 = 8
s8 = *****00000000*****; c8 = 5
s9 = *****11111111*****; c9 = 8
s10 = *****00000000*****; c10 = 5
s11 = *****11111111*****; c11 = 8
s12 = *****00000000*****; c12 = 5
s13 = *****11111111*****; c13 = 8
s14 = *****00000000*****; c14 = 5
s15 = *****11111111*****; c15 = 8
s16 = *****00000000*****; c16 = 5

```

than s_2). As a consequence, some extent of deceptiveness is introduced that prevents the GA from finding the global optimum (64 1's).

For each of the three mate selection schemes, the experiments performed are again based on one-point crossover rate 1, mutation rate 0.005, population size 128, and the number of maximum function evaluations allowed for each run is 64000.

Figure 5.4 illustrates the best-so-far performance through generations, averaged over 50 runs. The qualitative difference between the results for the three mate selection schemes in this figure is different from those in Figure 5.1 and Figure 5.2. We see that the best-so-far curves of the TDM and TS GAs go up faster than that of the RDM GA and soon get stuck, yet the curve of the RDM GA still keeps improving. Therefore, if experiments are allowed to run longer, RDM could be the best choice in improving the GA's performance. If it is not the case, then TDM is more beneficial than the other two schemes.[§]

[§]The degree of deceptiveness can be represented by the values c_i for $i = 2, 4, 6, \dots, 16$. For example, $c_2 = 5$ and $c_2 = 1$ represent two different versions of the deceptive functions. But the experiments I have conducted so far show similar qualitative results for the three mate selections.

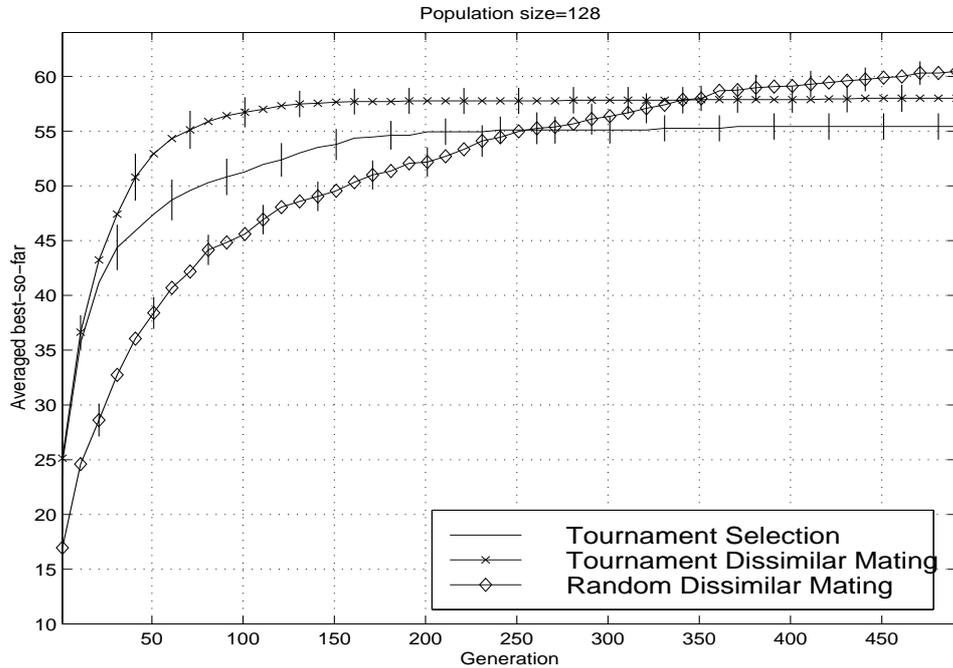


Figure 5.4: Best-so-far performance on incompatible IR_1 .

To understand why this occurs, we can again consider the nature of these mate selection strategies. Recall that the difference between TDM and TS is that TDM always chooses dissimilar mates. This generates an increased likelihood that mating partners come from distant parts of the search space, and the resulting matings could get offspring out of the local optima. This in turn facilitates the GA’s searching different parts of the search space, and increases the likelihood of exploring other optima of higher fitnesses. On the other hand, RDM differs from TDM by removing the selection pressure toward higher-fitness individuals while selecting mating partners. This can reduce the degree of convergence on certain local optima, and further the exploration of the search space. Simply stated, both TDM and RDM are able to facilitate the GA’s exploration of the search space, the difference being that RDM improves the best-so-far attained at the expense of a slower searching process. This

I suspect this is because the tournament-based selections are used. For other background selection schemes, such as fitness proportionate selection, the results might show that RDM is not beneficial on test functions of moderate deceptiveness. This direction will not be explored in this thesis, but will be deferred for future work.

Table 5.7: The mean function evaluations of first appearance of a schema or schema combinations of increasing order for IR_1 (over 50 runs).

	TS	TDM	RDM
Order 8	50 (16)	37 (11)	39 (6)
Order 16	674 (108)	657 (152)	970 (130)
Order 24	49 runs reached	1642 (218)	2624 (223)
Order 32	43 runs reached	47 runs reached	5437 (416)
Order 40	33 runs reached	42 runs reached	10044 (800)
Order 48	21 runs reached	33 runs reached	49 runs reached
Order 56	10 runs reached	19 runs reached	46 runs reached
Order 64	1 runs reached	9 runs reached	23 runs reached

phenomenon will be seen throughout the rest of this thesis.

We can further examine the effects of mate selection on the GA's performance by comparing the waiting times for the various schemata to be discovered.

Table 5.7 summarizes the average function evaluations over 50 runs at which the schema (or schema combinations) of a given order is first discovered by the GA, if it found the schemata of certain order in all 50 runs. (The numbers in parentheses are the corresponding standard errors.) If this is not the case, the number of runs in which the GA found that schema order is shown. As can be seen, for lower-order schemata (e.g., orders 8 and 16), the TDM and TS GAs found them more rapidly than the RDM GA. But for higher-order schemata, the results show that there are increasing runs in which the TDM and TS GAs did not find those schemata. For schemata of order 40, the RDM GA still found them in all 50 runs, yet it is not the case for the TS and TDM GAs. For orders more than 40 it is clear that RDM has more runs in which each of these orders is reached than can be reached by TDM and TS. This result accounts for the performance difference in Figure 5.4.

The next step I am interested in investigating is whether the dissimilarity-based mating strategy can suppress the founder effect on IR_1 . To do this, I would need to compare the performance of the GA on IR_1 with its performance on R_1 for each

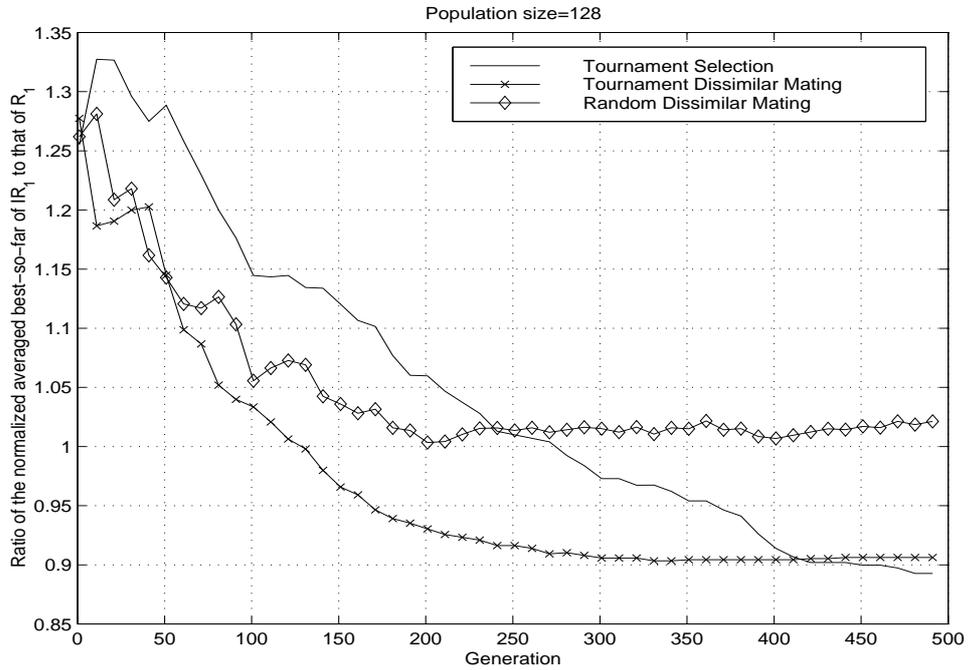


Figure 5.5: Ratio of the normalized averaged best-so-far of IR_1 to that of R_1 .

mate selection scheme. As in the study for hitchhiking on R_2 , by normalizing the best-so-far (i.e., dividing its fitness by the fitness value 64 of the optimal string for both IR_1 and R_1) as well as examining the ratio of the normalized best-so-far of IR_1 to that of R_1 , we can compare effects of mating strategies on the GA's performance. If the ratio approaches 1 for a given mate selection scheme, it then reduces the founder effect since the corresponding GA's performance on IR_1 is not much different from on R_1 .

Figure 5.5 illustrates the result for the ratios obtained from Figure 5.4 and Figure 5.1. Since the initial population has 128 randomly generated individuals, for IR_1 there are on average $\frac{128}{32} = 4$ total instances of schemata s_i for $i = 1, 3, 5, \dots, 15$ and of s_i for $i = 2, 4, 6, \dots, 16$, respectively. The mean individual fitness is then $\frac{4 \times 8}{128} + \frac{4 \times 5}{128} = .40625$. But for R_1 there are only an average of 4 total instances of the bottom-level order-8 schemata, and the mean individual fitness is $\frac{4 \times 8}{128} = .25$. Thus the expected fitness of an individual in the initial population for IR_1 is larger than that for R_1 . Since for these two cases, the best-so-far are normalized by the same

value, i.e., 64, which results in the higher-than-one, normalized ratio curves as shown in the figure. As more generations go on, the GA's population gradually converges. This reduces the likelihood of simultaneous presence of two incompatible schemata and then reduces the values of the ratios. Thereby the ratio curves fall down.

We can further see that when the searching process goes on, say after generation 300, the curve corresponding to the RDM GA approaches 1, yet the curves for TS and TDM fall below 1. This means that RDM effectively reduces the founder effect than the other two schemes, and thus is a better strategy for the GA's search on such a "deceptive" function.

In the next section, I will continue the study of mate selection based on more complicated testbeds (the hyperplane-defined functions), where we can see that RDM, in general, outperforms the other two schemes on such test functions.

5.3 Experiments on Hyperplane Defined Functions

The Royal Road functions previously used are a class of idealized building-block-based test functions. They facilitate investigating hitchhiking and the founder effect on the GA. For more testing of the GA's search power they are clearly inadequate since fitness landscapes of real problems may be highly nonlinear, nonseparable, and nonsymmetric. To more rigorously test the GA, Holland (2000) designed a class of functions—the hyperplane defined functions (HDFs), which capture several characteristics of good test functions (Whitley, Rana, Dzubera, and Mathias, 1996):

- test functions can be generated at random and are difficult to reverse engineer, so that the algorithms being tested do not inadvertently or deliberately exploit incidental features;
- test functions are capable of exhibiting an array of landscape-like features ("hills," "potholes," "badlands," "ridges," etc.) in controllable proportions,

so that one can examine what parts of the landscape are exploited by different genetic operators;

- test functions include all finite functions in the limit.

The Holland-style hyperplane-defined functions can be briefly described as follows.

The search space X over which the hyperplane-defined functions are defined is the set of all strings of length n (e.g., binary strings). The object is to provide a class of fitness functions generated entirely by values assigned to a broad range of schemata (hyperplanes) of different lengths and defining bits. The value (fitness) of each string is determined by the schemata present in the string; i.e., strings are scored in terms of the total increments and decrements provided by the schemata containing the string.

With each specific hyperplane-defined function f there is an associated set of schemata S for determining $f(x)$ (the fitness value of a string x). Each schema $s \in S$ is assigned a fundamental value $u(s)$ that can be either positive or negative. In the simplest version, the value of an x under f is simply the linear combination of the fitness contribution of all the schemata of which x is an instance. That is,

$$f(x) = \max\{0, \sum_{x \in s | s \in S} u(s)\}. \quad (5.1)$$

The set S can be created, via a proper random number generator, with a given distribution over the set H of all schemata, and the values $u(s)$ can be similarly assigned with a predefined distribution. Under this setup, the hyperplane-defined functions can be used as an infinite set of difficult functions. They are then easy to generate, hard to reverse-engineer, and easy to analyze after the fact by tracking the dynamics of the building blocks.

The HDFs also satisfy the following additional constraints:

- A set E of elementary schemata is chosen. These schemata are short relative to string length, but long enough to be rare in a randomly generated population.

The schemata in E may overlap and they may be incompatible at some common loci.

- Pairs of elementary schemata that are close to each other are selected at random and are combined in pairs to yield higher order schemata. These pairs are in turn combined in pairs to yield still higher order schemata, and so on until there are schemata of a length close to string length.
- Some schemata are generated by adding one or more additional defining bits to some elementary schema, where the additional bits are chosen to match the locus and value of some nearby elementary schema. The values assigned to these new schemata are selected to be less than the value of the elementary schemata involved in their definition, so that they constitute “valleys” that must be crossed to get from elementary schemata to higher order schemata.

In this section I intend to use three hyperplane-defined functions, with increasing complexity to investigate effects of mate selection.

5.3.1 Hyperplane-Defined Test Function H_1

The first hyperplane-defined function H_1 was generated to contain eight elementary schemata of various lengths for the lowest level of hierarchy and three levels of hierarchy were built upon, i.e, the equivalent of three levels of Royal Road functions. There are 29 schemata in total in this function and it can be characterized as follows (see appendix A for a detailed description of H_1):

- There are eight elementary schemata: s_1 – s_8 ; the value $u(s_i)$, $i = 1 \dots 8$, for each elementary schema is an integer randomly picked from the range of $\{4, 6\}$.
- There are four second-level schemata (s_9 – s_{12}) that are combinants of two elementary schemata; and there are three third-level schemata (s_{13} – s_{15}) that are

Table 5.8: Number of runs reached fitness value 80 in 100 runs on H_1 .

Mate Selection Scheme	Number of runs reached 80
Tournament Selection (TS)	38
Tournament Dissimilar Mating (TDM)	55
Random Dissimilar Mating (RDM)	67

combinants of three elementary schemata; the value $u(s_i)$, $i = 9 \dots 15$, is an integer randomly picked from the range of $\{10, 12\}$.

- There are fourteen pothole schemata ($s_{16}-s_{29}$) that are refinements of some elementary schemata; all $u(s_i)$, $i = 16 \dots 29$, is of value -1. (See (Belding, 2001) for a study of potholes.)

Notice that potholes present difficulty to the GA's search for higher order schemata if the potholes are overlapping with them. These potholes would then act like "valleys" (as discussed previously) that must be crossed by the GA to further explore the search space.

The experiments for H_1 are conducted based on one-point crossover rate 1, mutation rate 0.005, population size 100, and the number of maximum function evaluations allowed is 30,000.[¶] The GAs with TS, TDM and RDM are tested, and the results show that the highest best-so-far value reached by the GAs is 80 in all the experiments conducted for H_1 . (The GA with each mate selection scheme is tested for 100 runs). Table 5.8 summarizes the number of runs out of 100 in which the GA reaches the best-so-far of 80.

Since the potholes and some incompatible schemata on H_1 (e.g., s_4 and s_5 have an incompatible allele at the common locus 45) present difficulty such as deceptiveness to the GA's search, based on the results obtained in the previous section I expect

[¶]I use 30,000 function evaluations for H_1 because it turned out to serve well for displaying the difference between the mate selection schemes.

RDM to outperform the other two schemes. The results in Table 5.8 indeed show that RDM performs best.

A closer look at detailed schema dynamics illustrates a clear example of the founder effect that more easily occurs in the GA without dissimilarity-based mating preference (i.e., the tournament selection GA). In a run of the TS GA the search process was stuck rather quickly—after finding a string of value 29 at 1377 evaluations, there was no further improvement through 30,000 evaluations. Investigation of H_1 in that run revealed a pair of overlapping incompatible schemata, each of which was a starting point for a distinct sequence of improvements leading through higher levels. One of the sequences “dead-ended” at the string with value 29. In fact, in the 100 runs conducted for the TS GA, through 30,000 evaluations, the GA was constrained by the founder effect at a string with value 23 for three runs, and 29 for seven runs. However, for the RDM GA, the GA was constrained by the founder effect at a string with value 25 for only one run, and 29 for another one run, as well. These results can be more clearly illustrated by the best-so-far distribution collected for the 100 runs of experiments for each mating preference. Figure 5.6 illustrates the histogram that shows the best-so-far distribution when the number of maximum function evaluation is reached. The trend in these results shows that the RDM GA tends to locate higher best-so-fars than the TDM GA, and likewise the TDM GA tends to locate higher best-so-fars than the TS GA. These results again show us how RDM is more beneficial than TS and TDM.

We can further examine the GA’s performance in terms of the best-so-far dynamics. Figure 5.7 presents the averaged best-so-far found at each generation. The results show in more detail how the RDM GA outperforms the other two; and the TDM GA still outperforms the TS GA. These results are consistent with what I obtained in the foregoing section. In the rest of this section, I use another two hyperplane-defined test

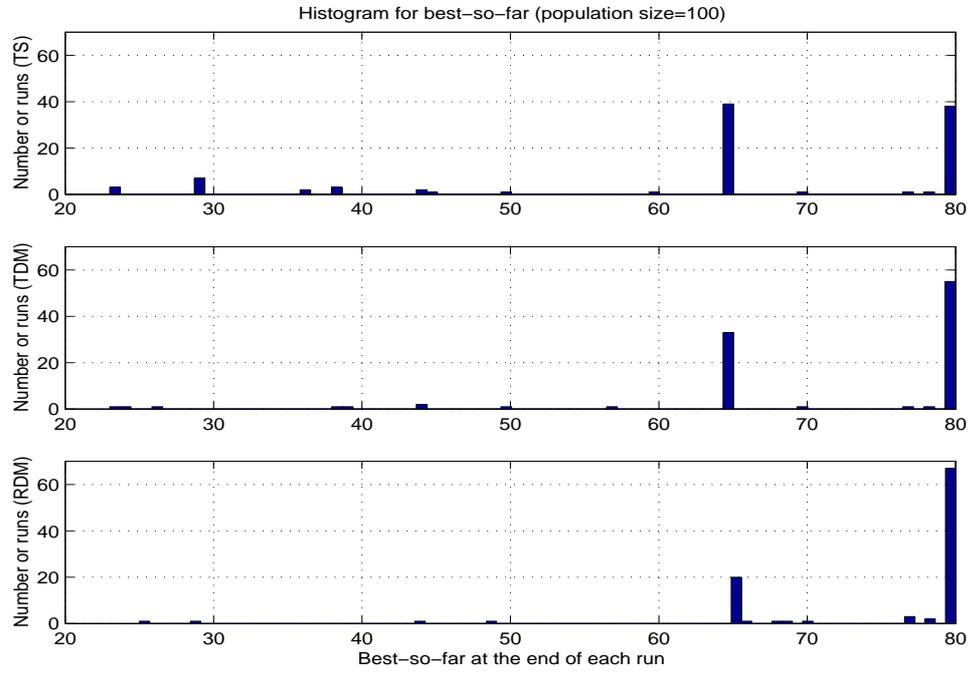


Figure 5.6: Histogram for best-so-far performance on H_1 .

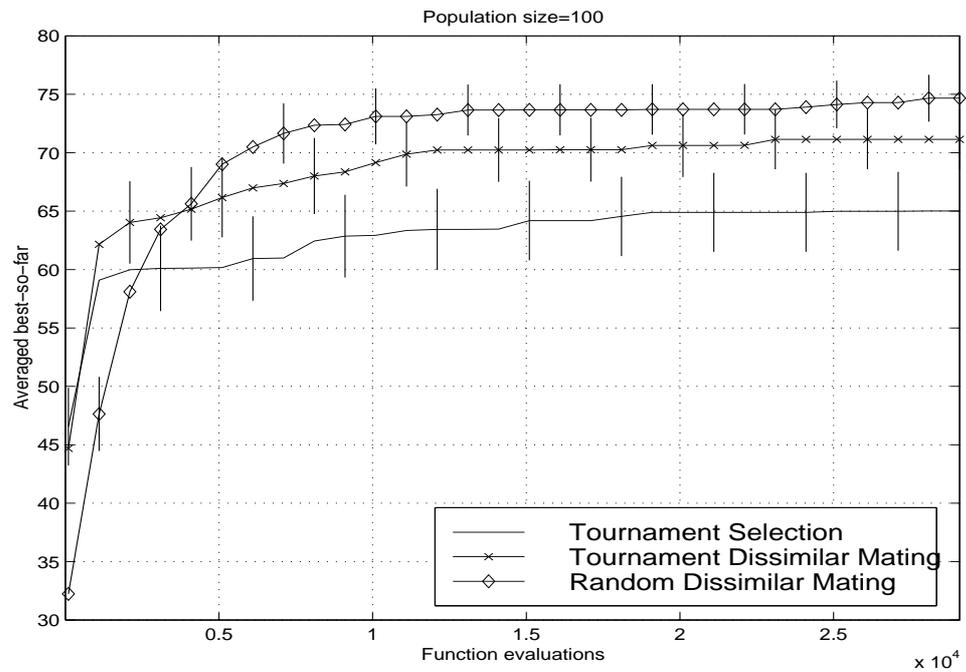


Figure 5.7: Best-so-far performance on H_1 .

functions for more investigation on the superior performance of RDM to the other two schemes.

5.3.2 Hyperplane-Defined Test Function H_2

The second hyperplane-defined function H_2 is a more complicated testbed that is generated to contain eight elementary schemata, eight levels of hierarchy. More levels of hierarchy implies that hitchhiking and the founder effect may present more difficulty to the GA's search, which allows us to further test RDM's power. The string length generated in this case is 200. There are 49 schemata in total in this function and it is characterized in the following (see appendix B for a detailed description of H_2):

- There are eight elementary schemata: s_1-s_8 ; the value $u(s_i)$, $i = 1 \dots 8$, for each elementary schema is an integer randomly picked from the range of $\{5, 6\}$.
- There are 27 combinants (s_9-s_{35}) in total for levels 2 to 8; the value $u(s_i)$, $i = 9 \dots 35$, is an integer randomly picked from the range of $\{8, 10\}$.
- There are fourteen pothole schemata ($s_{36}-s_{49}$) that are refinements of some elementary schemata; $u(s_i)$, $i = 36 \dots 49$, is an integer randomly picked from the range of $\{-2, -1\}$.

Again, the experiments for H_2 are conducted based on one-point crossover rate 1, mutation rate 0.005, population size 300, and the number of maximum function evaluations allowed is 60000.^{||} The GAs with the three mate selection schemes are tested. The results show that the highest best-so-far value reached by the GAs is 221 in all the experiments conducted for H_2 . I summarize the number out of 100 runs in which the GA reached the best-so-far of 221 in Table 5.9.

^{||}The testbed H_2 is generated with higher complexity than H_1 , and the string length is also longer than that used for H_1 ; thus I use the larger population size, 300, and maximum function evaluations, 60000. It turned out that 60000 function evaluations are sufficient for displaying the difference between the mate selection schemes.

Table 5.9: Number of runs reached fitness value 221 in 100 runs on H_2 .

Mate selection scheme	Number of runs reached 221
Tournament Selection (TS)	52
Tournament Dissimilar Mating (TDM)	60
Random Dissimilar Mating (RDM)	81

The qualitative difference between the three mate selection schemes shown in the table is similar to that on H_1 —RDM still facilitates the GA’s search for higher best-so-fars.

Figure 5.8 further illustrates the histogram that shows the best-so-far distribution when the number of maximum function evaluation is reached. The trend in these results again shows that the RDM GA tends to locate higher best-so-fars than the TDM GA, and likewise the TDM GA tends to locate higher best-so-fars than the TS GA. These results also show us how RDM outperforms TS and TDM.

We can further examine the GA’s performance by comparing the averaged best-so-far dynamics. Figure 5.9 presents the averaged best-so-far found at each generation. The results again show that the RDM GA outperforms the other two; and TDM outperforms TS. This is consistent with what I obtained for H_1 .

5.3.3 Hyperplane-Defined Test Function H_3

As the third testbed, a even more complicated hyperplane-defined function H_3 is generated to contain ten elementary schemata and eight levels of hierarchy. The string length used in this case is 400. My objective is to examine if the results obtained previously can be scaled to such a long string length case. There are 115 schemata in total in this function and it can be described by the following characteristics (see appendix C for a detailed description of H_3):

- There are ten elementary schemata: s_1 – s_{10} ; the value $u(s_i)$, $i = 1 \dots 10$, for each elementary schema is an integer randomly picked from the range of $\{5, 6\}$.

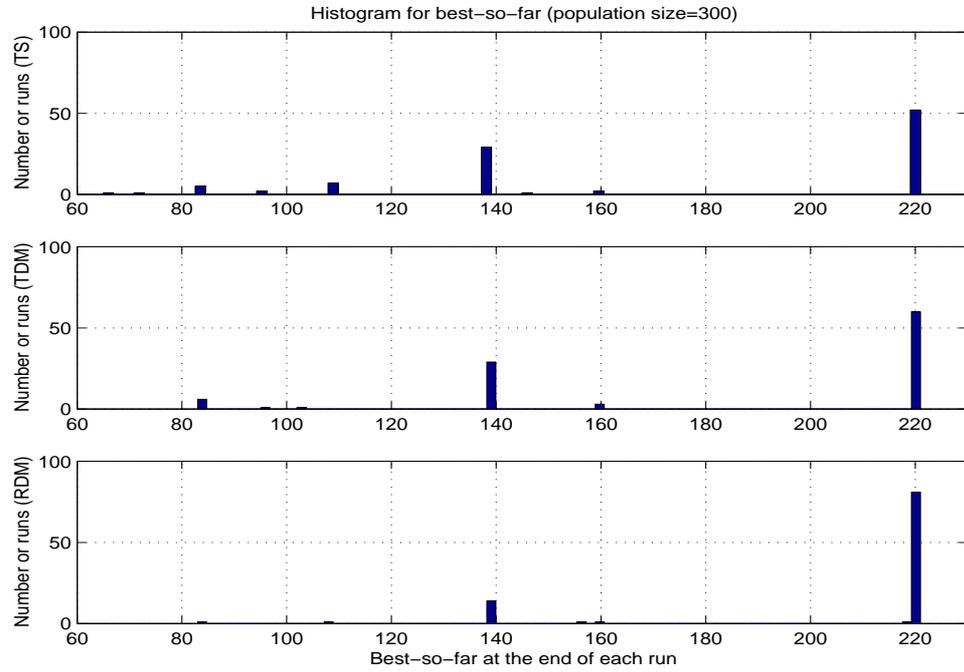


Figure 5.8: Histogram for best-so-far performance on H_2 .

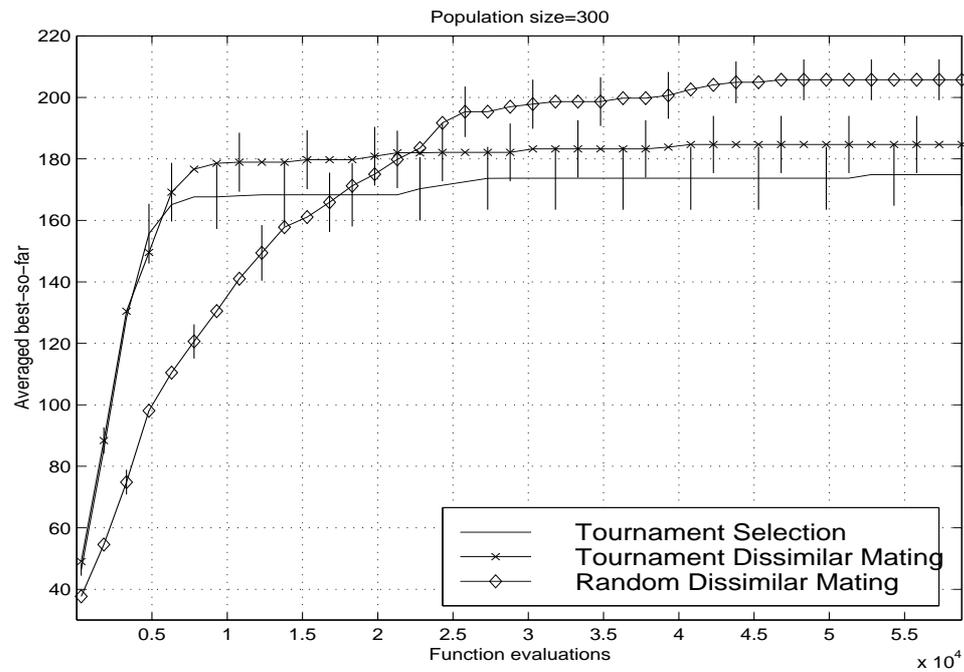


Figure 5.9: Best-so-far performance on H_2 .

Table 5.10: Number of runs reached fitness value 430 in 100 runs on H_3 .

Mate selection scheme	Number of runs reached 430
Tournament Selection (TS)	21
Tournament Dissimilar Mating (TDM)	26
Random Dissimilar Mating (RDM)	53

- There are 55 combinants (s_{11} – s_{65}) in total for levels 2 to 8; the value $u(s_i)$, $i = 11 \dots 65$, is an integer randomly picked from the range of $\{8, 10\}$.
- There are 50 pothole schemata (s_{66} – s_{115}) that are refinements of some elementary schemata; $u(s_i)$, $i = 66 \dots 115$, is an integer randomly picked from the range of $\{-2, -1\}$.

Again, the experiments for H_3 are conducted based on one-point crossover rate 1, mutation rate 0.005, population size 500, and the number of maximum function evaluations allowed is 100,000.** The results show that the highest best-so-far value reached by the GAs is 430 in all the experiments conducted for H_3 . The number out of 100 runs in which the GA reached the best-so-far of 430 is summarized in Table 5.10.

The qualitative difference between the three mate selection schemes shown in the table is similar to that on H_1 and H_2 —RDM is still more beneficial than the other two schemes.

Figure 5.10 illustrates the histogram for the best-so-far distribution until the maximum function evaluation. Again, the trend in these results shows that the RDM GA tends to locate higher best-so-fars than the TDM GA, and likewise the TDM GA

**Again, since H_3 is more complicated than H_2 and the string length used is longer than that used for H_2 , I use the larger population size, 500, and maximum function evaluations, 100000. It turned out that 100000 function evaluations are sufficient for displaying the difference between the mate selection schemes.

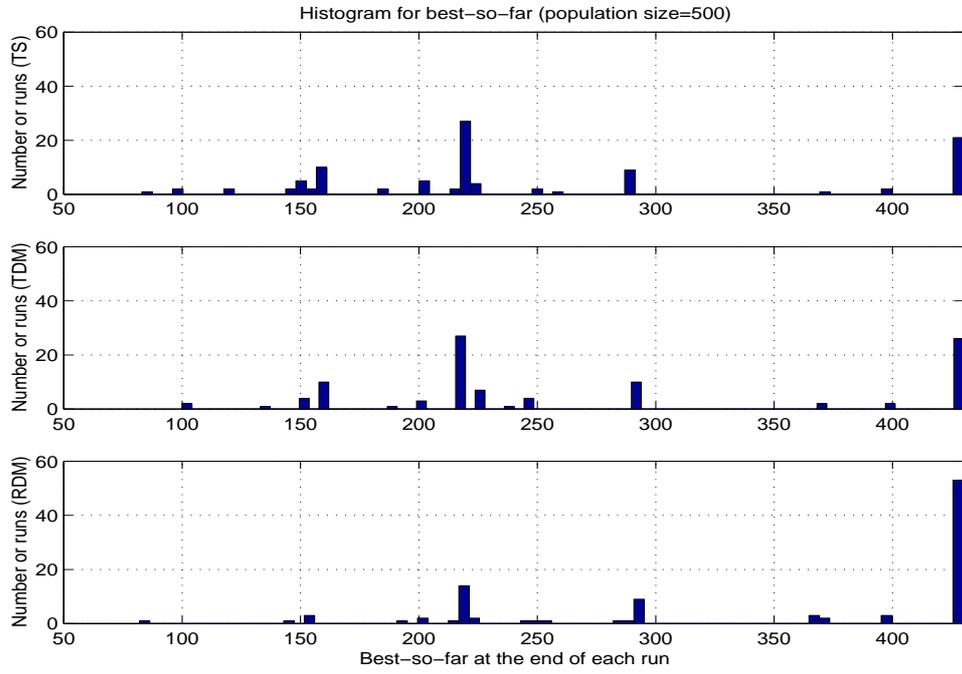


Figure 5.10: Histogram for best-so-far performance on H_3 .

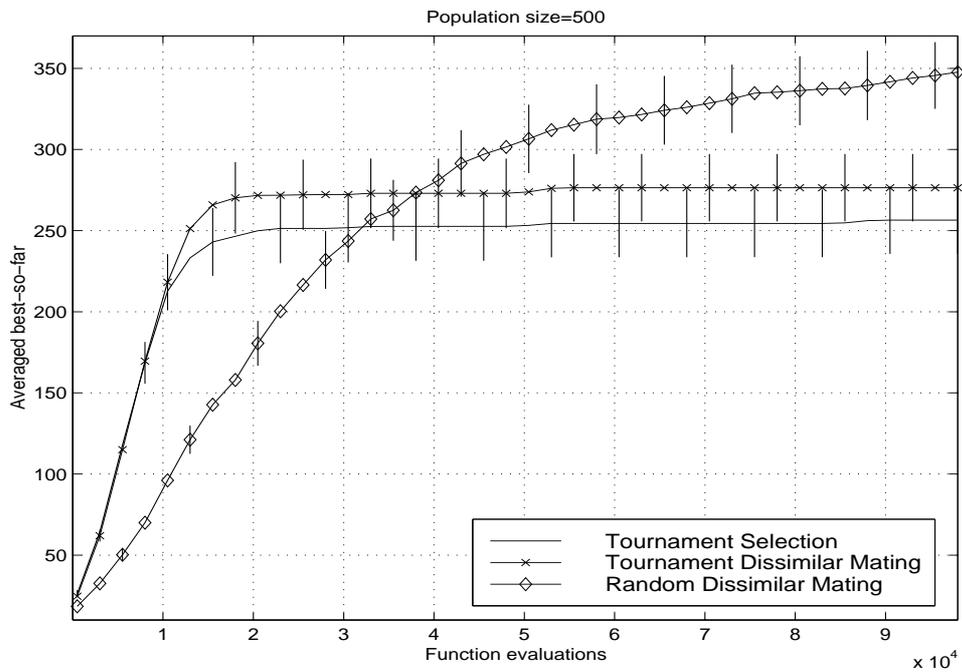


Figure 5.11: Best-so-far performance on H_3 .

tends to locate higher best-so-fars than the TS GA. Figure 5.11 presents the averaged best-so-far found at each generation. These results again show that the random dissimilar mating outperforms the other two mate selection schemes.

5.4 Summary and Discussions

In this chapter, I continued the study on the building-block-based Royal Road test functions. These functions serve as idealized testbeds for comparing effects of different mating preferences. I demonstrate how hitchhiking and the founder effect may be suppressed by means of dissimilarity-based mating strategies. I then further test the mate selection schemes on the hyperplane-defined functions. These functions extend the complexity of fitness landscapes, which are nonseparable, nonlinear, and nonsymmetric.

The results obtained show that mate selection is a crucial factor that affects GA's exploration of the search space. In short, I gained insights that indicate dissimilarity-based mate selections would get GAs away from hitchhiking and the founder effect, and facilitate the GA's search for further improvements (in terms of finding the best-so-far).

I have shown that the results obtained for small problems in Chapter 3 can be scaled to the original versions of the RR functions and the more complicated hyperplane-defined functions. For problems that are relatively easy to optimize, such as the two Royal Road R_1 and R_2 , the results show that the tournament dissimilar mating is a better strategy to facilitate the GA's search power. In other words, when an individual undergoes a mating event, it is beneficial to first use the selection pressure from the externally imposed fitness function to select candidate mates, and then mate with the most dissimilar among those candidates. On the other hand, on problems that present sufficient difficulty to the GA's search (e.g., the incompatible, deceptive Royal Road IR_1 and the hyperplane-defined functions H_1 , H_2 and H_3) the

results show that it is more beneficial that the dissimilar mate is selected from a number of candidates that are picked without the selection pressure imposed by an external fitness function. Simply stated, the random dissimilar mating GA outperforms the GA with the regular tournament selection or the tournament dissimilar mating scheme on such difficult problems.

I have briefly discussed possible reasons why all this occurs by comparing the nature of these mate selection strategies. The additional dissimilar mating preference of TDM generates an increased likelihood that mating partners come from distant parts of the search space, and the resulting matings could get offspring out of the local optima or reduce hitchhiking and the founder effect. This in turn facilitates the GA's searching different parts of the search space. The results indeed show that the TDM GA is able to outperform the TS GA on both the non-deceptive and deceptive test functions.

On the other hand, RDM differs from TDM by removing the selection pressure toward higher-fitness individuals while selecting mating partners. This can reduce the degree of convergence on certain local optima, and further the exploration of the search space. The results show that, on testbeds that present sufficient difficulty to the GA's search, the RDM GA indeed locates higher best-so-far values than the TDM GA, although its searching process is slower than that of the TDM GA.

Since the royal road and the hyperplane-defined test functions are explicitly constructed based on the ideas of building blocks, it is natural to examine if these results can be scaled to more realistic, non-building-block-based problems. My hypothesis is that, if the goal is to improve the best-so-far, then on easy problems TDM is more beneficial; but if problems present sufficient difficulty to the GA's search, then RDM is more desired.

In the next chapter, I will test this hypothesis by using several real benchmark testbeds.

CHAPTER 6

EMPIRICAL VALIDATION

6.1 Introduction

The results from the previous chapter on the building-block-based testbeds show that mate selection plays a crucial role in the GA's search power. Those results strongly suggest that, if one's goal is to search for best-so-far solutions (or the global optimum), then on non-deceptive test functions (e.g., the royal road R_1) the GA with TDM outperforms one with TS, and the GA with TS outperforms one with RDM. On the other hand, if a test function presents sufficient difficulty due to a high degree of deceptiveness (e.g., the hyperplane-defined function H_1), RDM is more beneficial than the other two schemes. For either case, the GA with TDM exhibits a higher (or equivalent) performance level than the GA with TS. In terms of the fitness landscape characteristics, the royal road R_1 is an example of unimodal testbeds, and the hyperplane-defined function H_1 belongs to the category of multimodal problems. I conclude that multimodality of the number of peaks in a search space is an important feature in determining the relative importance of dissimilarity-based mating preference. Therefore I predict that TDM would be more beneficial than TS on either unimodal or multimodal problems, and if a higher degree of multimodality presents sufficient difficulty to the GA's search, I expect that the GA using RDM would outperform that using the other two schemes.

Since the royal road and the hyperplane-defined test functions are explicitly con-

structured based on the ideas of building blocks, it is natural to examine if these results can scale to more realistic, non-building-block-based problems. To this end I chose several benchmark testbeds that are not explicitly constructed with building blocks (though building blocks may be embedded) to empirically test my hypothesis. Before doing so, it is worth discussing several related issues when empirical studies are employed.

In GA research, it is a common practice to run GAs to some fixed termination criteria, and then to report the results only after termination. For example, if a testbed's global optimum is known ahead of time, one can compare the mean function evaluations needed for different GAs to reach a given threshold (or the global optimum). One can also halt the search until a maximum of 10000 function evaluations, and then report the value of the best-so-far reached. However, these two methods ignore the dynamic aspects of a GA, and can lead to overly general conclusions. For example, in Figure 5.11, if one predetermines a threshold at value 150, it is natural to conclude that TDM and TS are more beneficial than RDM, since the averaged function evaluations taken by the TDM and TS GAs to reach the value of 150 are much smaller than those by the RDM GA. However, if the threshold is at 300, then it is clear that the RDM GA is the only one which reaches that value. This shows that conclusions can often turn out to be surprisingly dependent on the termination criteria, often reversing if a different cutoff is used. Therefore, inspecting the dynamic aspects of an GA would give us additional information for reliable judgements.

In addition, from both an engineering and scientific standpoint it is crucial to include results throughout the running of GAs. For instance, in cases where online performance is important, one needs to inspect the results over the whole running time of an GA and then judge if the GA is a good one in terms of the performance metric employed. Using which criterion depends on the difficulty of a problem, the goal of an investigation and the context of a search algorithm.

I have already used the three criteria above to compare the GAs' performance in the preceding chapters. In this chapter, I continue employing these criteria while conducting empirical examination of my hypothesis. If a test function is easy and its global optimum is known ahead of time, I compare different GAs' performance in terms of the mean function evaluations needed for finding the optimum. On simple test functions this performance measure suffices to illustrate the effects of mate selection. On the other hand, if the global optimum is unknown or difficult to reach, I halt the search after a maximum of certain function evaluations, and then report the number of runs in which an GA found the global optimum or the average of the best-so-fars over the whole running of an GA (as was done in the previous chapters). Since higher fitness indicates being closer to an optimum, the GA will have to maximize the function throughout this chapter. In some cases I will also use auxiliary techniques to gain deeper understanding for different mate selection schemes, such as examining the fitness distribution of a population's individuals to see how individuals are distributed over areas of the fitness landscape.

My main objective is to test the hypothesis described above as various aspects of the search space are included. Since the original versions of several testbeds employed in this chapter were aiming at minimization, proper modifications of these functions are required for comparing the GA's performance in terms of maximization. I adopt six test functions of increasing complexity and difficulty to deepen the understanding for the effects of mate selection. The first two testbeds I choose are modified versions of the De Jong test functions F_1 and F_3 (De Jong, 1975)—the modified F_1 is a sphere model and the modified F_3 is step function. These two testbeds are simple unimodal problems. The third testbed is the generalized Rosenbrock function (ICEO, 1997).* It is called Rosenbrock Saddle because a saddle shape is formed by a shallow “ravine” between two “ridges” merging at the global optimum of the fitness landscape. This

*De Jong's test function F_2 is a simple, two-dimensional version of this testbed.

function has many local optima on the ridges and shows some interesting phenomenon that is worth additional investigation. The fourth testbed is a problem generated from the idea of optimal control, in which a large portion of the search space is occupied by a hill, and two clusters of spikes are present at corners. The fifth test function is a modified version of the Schaffer function F_7 (Schaffer, Caruana, Eshelman, and Das, 1989). This testbed is a multimodal with patterns of deep wells and high barriers that were designed to present more or less difficulty to search by simulated annealing. Finally I employ Michalewicz's epistatic function (ICEO, 1997), which is a highly multimodal, nonlinear and nonseparable testbed.

Many of the characteristics in the testbeds discussed above are considered important (especially) by evolutionary algorithm practitioners, such as multimodality, deceptiveness, non-separability, etc. In the next section I start testing the hypothesis from the simplest case, i.e., the sphere unimodal.

6.2 Modified De Jong Function F_1

The first testbed used is a modified version of De Jong's test function F_1 :

$$f(\bar{x}) = 26.2144N - \sum_{i=1}^N x_i^2, \quad (6.1)$$

where $\bar{x} = [x_1, x_2, \dots, x_N]^T$, $-5.12 \leq x_i \leq 5.12$ for $1 \leq i \leq N$.[†]

A two-dimensional sketch of this function is presented in Figure 6.1. The X and Y-axis represent the index of sample points in parameters x_1 and x_2 that are used to compute $f(\bar{x})$, which is then represented on Z-axis. (This plotting method is applied to all other testbeds used in this chapter.) As can be seen, this function is a simple, unimodal problem. In this section, I use five variables (i.e., $N=5$) for experimental illustrations, each variable being encoded by 10 bits to cover the range (i.e., 10.24 in

[†]The original version of the De Jong function F_1 is $\sum_{i=1}^3 x_i^2$, and the goal was to minimize $f(\bar{x})$. This function is modified as Equation 6.1, with the constant 26.2144 obtained from the square of the boundary value, 5.12 (or -5.12), in order to be consistent with the goal of maximization.

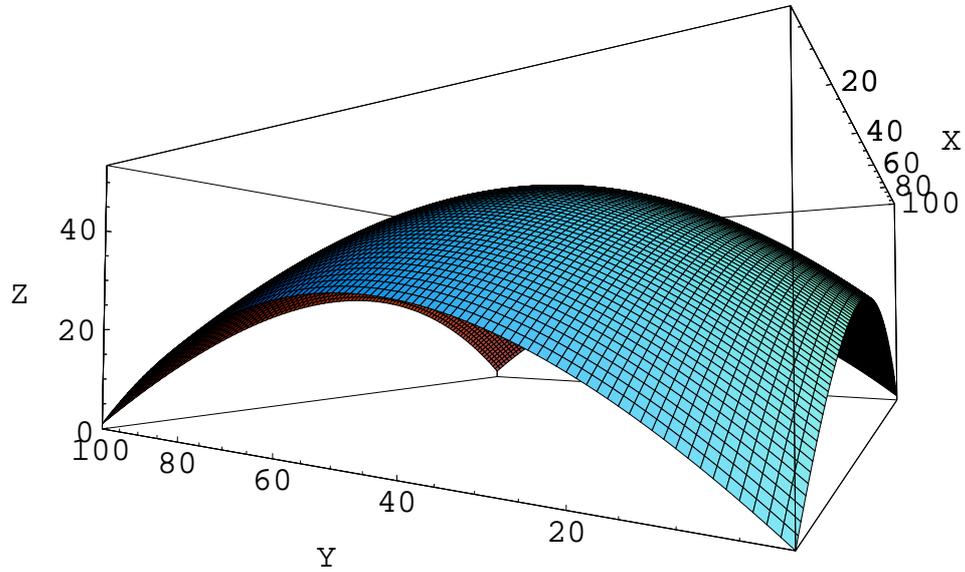


Figure 6.1: Modified, two-dimensional De Jong function F_1 .

this case) of the values each variable assumes. Each 10 bits is then concatenated to form a string representing an individual in the population; thus each individual is a binary string of length 50.

The first several experiments used a population size of 50, ran for 200 generations, with crossover and mutation rates of 1 and 0.005 respectively. I conducted 50 runs for each mating preference. Since this testbed's global optimum is easy to locate, I first compare the mean function evaluations needed for finding the optimum between GAs with TS, TDM and RDM if the GA found the global optimum in all 50 runs.[‡] If this is not the case, I record the number of runs in which the GA indeed found global optimum.

Since my hypothesis is that the TDM GA will outperform the TS GA, and the TS GA will outperform the RDM GA on a unimodal test problem, I expect that TDM is the best mating preference for the GA's search on this unimodal testbed. Table 6.1 shows the results obtained for these three mate selection schemes (the

[‡]The results presented here were based on the values reached close to the maximum to 3 digit accuracy. More digits generate higher precision, but my experiments showed that the qualitative difference remains the same.

Table 6.1: Mean function evaluations to the optimum (population size 50).

Mate Selection Scheme	Mean Function Evaluations to Optimum
Tournament Selection (TS)	2886 (68)
Tournament Dissimilar Mating (TDM)	2856 (64)
Random Dissimilar Mating (RDM)	2 runs found the global optimum

standard errors are shown in the parentheses). We can see that the GAs with TDM and TS outperform that with RDM. This partially confirms the hypothesis, but it is still not clear if the TDM GA could outperform the TS GA in this case.

Table 6.2: Mean function evaluations to the optimum (population size 20).

Mate Selection Scheme	Mean Function Evaluations to Optimum
Tournament Selection (TS)	1670 (18)
Tournament Dissimilar Mating (TDM)	1557 (15)
Random Dissimilar Mating (RDM)	69 runs found the global optimum

More experiments revealed that the reason for the similar performance level of the TDM and TS GAs is that population size 50 is already large enough on this simple testbed that the performance discrepancy between TDM and TS is suppressed. I then used a small population size, 20, and reran experiments for 500 runs, hoping to obtain more reliable results to further compare the performance difference between TDM and TS. The new results are shown in Table 6.2.

As can be seen, the standard errors in this case are much smaller than in the case of population size 50, which shows the results are sufficiently reliable and I can be sure that the TDM GA outperforms the TS GA, since the performance difference between the two schemes is an average of 120 function evaluations (six generations). This can be further confirmed in Figure 6.2, where the error bars (95% confidence intervals) are almost invisible, and we can see the clear difference between the averaged best-so-fars

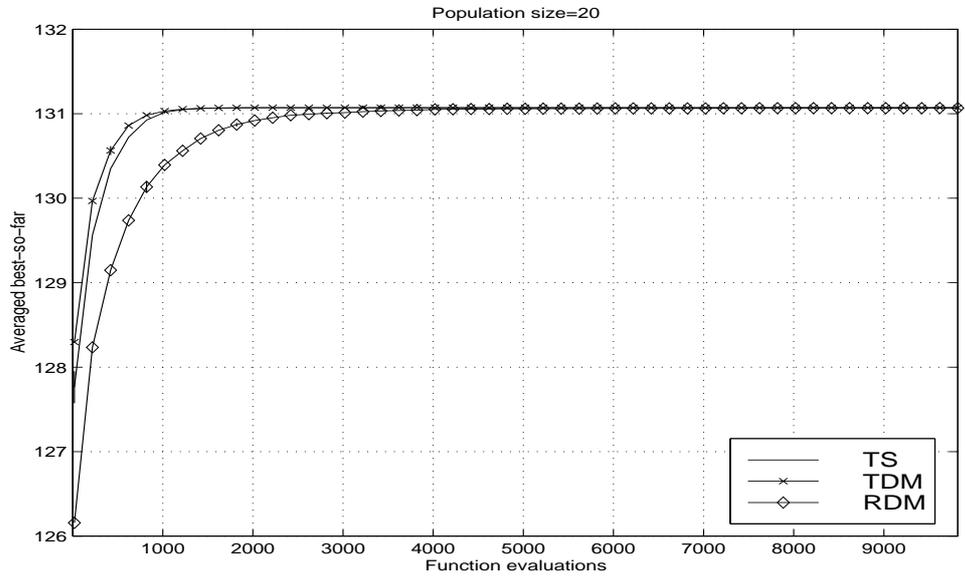


Figure 6.2: Averaged best-so-far performance on the modified De Jong function F_1 .

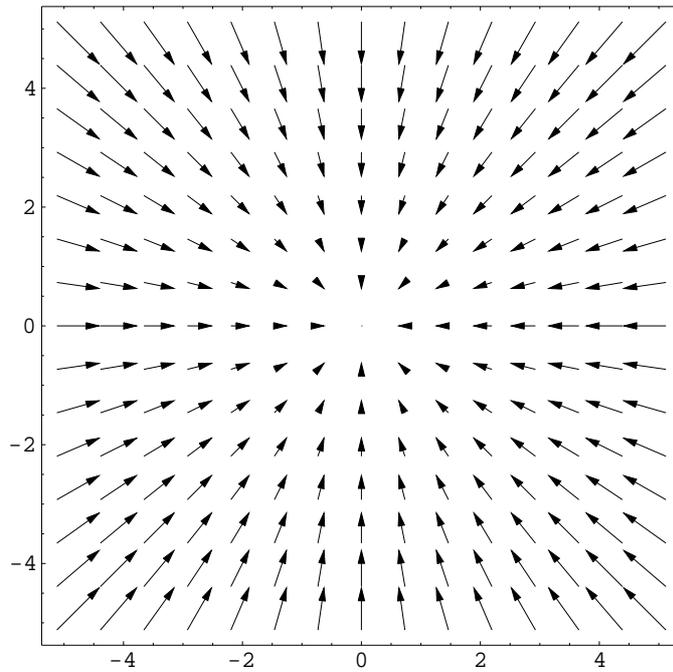


Figure 6.3: Vector field of the two-dimensional modified De Jong function F_1 .

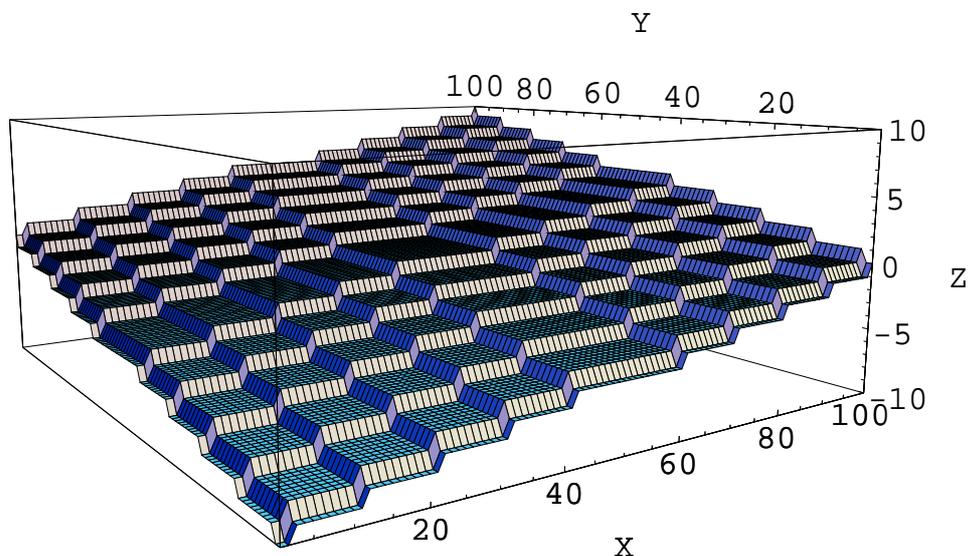


Figure 6.4: Two-dimensional De Jong function F_3 .

corresponding to TDM and TS in earlier generations. When the TS GA found the optimum, its best-so-far curve then overlapped that of the TDM GA.

Note, however, that eventually the performance curve corresponding to RDM picks up in later generations (though there is still a very tiny amount difference of .004, which is almost invisible in this plot), but there are only 69 runs out of 500 in which the GA found the optimum. A closer inspection shows that the population individuals of the RDM GA quickly converged to the area near the global optimum, and then wandered around for a relatively long period. What appears to be happening is that since this region is relatively flat,[§] and since RDM does not use additional selection pressure toward higher-fitness individuals, this in turn slows down the GA's search process.

6.3 De Jong Function F_3

[§]Figure 6.3 illustrates the local gradients of the two-dimensional version of this function. The length of an arrow represents the magnitude of the corresponding gradient. As can be seen, the magnitudes of the gradients around the global optimum are much smaller than those at outer regions, indicating this area is much flatter than other regions.

Table 6.3: Mean function evaluations to the optimum.

Mate Selection Scheme	Mean Function Evaluations to Optimum
Tournament Selection (TS)	2362 (202)
Tournament Dissimilar Mating (TDM)	1665 (109)
Random Dissimilar Mating (RDM)	3598 (145)

The second testbed employed is De Jong’s function F_3 :

$$f(\bar{x}) = \sum_{i=1}^N \text{integer}(x_i), \quad (6.2)$$

where $\bar{x} = [x_1, x_2, \dots, x_N]^T$, $-5.12 \leq x_i \leq 5.12$ for $1 \leq i \leq N$. A two-dimensional sketch of this function is illustrated in Figure 6.4. As can be seen, this function is a simple, unimodal step function; I expect that the global optimum is easy to locate. Thus comparing the mean function evaluations needed for locating the optimum is sufficient to determine the performance difference of different GAs.

In this section, I again use five variables (i.e., $N=5$) for illustrations, each variable being encoded by 10 bits. The five blocks of ten bits each are then concatenated to form a string of length 50. For each mating preference, I conducted experiments of 50 runs, and ran for 200 generations, based on population size 50, with crossover and mutation rates of 1 and 0.005 respectively.

Our objective is again to examine if my hypothesis holds—i.e., the TDM GA could outperform the TS GA, and the TS GA could outperform the RDM GA on such a unimodal test problem. The results are shown in Table 6.3 (the standard errors are shown in the parentheses), which indeed confirms the hypothesis.

6.4 Generalized Rosenbrock Function

The third testbed is the generalized Rosenbrock function:[¶]

$$f(\bar{x}) = 1 - \frac{1}{98221.9(N-1)} \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2], \quad (6.3)$$

where $\bar{x} = [x_1, x_2, \dots, x_N]^T$, $-5.12 \leq x_i \leq 5.12$ for $1 \leq i \leq N$.^{||} A two-dimensional sketch of this function is presented in Figure 6.5.

As one looks at Figure 6.5, he might expect that this function is easy to optimize because it looks like a unimodal function. In fact, the seeming smooth surface of this function is misleading. Figure 6.6 illustrates the local gradients of this function. As can be seen, two ridges that start from the vicinity of $(3, 5.12)$ and $(-3, 5.12)$ gradually merge at the global optimum at $(0,0)$. Moreover, the gradients around these two ridges point away from the global optimum and towards the nearest ridge, except those that are close to the global optimum. This indicates that points at these areas would be attracted to the ridges, instead of the global optimum. Therefore, there are, in fact, many local optima embedded in this function. This in turn presents more difficulty to search by the GA. Since this function is no longer unimodal, my hypothesis is that TDM and RDM are more beneficial than TS in facilitating the GA's search for the global optimum on this testbed.

In this section, I again use five variables for illustration, and each variable is encoded by 10 bits so that the string length is 50. Other parameter values remain the same as those used in the previous section, except that a higher precision of five digit accuracy is used to examine if the global optimum is found. This is because the fitness values at the regions around the global optimum are rather close to it and

[¶]This function was used as a testbed in the Second International Contest on Evolutionary Optimization held in the IEEE-ICEC 97 conference (ICEO, 1997). De Jong's function F_2 is a two-dimensional Rosenbrock function

^{||}The original version of the generalized Rosenbrock function is $f(\bar{x}) = \sum_{i=1}^{N-1} [100(x_i^2 - x_{i+1})^2 + (1 - x_i)^2]$, and the goal was to minimize $f(\bar{x})$. This function is normalized (by dividing $98221.9 * (N - 1)$), where 98221.9 is obtained based on $(x_i, x_{i+1}) = (-5.12, -5, 12)$ and modified as Equation 6.3 for maximization.

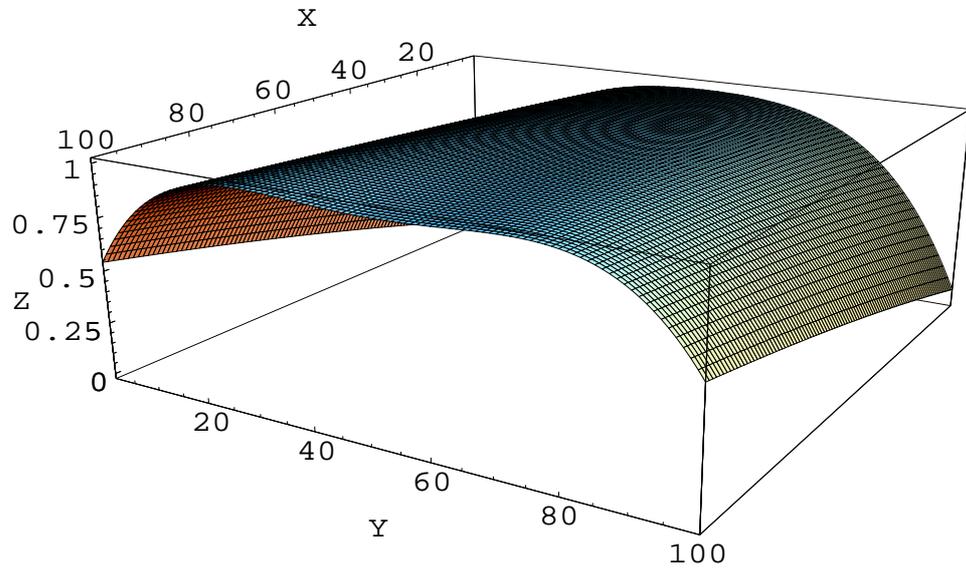


Figure 6.5: Modified, two-dimensional Rosenbrock Saddle.

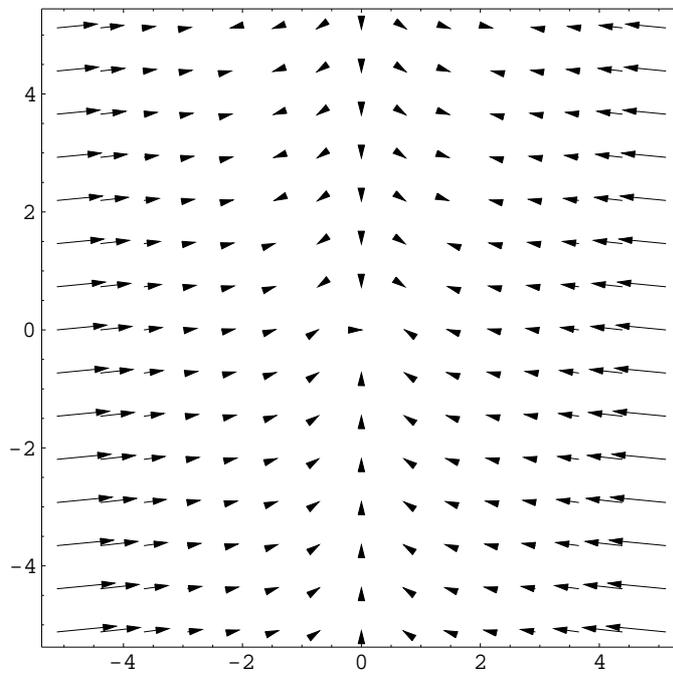


Figure 6.6: Vector field of the two-dimensional Rosenbrock Saddle.

Table 6.4: Number of runs out of 50 in which the optimum is found.

Mate Selection Scheme	Number of runs out of 50
Tournament Selection (TS)	18 runs found the global optimum
Tournament Dissimilar Mating (TDM)	29 runs found the global optimum
Random Dissimilar Mating (RDM)	29 runs found the global optimum

thus a higher precision is required for further comparison. I report the results for the number of runs in which the GA indeed found global optimum in Table 6.4.

The results show that both the TDM and RDM GAs outperform the TS GA, but at this point it is not clear which of TDM and RDM performs better.

Figure 6.7 illustrates the averaged best-so-far performance, where we can more clearly see that the RDM GA outperforms the TDM GA.

To understand why all this occurs, consider the nature of these mate selection strategies. Recall that the difference between TDM and TS is that TDM always chooses dissimilar mates. This generates an increased likelihood that mating partners come from distant parts of the search space, and the resulting matings could get offspring out of the local optima. This in turn facilitates the GA's searching different parts of the search space, and increases the likelihood of exploring other optima of higher fitnesses. On the other hand, RDM differs from TDM by removing the selection pressure toward higher-fitness individuals while selecting mating partners. This can reduce the degree of convergence on certain local optima, and further the exploration of the search space. Simply stated, both TDM and RDM are capable of facilitating the GA's exploration of the search space; the only difference is the strategies they employed.

A closer look at the best-so-far distribution can deepen our understanding of the difference between these three schemes. Figure 6.8 illustrates the histogram that shows the best-so-far distribution when the number of maximum function evaluation (20,000) is reached. The trend in these results shows that the RDM GA tends to

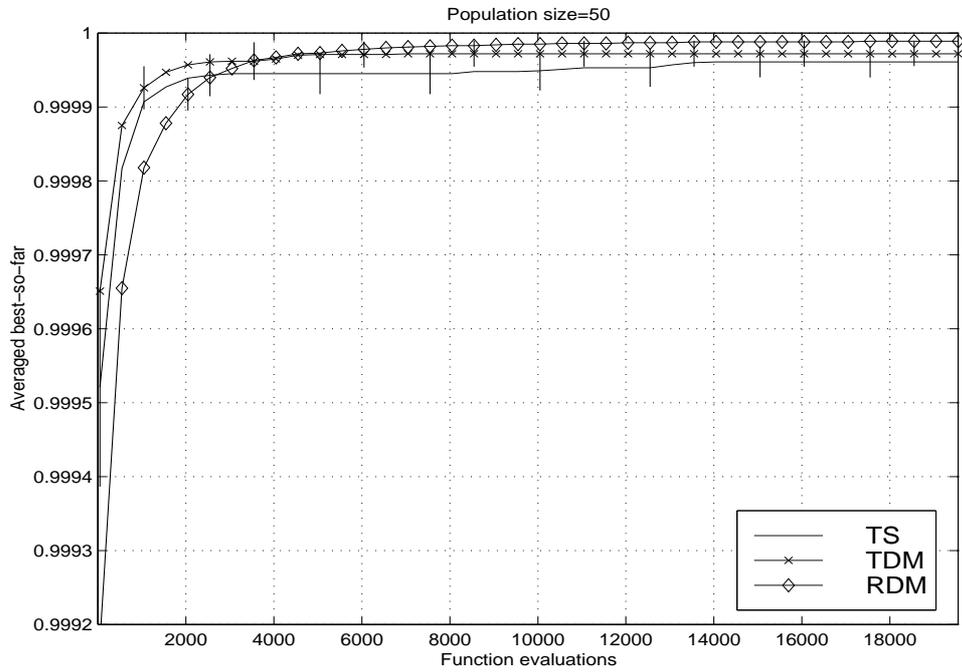


Figure 6.7: Averaged best-so-far performance on the generalized Rosenbrock function.

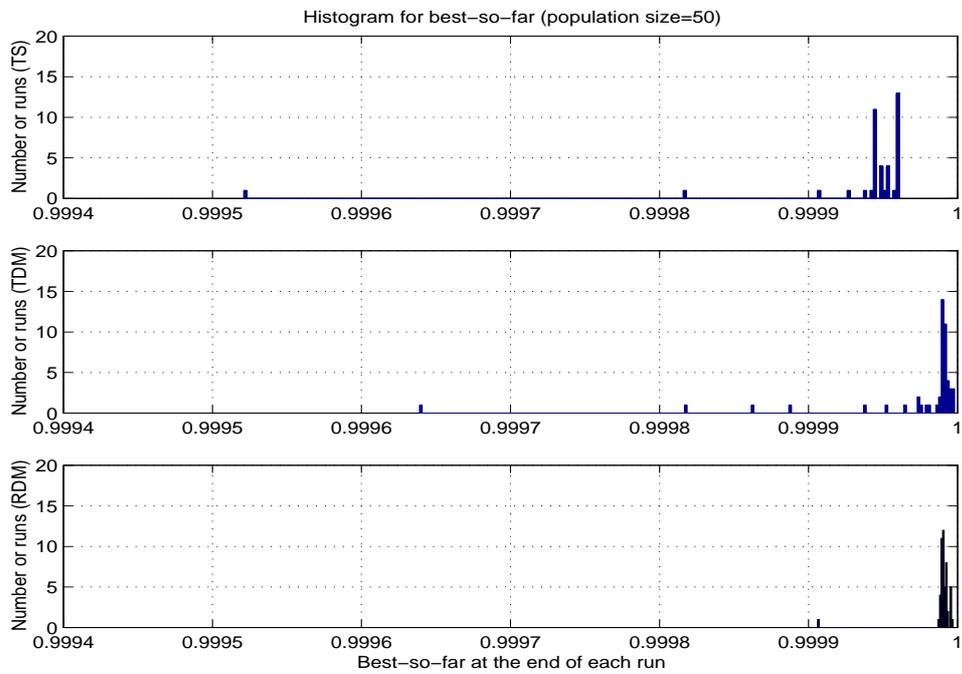


Figure 6.8: Histogram for the best-so-far values on the generalized Rosenbrock function.

locate higher best-so-fars than the TDM GA, and likewise the TDM GA tends to locate higher best-so-fars than the TS GA. This indicates that the best-so-fars found by the RDM GA are closer to the global optimum than the TDM GA; and the best-so-fars found by the TDM GA are also closer to the global optimum than the TS GA. I conclude that the strategy used by RDM for exploring the search space is more beneficial than that used by TDM on this testbed. Therefore, it is reasonable to hypothesize that if a problem presents sufficient difficulty to the GA's search, such as the large number of local optima present on the generalized Rosenbrock Saddle, RDM would be a more desired strategy for improving the search of best-so-far solutions. In the following sections, I will present more results to further support this idea.

6.5 An Optimal Control Test Problem

Optimal Control problems often arise in many different fields of engineering and sciences. This class of problems has been well studied from both theoretical and computational perspectives. The models used to describe optimal control problems almost always involve more or less nonlinearity in nature. This often results in the existence of multiple local optima in the area of interest. (See Hager and Pardalos (1998) for a sample of the available material and applications.)

In this section I designed a simple optimal control problem with one state, z , and two control variables, u_1 and u_2 , for further testing of my hypothesis. The following is the description of this test problem.

Formulation

Objective function

$$\max_{u_1, u_2} z(t_f)^2.$$

Constraints

$$\frac{d^2 z(t)}{dt^2} + \sin(z(t)) \frac{dz(t)}{dt} + \sin(t) \cos(z(t)) z(t)^3 = \sin(t) u_1^2 + \cos(t) u_2^2 + \sin(t) u_1 u_2.$$

Initial conditions

$$z(t_0) = 2, \dot{z}(t_0) = 2.$$

Control

u_1 and u_2 are constants over the whole time horizon.

Control variable bounds

$$-5 \leq u_i \leq 5, i=1, 2.$$

$$t \in [0,1].$$

A sketch of this function is illustrated in Figure 6.9. The X and Y-axis represent the index of sample points in parameters u_1 and u_2 that are used to compute $z(t_f)^2$, which is then represented on Z-axis. There are clusters of spikes at two corners of the search space, and a hill that occupies most of the space. The magnified view in Figure 6.10 shows a clearer picture of the height and area of the hill.

Since on the generalized Rosenbrock function the height differences (i.e., the fitness differences) between many local optima and the global optimum are rather small, I design this optimal control problem to enlarge the height differences for further comparing the three mate selection schemes. As can be seen, the height of the hill is much lower than that of the spikes, but since it occupies most of the search space, I expect that most of the population individuals would be attracted to the hilltop. To improve the GA's search power, as hypothesized in the last section, the GA can either employ TDM to choose dissimilar mates that may come from distant parts of the search space (preferably from the spikes) to get away from the hill, or employ RDM to decrease the selection pressure in order to reduce the degree of convergence on local optima.

In this section, each of the two variables is encoded by 30 bits, and thus each individual is a binary string of length 60. (All other parameter values are the same as those used in the previous sections.)

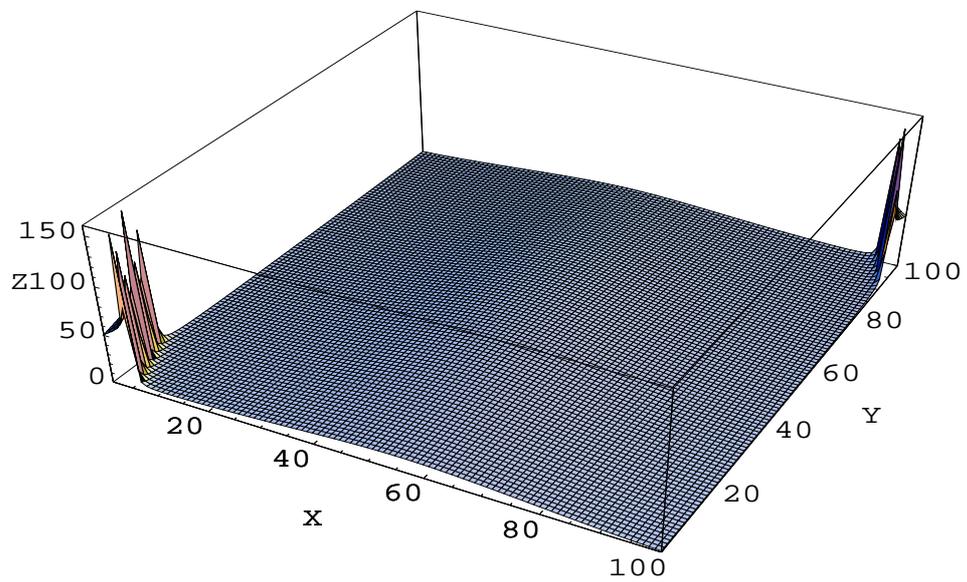


Figure 6.9: Optimal control test problem.

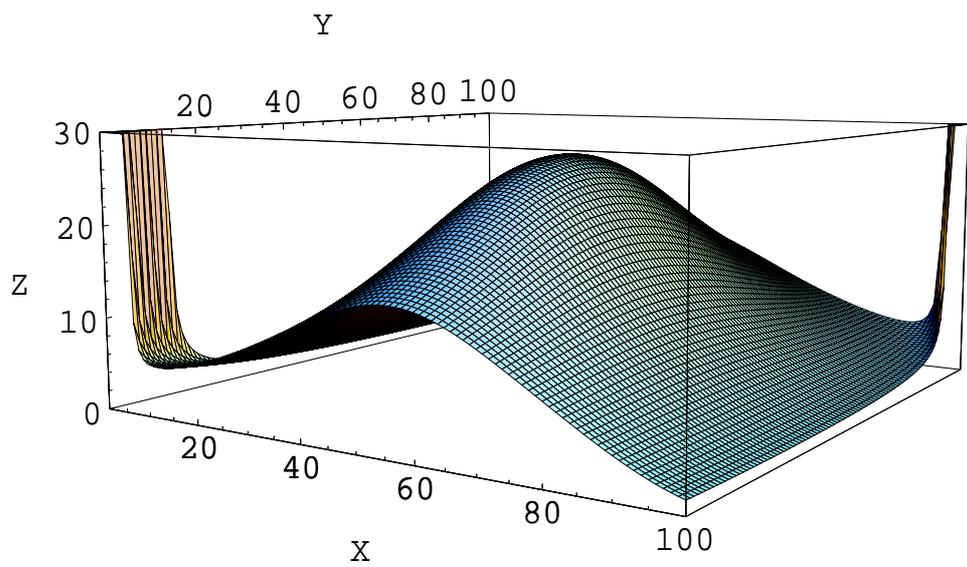


Figure 6.10: Magnified schematic of the optimal control test problem.

Figure 6.11 shows the averaged best-so-far performance. We can see that RDM and TDM considerably outperform TS, which validates my hypothesis. We can also compare the three mate selection schemes by counting the frequency of best-so-fars attained at the end of each run. Figure 6.12 illustrates the histogram that shows the best-so-far distribution until the maximum function evaluations. We see that the TS GA's population converges on the hilltop (fitness value of 27.01) in more than 30 runs (out of 50), and the TDM GA can explore more of the search space to extend the best-so-fars to higher values. The most noticeable in the case of RDM is that the number of runs in which the GA's population get trapped on the hill is decreased, thus generating higher averaged best-so-far values than the other two schemes.

This confirms my hypothesis in the last section that if a problem presents sufficient difficulty to the GA's search, RDM would be a more desired strategy for improving the search of best-so-far solutions.

In the next two sections, I use additional two test functions that are also highly nonlinear, multimodal, and difficult to optimize to further test the hypothesis.

6.6 Modified Schaffer Function F_7

The fifth test function is a modified version of Schaffer's test function F_7 (Schaffer, Caruana, Eshelman, and Das, 1989). This testbed is a multimodal with patterns of deep wells and high barriers that were designed to present more or less difficulty to search by hill climbing.

The test function is:

$$f(\bar{x}) = 1.5 - (x_1^2 + x_2^2)^{0.25}[\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1],$$

where $-1 \leq x_i \leq 1$ for $1 \leq i \leq 2$.** A sketch of this function is displayed in Figure 6.13. To attain the global optimum at the center of the search space, the GA

**The original Schaffer's test function F_7 is $f(\bar{x}) = (x_1^2 + x_2^2)^{0.25}[\sin^2(50(x_1^2 + x_2^2)^{0.1}) + 1]$, and the goal was to minimize $f(\bar{x})$. I modify this function by adding the negative sign for maximization.

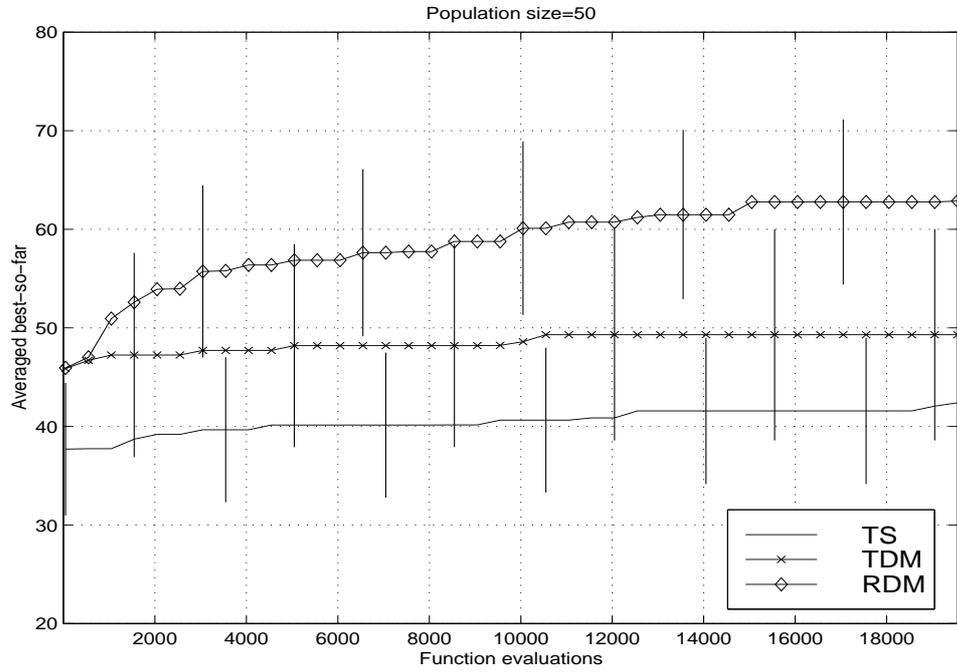


Figure 6.11: Averaged best-so-far performance on the optimal control test problem.

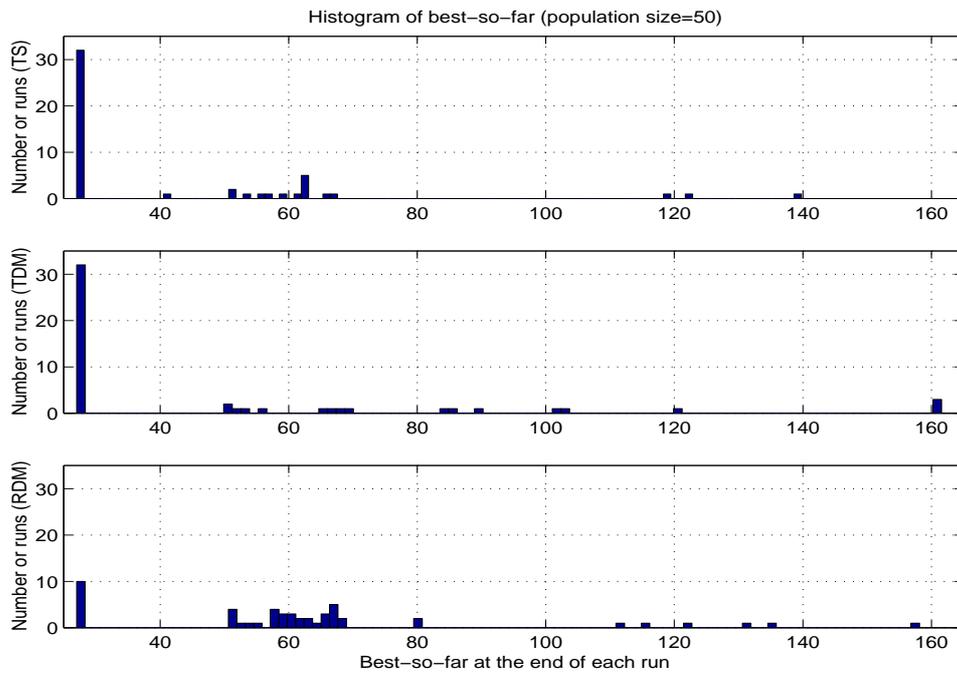


Figure 6.12: Histogram for the best-so-far values on the optimal control test problem.

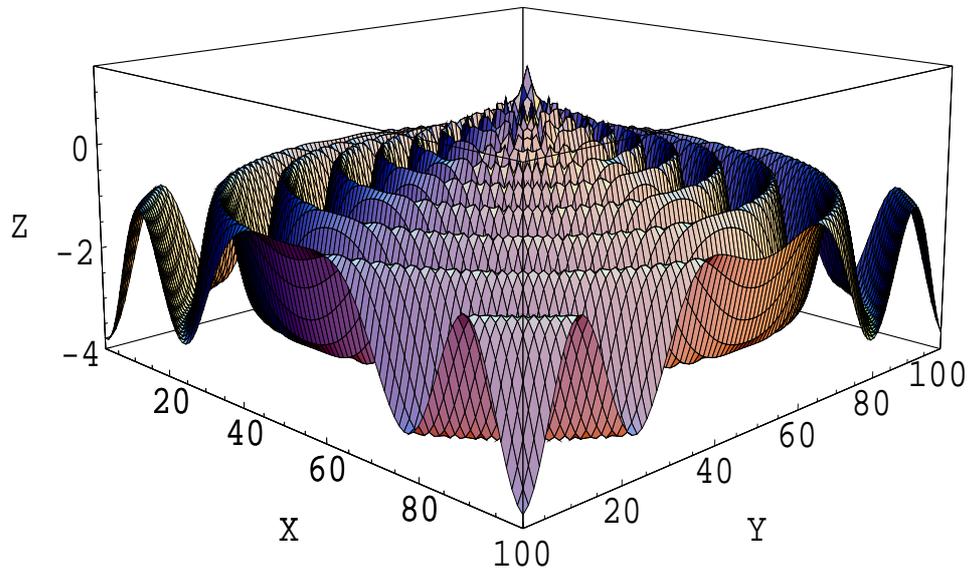


Figure 6.13: Modified Schaffer function F_7 .

would have to cross over many deep wells and high barriers. Since there are many local optima in the search space, a simple GA can easily converge on any of them. I expect that this problem presents sufficient difficulty to the GA's search.

In this section, each of the two variables is also encoded by 30 bits, and each individual is a binary string of length 60. All other parameter values are the same as those used in the previous sections. Figure 6.14 shows the averaged best-so-far performance, where we can see that the RDM GA substantially outperforms the TDM GA, and the TDM GA also outperforms the TS GA. This again confirms my hypothesis. Figure 6.15 further illustrates the histogram that shows the best-so-far distribution until the number of maximum function evaluation is reached. It is clear that there are more runs in which the RDM GA attains values that are close to the global optimum (i.e., 1.5) than the TDM GA. Likewise, there are more runs in which the TDM GA attains values that are close to the global optimum than the TS GA. All this shows that the sufficient difficulty presented by the modified Schaffer function F_7 results in the performance advantage of RDM.

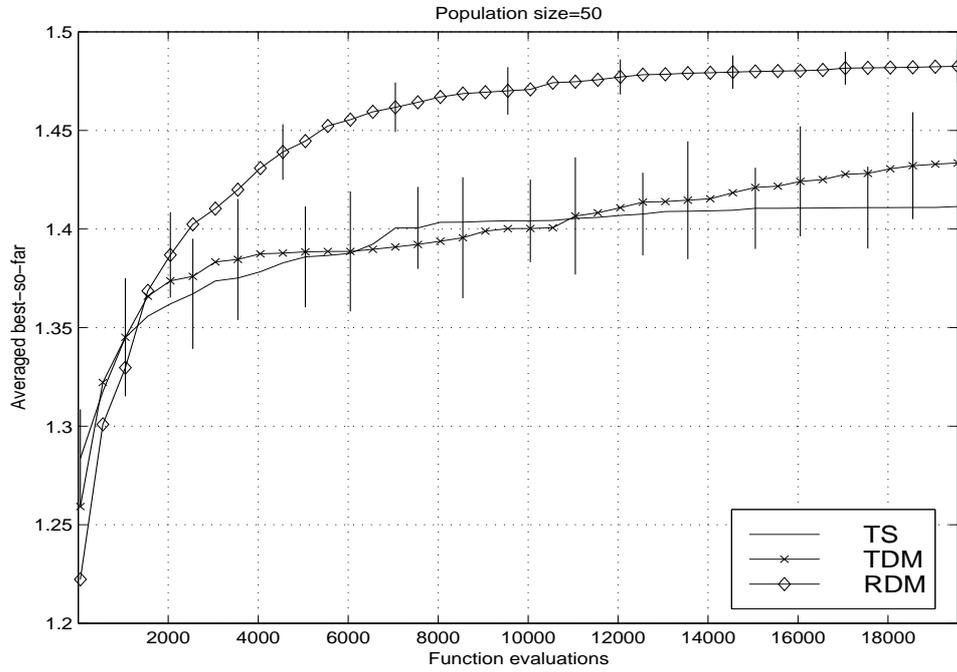


Figure 6.14: Averaged best-so-far performance on the modified Schaffer function F_7 .

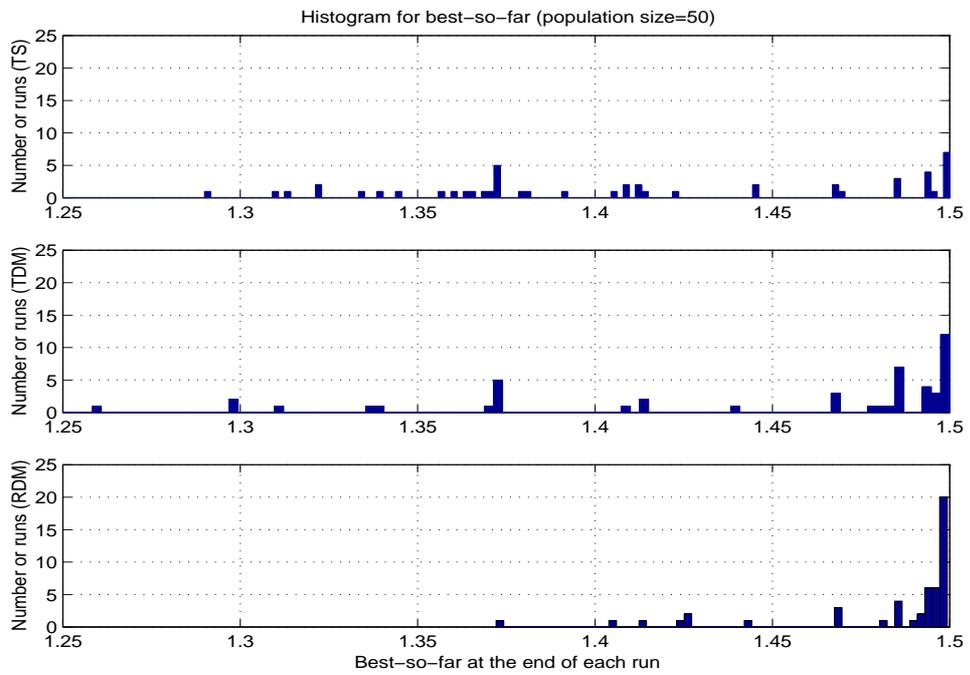


Figure 6.15: Histogram for the best-so-far values on the modified Schaffer function F_7 .

6.7 Epistatic Michalewicz Function

The final testbed employed in this chapter is Michalewicz's epistatic function (ICEO, 1997):

$$f(\bar{x}) = \sum_{i=1}^N \sin(y_i) \sin^{2m}\left(\frac{iy_i^2}{\pi}\right),$$

where

$$\begin{aligned} y_i &= x_i \cos \frac{\pi}{6} - x_{i+1} \sin \frac{\pi}{6}, \quad \text{if } i \bmod 2 = 1; \\ y_i &= x_{i-1} \sin \frac{\pi}{6} + x_i \cos \frac{\pi}{6}, \quad \text{if } i \bmod 2 = 0 \text{ and } i \neq N; \\ y_N &= x_N, \end{aligned}$$

and $m = 10$, $0 \leq x_i \leq \pi$ for $1 \leq i \leq N$.^{††} A system is lowly (highly) epistatic if the optimal allele for any locus depends on a small (large) number of alleles at other loci. The concept of epistasis in nature corresponds to nonlinearity in the context of GA (Goldberg, 1989). A sketch of a two-dimensional version of this function is shown in Figure 6.16. This function is a highly multimodal, nonlinear and nonseparable testbed. Due to the complicated, nonlinear dependence among alleles, I expect that this problem also presents considerable difficulty to the GA's search.

In this section, I again use five variables, each variable being encoded by 10 bits. Each individual is a binary string of length 50. All other parameter values remain unchanged.

Figure 6.17 shows the averaged best-so-far performance, where we can see that the RDM GA still outperforms the TDM GA, and the TDM GA still outperforms

^{††}The original Michalewicz's test function is $f(\bar{x}) = -\sum_{i=1}^N \sin(y_i) \sin^{2m}\left(\frac{iy_i^2}{\pi}\right)$, which was used as a testbed in the Second International Contest on Evolutionary Optimization held in the IEEE-ICEC 97 conference, and the goal was to minimize $f(\bar{x})$. Again, I remove the negative sign to maximize the test function.

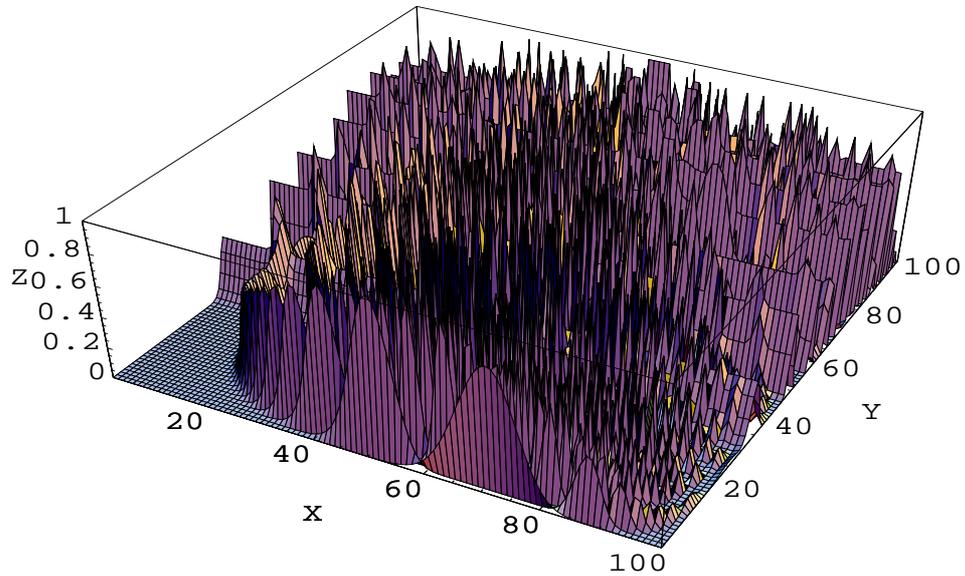


Figure 6.16: Modified, two-dimensional Michalewicz function.

the TS GA. This again confirms my hypothesis that RDM is most likely to improve the GA's search performance if a test function presents substantial difficulty to the GA's search.

Figure 6.18 illustrates the histogram that displays the best-so-far distribution. We see that there is a trend that RDM facilitates searching for higher best-so-fars than TDM and TS. This again agrees with my expectation.

6.8 Summary and Discussions

The results obtained in this chapter on the non-building-block-based test functions confirm the hypothesis that mate selection plays a significant role in the GA's search power. These results show that, if one's goal is to search for best-so-far solutions (or the global optimum), then on unimodal test functions TDM is the most desired mate selection strategy in improving the GA's performance. On the other hand, if a multimodal test function presents sufficient difficulty, the RDM GA could substantially outperform the TDM and TS GAs. For either case, the TDM GA exhibits a higher (or equivalent) performance level than the TS GA. Therefore, multimodality

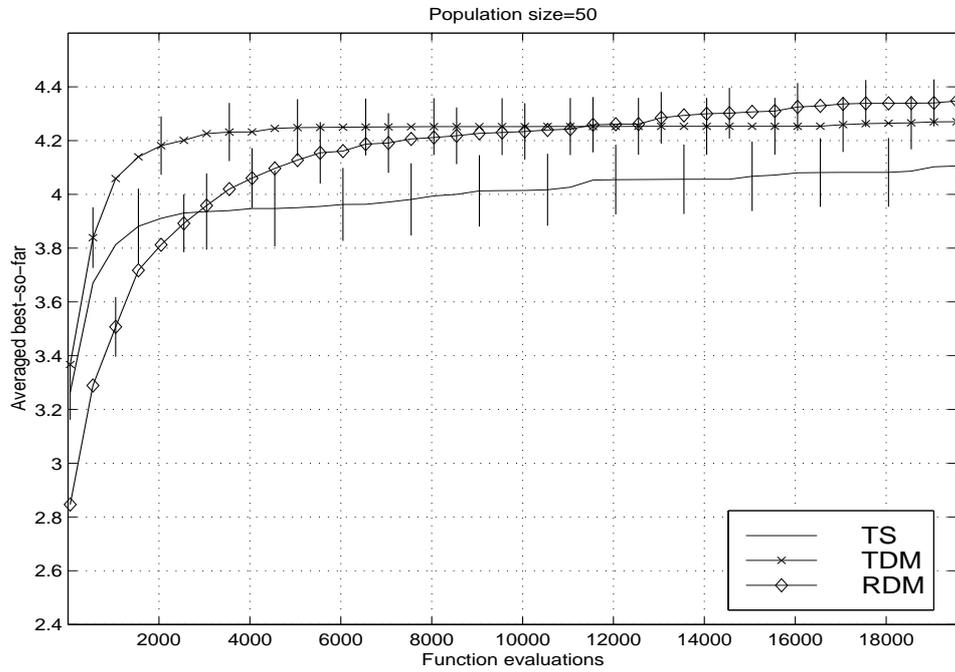


Figure 6.17: Averaged best-so-far performance on the epistatic Michalewicz function.

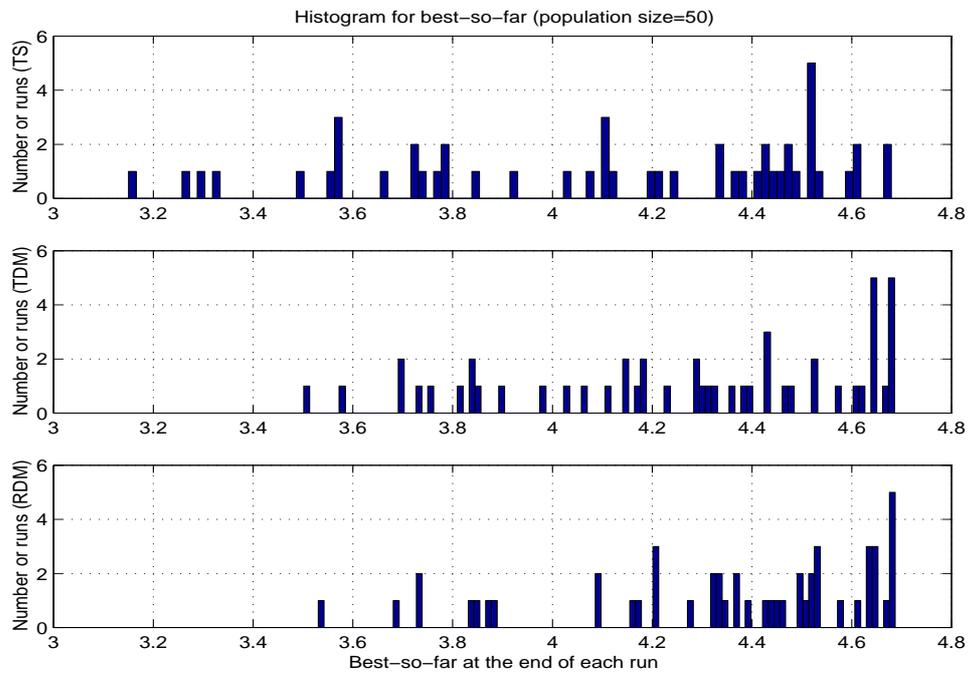


Figure 6.18: Histogram for for the best-so-far values on the epistatic Michalewicz function.

of a search space determines the relative importance of dissimilarity-based mating preferences.

My further understanding of the mate selection schemes used in this chapter is twofold: (1) by choosing dissimilar mates, TDM increases the likelihood of mating with individuals from distant parts of the search space to generate offspring that can get out of the local optima and further exploration of the search space; and (2) by further removing the selection pressure toward higher-fitness individuals while selecting mates, RDM can reduce the degree of premature convergence that occurs when TDM is employed. The benefit of using RDM is manifested in testbeds that presents sufficient difficulty to the GA's search.

The results obtained based on the real test functions are encouraging since it means that the ideas of mate selection proposed in this thesis can be applied to practical problems. In future work, I hope to develop an analytical analysis to enhance the understanding of the effects of mate selection. In particular, I will emphasize to what degree of a problem's difficulty, in terms of the fitness landscape characteristics, TDM or RDM is more beneficial than the other two schemes. This is important in the setting of function optimization, because we would need a guideline to determine what mate selection strategy to use to more effectively solve the problems at hand.

CHAPTER 7

IMMUNE SYSTEMS

7.1 Introduction

Our immune system is a complex of cells, molecules and organs which is able to perform several tasks, such as pattern recognition, learning, memory acquisition, generation of diversity, generalization and optimization. It protects us from an extraordinarily large variety of viruses, bacteria and other pathogenic organisms. Based on the immunological metaphor, new computational techniques have been developed for a better understanding of the system, as well as applications to practical problems. Among various immune system models, Smith, Forrest, and Perelson (1993) demonstrated their model, combined with ideas from classifier systems, can maintain diverse subpopulations of antibodies that recognize antigens. This is important in understanding the pattern-recognition capability of the immune system in that an almost limitless number of foreign cells and molecules (antigens) have to be detected and distinguished from self molecules. The key to this capability is the enormous diversity of the molecules employed by the immune system.

In the setting of multimodal function optimization, engineering and machine learning, there are two important issues when the GA is employed: (1) how fast can the GA discover one or several peaks (or best-so-fars)? And (2) can the GA maintain diverse subpopulations in different parts of the search space?*

*The first issue was briefly discussed in Section 3.4.3. For the second issue, in addition to

the dissimilarity-based mate selection facilitates locating a single, best-so-far solution at the expense of generating lethal offspring; and the similarity-based mate selection may enhance selection pressure toward highly-fit individuals such that the GA's population converges rapidly to a certain region of a fitness landscape (if mutation is turned off). Therefore, for the first question, I would expect the dissimilarity-based mate selection to improve the GA's performance based on that metric. On the other hand, our empirical results showed that simple GAs with the mate selection schemes are all subject to convergence (i.e., the GAs cannot maintain subpopulations). Thus for the second question, I intend to employ Smith et al.'s immune system model which is capable of maintaining population diversity, in order to offer additional insights into how the mate selection schemes compare to traditional selection schemes. In particular, I am interested in studying how different mate choices affect the capability of Smith et al.'s approach for maintaining subpopulations. Since I have shown that the dissimilar mating mechanisms are harmful in the sense of producing more useless hybrids I expect that such mating preferences will reduce the proportions of individuals in subpopulations. If so, the next question would be to study if reducing the probability of dissimilar mating (or increasing the probability of similar mating) can improve the capability for maintaining subpopulations.

In the next section, I illustrate three applications of immunological principles in order to gain ideas for future research in this field. Then I review Goldberg and Richardson's fitness sharing mechanism (1987) that serves as an idealized approach for maintaining population diversity, and present Smith et al.'s immune system model to discuss how it implements a form of implicit fitness sharing so as to facilitate formation of subpopulations. Section 5 presents experimental results that answer the two questions mentioned above. Finally, this chapter is concluded with the insights

the discussions for the immune systems in Section 7.4, there are some practical problems where maintaining subpopulations are critical. For example, see (Vlachos, Williams and Gomm, 1999) for an application of genetic approach to decentralized PI controller tuning for multivariable processes.

obtained for the mate selection schemes and future research lines.

7.2 Immune System Applications

Models of immunity-based systems are finding increasing applications in the fields of science and engineering. As of today, several computational algorithms have been developed and applied to different problems in order to demonstrate how principles gleaned from the immune system can be used for the design of tools for solving complex tasks. A few representative applications include robotics, control, optimization, neural network approaches, anomaly detection, agent-based modeling, machine learning, pattern recognition, to name a few. (Many applications are now introduced as an emerging area of research, called *immune engineering*. For a nice review of immune system applications, see (De Castro and Von Zuben, 2000)).

In this section I briefly discuss three applications of the immune system principles for possible future research lines. The three applications are relevant to engineering and computer science—fault diagnosis, control, and computer security.

7.2.1 Fault Diagnosis

Ishida (1993) adopted the insight from the mutual recognition feature of the immune network model (Jerne, 1974) to study fault diagnosis. In his approach, fault tolerancy was attained by mutual recognition of interconnected units in the studied plant. The system level recognition was achieved by unit level recognition.

The advantage of his model is that it can do parallel processing and handle incomplete information and data. In addition, his approach shows the phenomenon of self-organization, and no feedback loop is necessary in the failure propagation. Ishida and Mizessyn (1992) presented an application based on this mutual recognition model to the process instrumentation system of a chemical plant. Using the relationship among sensors, sensor networks are constructed by bi-directional arcs, in

which the model is applied to fault diagnosis. The results reported are very promising and worth further investigation.

7.2.2 Control

In modern control engineering a lot of effort is put into designing robust control algorithms that can meet the steadily increasing demands for high performance. The mechanisms of immune-based systems, like clonal selection and affinity maturation, along with its network dynamics, suggest successful applications of these metaphors to the control theory.

Krishnakumar et al. (1995) and Krishnakumar and Neidhoefer (1997, 1999) defined the Immunized Computational Systems (ICS) that used the immune system metaphor and computational techniques to introduce robustness and adaptability of biological immune systems to tackle control problems. The system consists of a hybrid structure whose building blocks, mimicking the immune building blocks, are composed of artificial neural networks, fuzzy systems and evolutionary algorithms. A collection of building blocks are combined together in order to generate a population of computational systems (representing the antigen and antibody equivalents), which are subject to evolution through selection and genetic operators. In addition, their immune metaphor focused on the clonal selection principle, as well as the affinity maturation of the immune response. The strategy was tested on an autonomous aircraft control problem and the performance is satisfying.

The immune system metaphor has also been applied to sequential control—finding a suitable execution sequence for a set of actuators so that the system achieves a desired state from a known initial state. Ootsuki and Sekiguchi (1999) proposed a method for determining control sequences of a sequential control plant based on the immune system. They used the Petri Nets formalism and the immune network theory to develop their model. The determination of control sequences is equivalent to the

determination of firing sequences for the Petri net models, which are decomposed into several subproblems. The objective function of each subproblem is set to satisfy constraints and minimize the objective function of a distributed (immune) network. Their method was compared to a conventional one, and its performance was superior for larger problems with distributed characteristics and high degree of freedom in the components.

7.2.3 Computer Security

Stephanie Forrest et al. at the University of New Mexico are working on a research project with a long-term goal to build an artificial immune system for computers. The security of computer systems depends on many activities, such as maintaining the integrity of data files, detecting unauthorized use of computer facilities, and preventing the spread of computer viruses. The problem of protecting computer systems from harmful viruses is viewed as an instance of the more general problem of distinguishing self (e.g., legitimate users, uncorrupted data) from dangerous other (e.g., unauthorized users, viruses, and other malicious agents). This approach is intended to be complementary to the more traditional cryptographic and deterministic approaches to computer security.

The virus-detection application of Forrest et al. (1994) employed a negative-selection algorithm to detect changes in the protected data and program files.[†] A number of experiments are performed in a DOS environment with different viruses, including file-infector and boot sector virus samples. They reported that the method could easily detect the modification that occurred in the data files because of virus infection. Compared to other virus detection methods, this algorithm has several

[†]The negativeselection algorithm developed by Forrest et al. works on principles that are similar to those of self/nonself discrimination in the immune system, in which the discrimination is achieved in part by Tcells, which have receptors on their surface that can detect antigens. That is, the negative-selection algorithm generates detectors randomly, and eliminates the ones that detect self, so that the remaining Tcells can detect any nonself.

advantages over the existing change detection methods: it is probabilistic and tunable in the sense that the probability of detection can be traded off against CPU time. It can also be distributed to provide high system-wide reliability at low individual cost, and detect novel viruses that have not previously been identified.

7.3 Fitness Sharing

Fitness sharing was an idea motivated by Holland's discussion (1975) in which the number of individuals occupying a niche is limited to that niche's carrying capacity. Goldberg and Richardson (1987) then introduced a fitness sharing mechanism that induces population diversity by penalizing individuals for the presence of similar individuals in the population. As the number of individuals in a given niche increases, the availability of resources in the niche decreases, leading to an effective decrease in the viability of individuals in the niche, and the subsequent decrease in their numbers. To maintain a stable population in a niche, the population size must come into equilibrium with the availability of resources.

To illustrate the effect of sharing resources, consider derating an individual's fitness by an amount related to the number of similar individuals in the population. In particular, an individual's new shared fitness

$$f'_i \equiv \frac{f_i}{\sum_{j=1}^N sh(d_{ij})}, \quad (7.1)$$

where d_{ij} is the distance between individual i and j , and $sh(d_{ij})$ is the sharing function:

$$sh(d_{ij}) = \begin{cases} 1, & \text{if } d_{ij} = 0; \\ 1 - (\frac{d_{ij}}{\sigma_s})^\alpha, & \text{if } d_{ij} < \sigma_s; \\ 0, & \text{otherwise.} \end{cases} \quad (7.2)$$

N is the number of individuals in the population, and α and σ_s are parameters.[‡] Both genotypic and phenotypic distance measures can be employed, the appropriate choice

[‡]As pointed out by Smith et al. (1993) the condition $sh(d_{ij}) = 1$ for $d_{ij} = 0$ is only implied in previous fitness sharing studies. It is included here to clarify the limiting case corresponding to $\sigma_s = 0$.

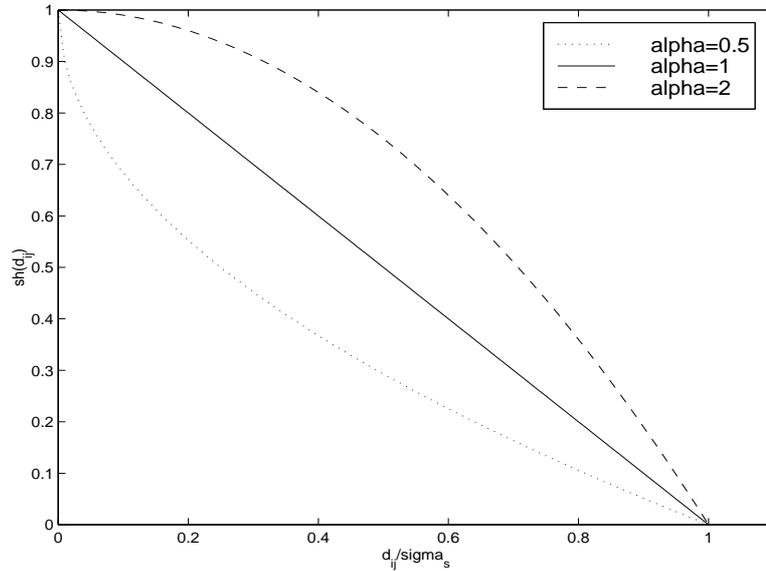


Figure 7.1: The sharing function $sh(d_{ij})$ for $\alpha=0.5, 1$ and 2 .

depending on the problem.

Note that since the distance between any individual and itself is zero (i.e., $d_{ij}=0$), the denominator of Equation 7.2 must be no less than one. If there are more individuals that are close together, Equation 7.1 yields smaller shared fitness.

α is a constant used to regulate the shape of the sharing function. Figure 7.1 illustrates three sharing functions for $\alpha=0.5, 1$ and 2 . σ_s is the more critical parameter in the fitness sharing scheme, which represents a cutoff distance, beyond which no sharing will occur. If σ_s is small enough that population individuals near a peak may not be sufficiently affected by other nearby individuals, the complete convergence to this peak may not occur because the nearby individuals may remain in the final population. In the limiting case, $\sigma_s = 0$ is such that each individual's fitness is divided by the number of identical population members. Let P_i^t be the proportion of individual i in the population at time t . Then under fitness proportionate selection

and the shared fitness f'_i , the value of P_i at time $t + 1$ is

$$\begin{aligned}
 P_i^{t+1} &= \frac{P_i^t f'_i}{\sum_{j=1}^{2^l} P_j^t f'_j} \\
 &= \frac{P_i^t \frac{f_i}{P_i^t}}{\sum_{j=1}^{2^l} P_j^t \frac{f_j}{P_j^t}} \\
 &= \frac{f_i}{\sum_{j=1}^{2^l} f_j},
 \end{aligned}$$

where l is the string length. In this case, selection assigns each individual a population proportion equal to its fitness relative to the sum of all fitness values in one step, without any search for peaks.

On the other hand, if σ_s is too large, and two peaks are within σ_s of one another, individuals at these peaks will affect one another's shared fitnesses, and the result is that the GA will easily converge to a population that contains one or the other of these peaks, but not both. In the limiting case where σ_s equals the maximum distance between two individuals (i.e., the radius of the search space), the fitness of each individual fitness is divided by the same value, and the effects of sharing are eliminated. One can thus expect the GA to converge towards a single type of individual in the usual fashion.

Fitness sharing is an effective technique for maintaining subpopulations over several high-fitness regions of the search space. However, it has two serious limitations: (1) the peaks must be equidistant or nearly so, and (2) setting σ_s requires knowledge about the number of peaks in the search space. These limitations arise from the fact that fitness sharing is defined explicitly.

Smith, Forrest and Perelson (1993) introduced an algorithm that does not require explicit construction of the sharing function, and thus avoids the difficulty of appropriately choosing σ_s . As will be seen in the following discussions, their approach can *implicitly* achieve fitness sharing that discovers for itself how many peaks are in the search space (including the case of not equally spaced peaks), and allocate trials

appropriately. The idea is to use the metaphor of biological immune systems which can maintain the diversity needed for it to detect multiple antigens. Then the GA, combined with the immune system idea, effectively distributes the population over several high-fitness areas of the search space.

7.4 Binary Immune System Model

The immune system model considered in this chapter is based on a model introduced by Farmer et al. (Farmer et al., 1986), where both antigens and antibodies are represented by binary strings. It is a simplification from the real biology in which genes are specified by a four-letter nucleic acid alphabet and recognition between antibodies and antigens is based on their three-dimensional shapes and physical properties. In addition, the model does not distinguish between receptors on B cells and the soluble, secreted form of the receptor, which is antibody. The model includes only recognition of the idealized antigens by receptors and does not consider how the immune system neutralizes an antigen once it is recognized. However, this abstract model of binary strings is rich enough for exploring how a relatively small number of recognizers (the antibodies) can evolve to recognize a much larger number of different patterns (the antigens).

In this binary immune system model, recognition is evaluated through a string matching procedure. The antigens are considered fixed, and a population of N antibodies is evolved to recognize the antigens using a GA. For any set of antigens, the goal is to obtain an antibody *cover*—a set of antibodies such that each antigen is recognized by at least one antibody in the population. Maintaining diverse antibodies is crucial for obtaining a cover (Smith, Forrest, and Perelson, 1993).

An antibody is said to match an antigen if their bit strings are complementary (maximally different). Since each antibody may have to match against several different antigens simultaneously, we do not require perfect bit-wise matching. Many

possible match rules are plausible physiologically (Perelson, 1989). The degree of match is quantified by a class of match score functions $M : Antigen \times Antibody \rightarrow \mathfrak{R}$. For example, M can simply count the number of complementary bits or M can identify contiguous regions of complementary bitwise matches within the string. Stadnyk (Stadnyk, 1987) introduced a function M that computes the lengths l_i of the complementary regions, and combines them such that long regions are rewarded more than short ones. Using this idea, many different specific functions can be defined that are linear or nonlinear in l_i .

Smith et al. (1993) adopted a model in which a fixed set of antigens is given, and the antibodies are initialized either to be completely random (to see if the GA can learn the correct antibodies) or initially given the answer by setting the population to include the correct antibodies (to test the stability of the answer). Their mechanism for fitness scoring is as follows:

1. A single antigen is randomly selected from the antigen population.
2. From the population of N antibodies a randomly selected sample of size σ is taken without replacement.
3. For each antibody in the sample, match it against the selected antigen, determine the number of bits that match, and assign it a match score.
4. The antibody in the sample population with the highest match score is determined. Ties are broken at random.
5. The match score of the winning antibody is added to its fitness. The fitness of all other antibodies remains unchanged.
6. This process is repeated for C cycles (typically one to three times the number of antibodies).

In this scheme, since an antibody's fitness is increased only if it is the best matching antibody in the sample, the fitness values of antibodies are interdependent. Forrest et al. (1993) have reported that this scheme can maintain subpopulations of antibodies that cover a set of antigens.

The next section discusses how this procedure implicitly embodies fitness sharing. The process is iterated so that each antigen in the antigen population has a chance of being selected and each antibody in the antibody population will receive a fair evaluation of its fitness.

7.4.1 Emergent Fitness Sharing in the Immune System Model

In this section, I briefly discuss how Smith et al.'s immune system model implicitly achieves fitness sharing. Before doing so, some new notation is required:

- d_{ij} represents the distance between an antibody i and an antigen j .
- $s(d_{ij})$ represents the match score assigned to antibody i when it is matched against antigen j .
- α_j represents the probability of selecting antigen j for matching.
- f_i represents the expected fitness of antibody i .

Note that there are several ways to define the distance between an antibody and an antigen. For example, d_{ij} can be the number of bits of antibody i that are not complementary to those in antigen j . (Other distance metrics include the 1, 0, # matching rule for classifiers, and r-contiguous bits (Percus, Percus, and Perelson, 1993), etc.) Under this distance metric, antibody i and antigen j are said to *perfectly match* if $d_{ij} = 0$, and *partially match* if $d_{ij} \neq 0$. The maximum distance possible between an antibody and an antigen is l , the bit string length.

Consider a given antigen j . Assume that the antibody population is of size N , and that it contains $N_j(m)$ antibodies at distance m from antigen j . The probability

that w antibodies at exactly distance m from antigen j are in a sample of size σ taken without replacement from this population, $p(w; \sigma, N, N_j(m))$, is:

$$p(w; \sigma, N, N_j(m)) = \frac{\binom{N_j(m)}{w} \binom{N - N_j(m)}{\sigma - w}}{\binom{N}{\sigma}}, w = 0, 1, \dots, \sigma. \quad (7.3)$$

As will be seen, the hypergeometric distribution above plays an important role in understanding how the immune system works. It is thus necessary to understand how it is obtained. Think of the $N_j(m)$ antibodies at distance m as successes and the remaining $N - N_j(m)$ antibodies as failures. We choose a sample of size σ without replacement and are interested in the probability of picking w success elements and $\sigma - w$ failure elements. There are $\binom{N}{\sigma}$ ways of selecting a sample of size σ . The number of ways of picking w successes from a total of $N_j(m)$ elements corresponds to the first term of the numerator in Equation 7.3; and the number of ways of picking $\sigma - w$ failures from $N - N_j(m)$ elements is the second term of the numerator of Equation 7.3. Thus, the fraction of times a sample is drawn with w success elements and $\sigma - w$ failure elements is shown as Equation 7.3.

Expected fitness of an antibody for the perfect matching case

To introduce the method of calculating the expected fitness of an antibody, Smith et al. first considered the perfect match case, where an antibody receives a non-zero score only if it perfectly matches the antigen—i.e., the match score $s(d_{ij}) \neq 0$ if and only if $d_{ij} = 0$. Let s_p be the score received for a perfect match.

Then, Smith et al. derived the calculation for the expected fitness of antibody i after one cycle:

$$f_i = \frac{s_p \alpha_j}{N_j(0)} (1 - p(0; \sigma, N, N_j(0))). \quad (7.4)$$

Note that the expected fitness for C cycles is $C f_i$. Since C will be a common factor for all expected fitness values, it will not have a bearing on the expected behavior of fitness proportionate selection, and thus it is not considered in the subsequent

discussion.

Comparing with the fitness sharing scheme, the term $s_p\alpha_j$ roughly corresponds to the height of the fitness function at point j in the l -dimensional hypercube. The expected fitness calculation indicates that this value is divided by $N_j(0)$, the number of individuals at that point. This corresponds to explicit fitness sharing with $\sigma_s = 0$ where an individual's fitness is divided by the number of identical individuals. The final hypergeometric term in the calculation is due to the sampling scheme, which will be clarified in the following discussion.

Expected fitness of an antibody for the partial matching case

Smith et al. then considered the general case in which an antibody receives a score for a partial match with an antigen at distance $d_{ij} = m$, where m ranges from 0 to l .

The calculation for the expected fitness of antibody i after one cycle is (see Smith et al. (1993)):

$$f_i = \sum_{m=0}^l \sum_{j \in S_i(m)} \frac{s(d_{ij})\alpha_j}{N_j(m)} p(0; \sigma, N, V_j(m)) (1 - p(0; \sigma, N - V_j(m), N_j(m))), \quad (7.5)$$

where $S_i(m)$ represents the set of all antigens j at distance m from antibody i (i.e., $d_{ij} = m$).

The hypergeometric term $p(0; \sigma, N, V_j(m))$ represents the probability that no antibody within distance $m-1$ of antigen j will be chosen in the sample. The other term $1 - p(0; \sigma, N - V_j(m), N_j(m))$ represents the probability that, given the previous condition, at least one copy of an antibody at distance m from antigen j will be in the sample.

To clarify the role of the hypergeometric terms in fitness sharing, two special cases are first examined. In case of sample size 1, Equation 7.5 yields

$$f_i = \sum_{m=0}^l \sum_{j \in S_i(m)} \frac{s(d_{ij})\alpha_j}{N}.$$

In this case, there is no fitness sharing, and essentially, the relative, expected fitness values are equivalent to those one would expect under a standard genetic algorithm—the GA would be expected to converge to a single type of antibody. (Note that this corresponds to fitness sharing with σ_s set to a value that spans the entire search space.)

As a second special case, we consider $\sigma = N$. If one assumes that a perfectly matching antibody exists for every available antigen in the population, the expected fitness reduces to

$$f_i = \frac{s_p \alpha_i}{N_i(0)}.$$

In this case, each antibody is only divided by its own effective proportion in the population. Like fitness sharing with $\sigma_s = 0$, the GA with fitness-proportionate selection would be expected to distribute the population based on relative fitness in one step, without a search for peak antibodies.

Smith et al. then further showed that the hypergeometric terms, in fact, correspond to a sharing function, and that σ plays a role in the immune system algorithm that is similar to that of σ_s in fitness sharing. Its value essentially indicates a cutoff beyond which no sharing can occur.

7.5 Experimental Results

To illustrate the effects of mate selection on the subpopulation-maintaining ability of Smith’s immune system model, I use a simple example in which antigen populations cannot be matched by a single antibody type. Consider an antigen population that is composed of 50% 000...000 (all 0’s) and 50% 111...111 (all 1’s). In order for an antibody population to recognize these antigens, there would need to be some antibodies that are all 1’s and others that are all 0’s. Thus, a solution to this problem requires the GA to maintain two different solutions simultaneously. This is an example of a “multiple peaks” problem because there are two incompatible solutions

that are maximally different. Typically, on multiple-peaks problems it is difficult for simple GAs to distribute the population over several peaks of a fitness landscape (two different subpopulations of antibodies that match two types of antigens, in this case). This is because the selection pressure in a simple standard GA usually entails strong convergence tendency to only one peak. Even without selection pressure, genetic drift due to sampling error can still lead the GA to (randomly) converge on one of the peaks (Goldberg and Segrest, 1987).

Forrest et al. (1993) reported in their numerical experiments that the GA with Smith's immune-based approach can effectively avoid strong convergence to one peak and distribute the population over multiple peaks. As has been discussed in the beginning of this chapter, I expect the mate selection schemes play an important role in maintaining subpopulations. In particular, my objective is to address the following questions concerning the capability of the GA, along with Smith's algorithm, for maintaining subpopulations:

- Can the GA with different mate selection schemes maintain stable subpopulations of antibodies for recognizing different antigens, or does it always converge on one peak? If it can maintain diverse subpopulations, then
- Is the proportion of antibodies in each subpopulation being affected by different mating preferences?[§]
- Do different mating preferences have influence on the discovery time of antigens?

In light of pattern-recognition, Forrest et al. (1993) pointed out that our immune

[§]How many antibody representatives must be in the population for an antigen to be identified is critical. As pointed out by Forrest et al. (1993), the presence of a single antibody is all that is required to recognize an antigen—a large number of clones that interconvert by mutation is not required. However, the immune system appears to have evolved redundant recognition. If only one antibody were used to recognize each antigen, then that antibody may fail to recognize the antigen when minor changes occur to the antigen. Thus, the immune system seems to use many different antibodies that recognize the antigen in different ways. In biology, typically tens or hundreds of different antibodies are found that match an antigen with sufficient strength to be important in an immune response.

Table 7.1: Building blocks of antigens.

$$\begin{aligned}
 b_1 &= 11111*****; s_1 = 10 \\
 b_2 &= *****11111*****; s_2 = 10 \\
 b_3 &= *****11111*****; s_3 = 10 \\
 b_4 &= *****11111; s_4 = 10 \\
 b_5 &= 00000*****; s_5 = 10 \\
 b_6 &= *****00000*****; s_6 = 10 \\
 b_7 &= *****00000*****; s_7 = 10 \\
 b_8 &= *****00000; s_8 = 10
 \end{aligned}$$

system needs to recognize bacteria partially on the basis of the existence of certain unusual molecules that are inherently different from human cells, since many bacteria have cell walls made from polymers that do not occur in humans. With this as motivation, I study the GA's ability to detect common patterns (building blocks) in the antigen population and adopt the building-block idea in the previous chapters to calculate fitnesses of antibodies.

Table 7.1 illustrates the building blocks of antigens $111\dots 1$ and $000\dots 0$. An antibody is said to match an antigen if its bit string is complementary to the antigen at certain building blocks. Specifically, the match score function M_b is to identify the building blocks for which an antibody matches an antigen, and then assign corresponding scores to that antibody. For example, given an antigen $111\dots 1$, an antibody with the first five and the last five bits being all 0's will receive score $s_1 + s_4 = 20$, since these ten bits are complementary to those of the antigen.

In Section 7.4, I discussed that Smith et al. considered two cases for the score calculation of antibodies—perfect match and partial match. In case of perfect match, an antibody receives a non-zero score only if it perfectly matches the antigen. In contrast, an antibody receives a non-zero score if it partially matches the antigen. In terms of the distance d_{ij} between antibody i and antigen j , partial match indicates the degree by which an antibody matches an antigen—i.e., the number of bits of

an antibody that are complementary to the corresponding bits of an antigen. The degree of match determines the specificity of an antibody. For example, if $d_{ij} = 0$, the matching is completely specific (that is, the antibody must perfectly match the antigen), but if $d_{ij} \neq 0$, it is partially matched. The consequence of a partial matching rule is that there is a trade-off between the number of antibodies used and their specificity—as the specificity of antibodies increases, so does the number of antibodies required to achieve a certain level of detection (Hofmeyr and Forrest, 2000).

For the scoring rule discussed in the building-block-based recognition problem, we can also expand its definition by allowing partial match. In other words, if an antibody matches an antigen at all the bits of a building block, it is a perfect building-block match; if not all the bits of that building block are required for matching, it constitutes a partial building-block match. Therefore, the perfect building-block match case is that an antibody scores if all of its bits at a building block are complementary to those of an antigen. On the other hand, a case for partial match could be to allow an antibody to score with only 80% bits (i.e., 4 bits in case of the building blocks shown in Table 7.1) of a building block at which it matches an antigen. The result of this flexible scoring is thus a smaller population size required to achieve a certain level of recognition performance. In this chapter, I mostly concentrate on this latter case for calculating antibody scores. (In case of the 100% building-block match, a few experiments conducted so far show similar qualitative results as the 80% building-block match case, but it requires much larger population sizes, i.e., much higher computational costs, to achieve similar levels of performance.)

7.5.1 Effects of Mate Selection on Maintaining Subpopulations

Using Smith et al.'s immune system algorithm (I call it the diversity algorithm from here on), antibodies are matched against antigens, scored according to their matching function, and evolved using a conventional GA. Forrest et al. (1993) showed

Table 7.2: Illustration of the immune-GA's mechanism.

1. Randomly generate an initial population of n antibodies.
2. Evaluate antibodies' fitnesses by the six steps of the diversity algorithm. (See Section 7.4.)
3. Repeat until n offspring have been created.
 - a. select a pair of parents for mating by particular selection schemes;
 - b. apply crossover operator;
 - c. apply mutation operator.
4. Reset all the new individuals' fitnesses to zero and
replace the current population with the new population.
5. Go to Step 2 until terminating condition.

that this approach can facilitate maintaining subpopulations of antibodies that recognize different antigens. To address the questions mentioned in the preceding subsection I conduct a series of GA experiments using the diversity algorithm as shown in Figure 7.2.[¶] My first objective is to investigate effects of mate selection on the diversity algorithm's subpopulation-maintaining ability. Unless stated otherwise, these experiments use an antibody population size of 100, mutation rate of 0.005, and ran for 150 generations. (Since crossover rates turn out to play a crucial role in maintaining subpopulations, I also present results based on various crossover rates.) The antigen population is 50% 000...0 and 50% 111...1, and both antigens and antibodies are binary strings of length 20. The number of samples, σ , is 10, which is 10% of the population size. I choose this value because Smith et al.'s analysis suggests that

[¶]Since in the diversity algorithm the match score of the winning antibody is continuously accumulated, after each generation the antibodies' fitness values are expected to be large. Thus at step 4 of Figure 7.2 I reset the fitnesses of the new population's individuals to zero after each generation to prevent fitnesses from unlimited increase.

too small or too large a sample size cannot show fitness sharing's effect. In addition, as mentioned in the preceding section, the number of cycles (C) does not have a bearing on the antibodies' expected fitnesses, 100 cycles (i.e., population size) used for each generation turned out to serve well for displaying subpopulation-maintaining results. Thus the total function evaluations for each run are generations \times cycles \times sample size, which equal 150,000.

Figure 7.2 illustrates the experimental results of the diversity algorithm (averaged over 50 runs), evolved by the GAs with tournament selection (TS), tournament dissimilar mating (TDM), tournament similar mating (TSM), random dissimilar mating (RDM) and random similar mating (RSM).

These are the results for the numbers of antibodies that recognize antigens when all four building blocks are 80% correctly matched. Note that only the curves with small error bars (95% confidence intervals) can be used for reliable judgements (I will discuss the reason for the larger error bars shortly), and thus the results for TS, TDM and RDM can be compared. It is clear that the dissimilar mating schemes, TDM and RDM, generate less desired antibodies than the regular tournament selection. The reason is in the following:

When crossover is turned on (crossover rate is 1, in this case), the dissimilarity-based mate selection increases the probability of producing useless hybrids—e.g., given an individual 111...1, and two candidate mates 111...1 and 000...0, the GAs with the dissimilar mating schemes tend to choose 000...0 for mating with 111...1, and the crossing-over between these two strings generates offspring that fall into the valley between the two peaks. Therefore, TDM and RDM maintain a smaller fraction of desired antibodies. (In the next subsection, I will discuss the effects of crossover rates, where we can see that this situation is improved when crossover is turned off.)

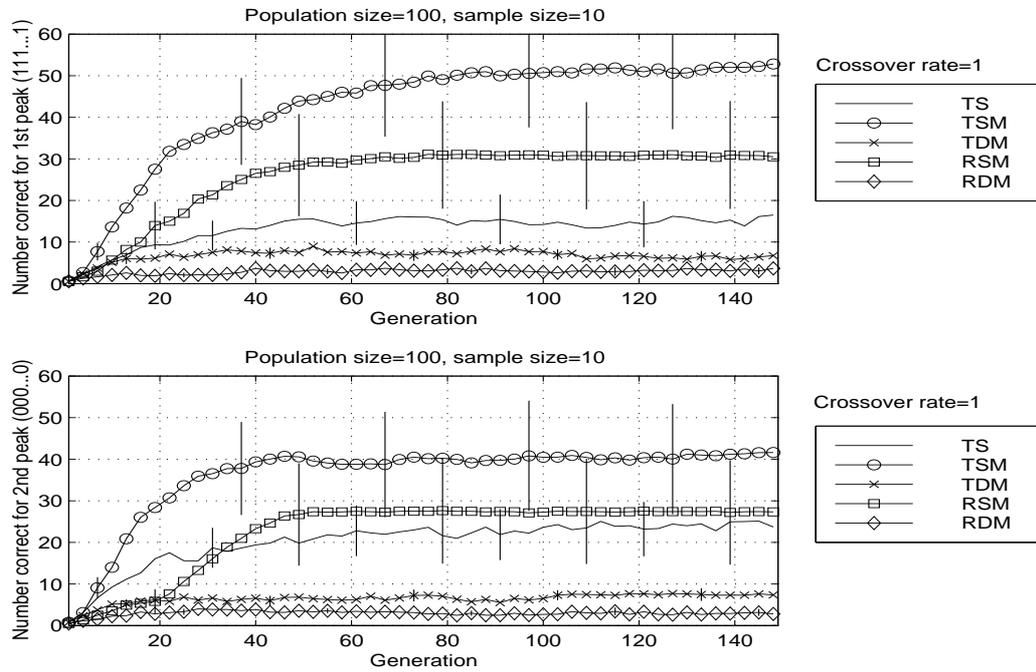


Figure 7.2: The number of antibodies that correctly recognize antigens.

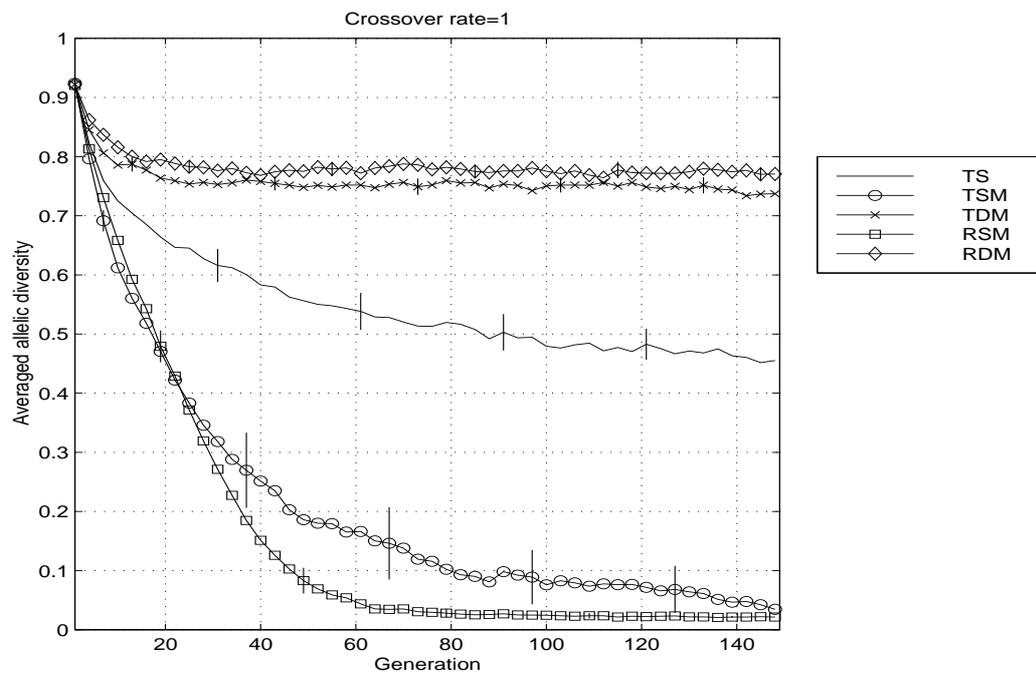


Figure 7.3: Averaged allelic diversity.

On the other hand, we see that TDM generates a larger fraction of desired antibodies than RDM. The difference between these two schemes is the method of selecting the second individual for mating—that is, in TDM fitter individuals have higher probabilities of being selected as mates, but this is not the case for RDM. As a result, TDM can pick out more individuals from the two peaks than RDM, which in turn increases the proportion of desired antibodies. This phenomenon can be seen in the results shown throughout this chapter.

A remedy for the problem of producing useless hybrids would be to reduce dissimilar mating rates. In terms of the example above, the regular tournament selection confers 111...1 and 000...0 with equal probability of being selected for mating, thereby reducing the likelihood of two mating individuals chosen from the two peaks. However, if individuals tend to select similar mates, the selection pressure toward these individuals may be strong enough that the GA's population converges on only one peak. If this is the case, the diversity algorithm's capability for maintaining subpopulation is degraded.

The larger error bars for TSM and RSM in Figure 7.2 illustrate this situation. Since TSM and RSM induce too strong a selection pressure, most of the GA's population members converge to only one peak. At generation 150, the GA with TSM has 25 (out of 50) runs in which most of the individuals converge to all 1's, and in 24 (out of 50) runs most of the individuals converge to all 0's, and there is one run in which the two peaks are lost. In case of RSM, there are 17 runs in which most of the individuals converge to all 1's, 15 runs in which most of the individuals converge to all 0's, and 18 runs where the two peaks are lost. Since TSM introduces an additional selection pressure toward higher-fitness individuals while choosing mates, it is more likely to pick out fitter individuals than RSM. This in turn results in the observed larger proportion of desired antibodies obtained based on TSM in Figure 7.2, or in more runs the GA's population converges on one peak when the maximum generations

in each run is limited.

We can use the allelic diversity metric employed in Section 3.3.5 to further compare the difference between these mate selection schemes. Recall that in Section 3.3.5 the allelic diversity metric D has a value of 1 when the proportion of 1s at each locus is 0.5 and 0 when all of the loci are fixed to either 0 or 1. Thus, if the population members are equally distributed over the two peaks (all 1's and all 0's), or the population is truly random, the calculation of D yields 1. By contrast, if all the population members converge on only one peak, the value of D is 0.

Figure 7.3 shows the averaged allelic diversity calculated for the GAs being tested. As can be seen that TDM and RDM produce allelic diversity values of between 0.7 and 0.8 when enough generations have passed. These values are relatively close to 1. Since Figure 7.2 shows that TDM and RDM do not generate large proportions of antibodies that match all 1's and all 0's, it implies the populations of the TDM and RDM GAs are close to a random composition. This is a consequence of using crossover rate 1. As we shall see in the later discussions, when crossover is turned off, the diversity algorithm based on TDM and RDM can really maintain subpopulations over the two peaks, instead of generating a nearly random population.

We can also see that the averaged allelic diversity of TS is smaller than that of the dissimilarity-based mating schemes. As shown in Figure 7.2, the larger error bars resulted from TS than from TDM and RDM imply that the antibody population created by TS is more skewed, thus generating smaller allelic diversity values. (Although TS can produce less offspring that fall into the valley by keeping away from choosing dissimilar mates, this in turn is likely to generate a more skewed antibody populations than TDM and RDM.)

On the other hand, since Figure 7.2 showed that the GA's population is likely to converge to one peak in cases of TSM and RSM, I would expect the calculations of allelic diversity metric for similar mating yield values close to 0 as the population

evolve. Figure 7.3 indeed agrees with our expectation.

A closer look at the micro-level dynamics gives us more understanding for the effect of similar mating. Figure 7.4 is the experimental results of a typical run for the number of desired antibodies obtained based on TSM. This figure shows that 000...0 are drown out by 111...1 in most of the generations, although they do show up in few early generations. This is because in TSM, similar individuals are always chosen as mates (with probability one)—a selection pressure toward similar mates enhances the convergence on one peak.

Using the schema density plots, Figure 7.5 clearly shows how the schemata of antibodies 000...0 are lost.

7.5.2 Effects of Crossover Rate

In the previous subsection, I showed that when crossover is turned on dissimilar mating can generate more useless hybrids to degrade the GA's performance with respect to maintaining antibody subpopulations. It is now natural to ask if reducing crossover rates would improve this situation. Our expectation is that the absence of crossover will prevent matings of individuals that are chosen from the two peaks by dissimilarity-based mating preferences.

Figures 7.6, 7.7 and 7.8 illustrate the results of decreasing crossover rates. Including Figure 7.2, we see that, for cases of TS, TDM and RDM, smaller crossover rates indeed reduce the likelihood of producing lower-fitness antibodies—that is, the proportions of desired antibodies increase as crossover rates decrease.

The effect of crossover rates on similar mating is not as obvious as can be seen on dissimilar mating. For TSM and RSM, the correct number of desired antibodies is always maintained at the range of 30 to 40, and the large error bars still imply that the GAs with the similar mate selection schemes are subject to strong convergence on one peak, no matter what crossover rates are. Consider the example illustrated in the

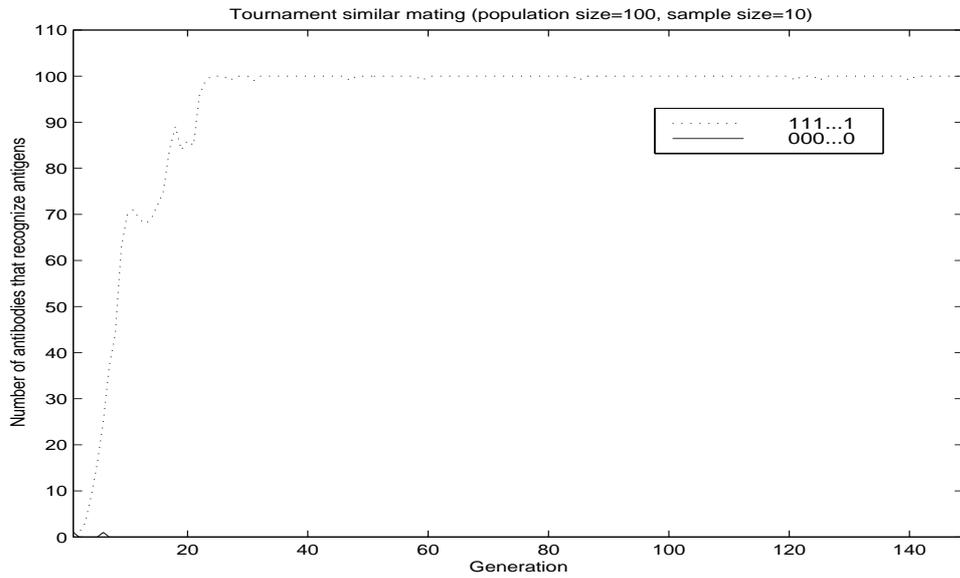


Figure 7.4: The number of antibodies that correctly recognize antigens (based on the tournament similar mating), where all portion of the solid line (i.e., corresponding to 000...0) after generation 10 is on the 0 level.

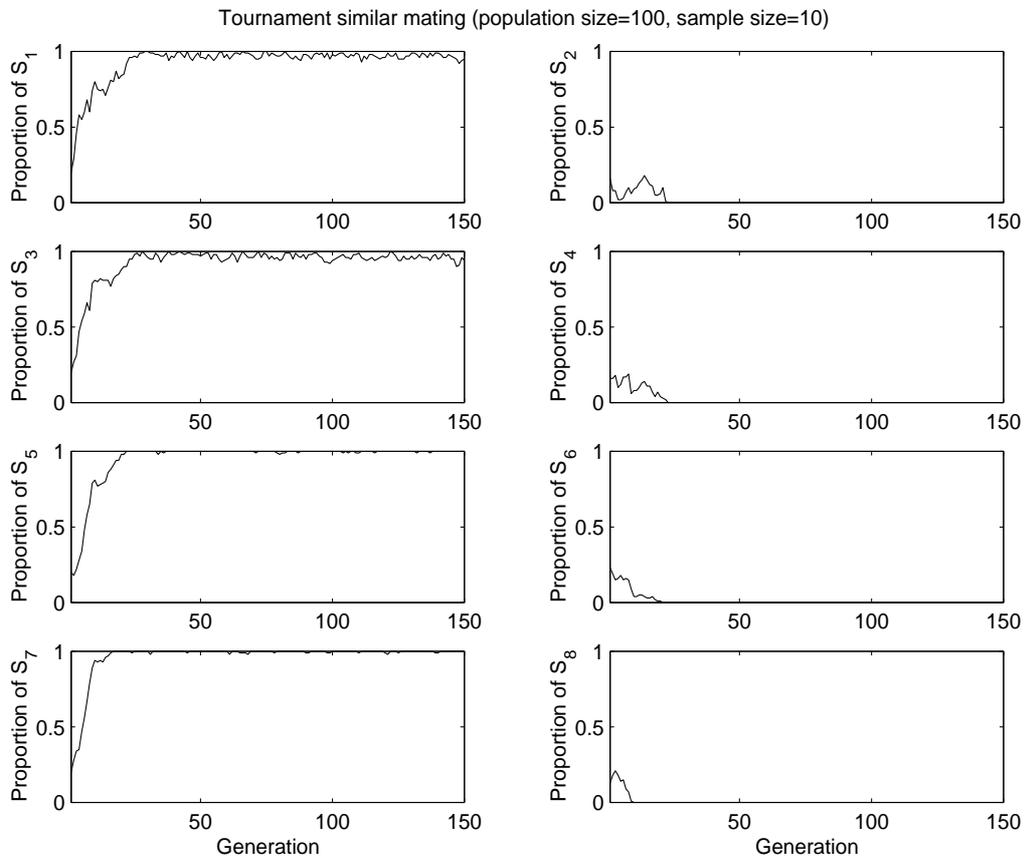


Figure 7.5: Schema dynamics for observing the effect of the tournament similar mating.

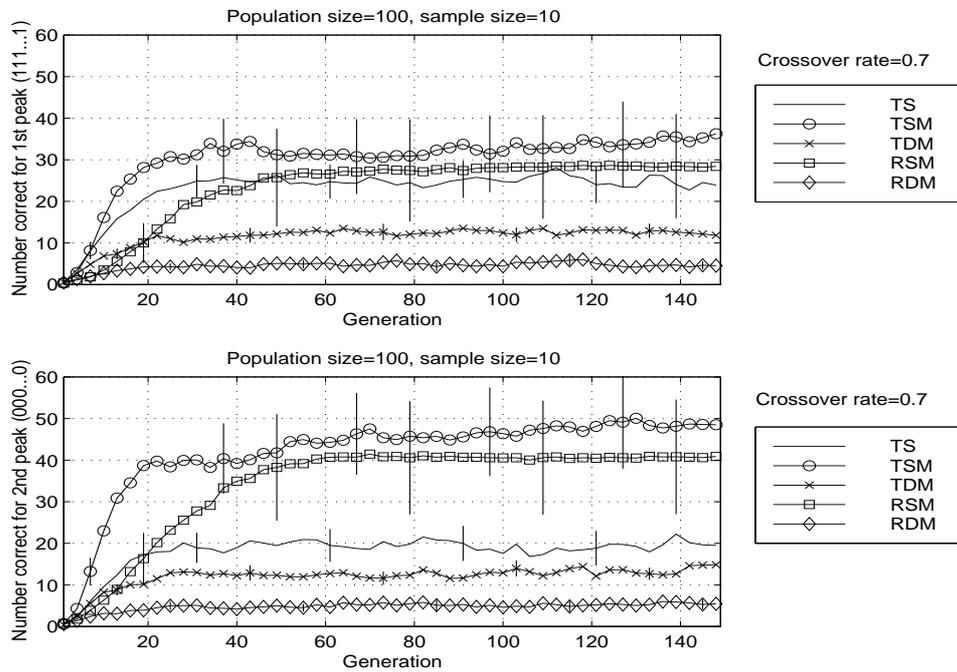


Figure 7.6: The number of antibodies that correctly recognize antigens (crossover rate = 0.7).

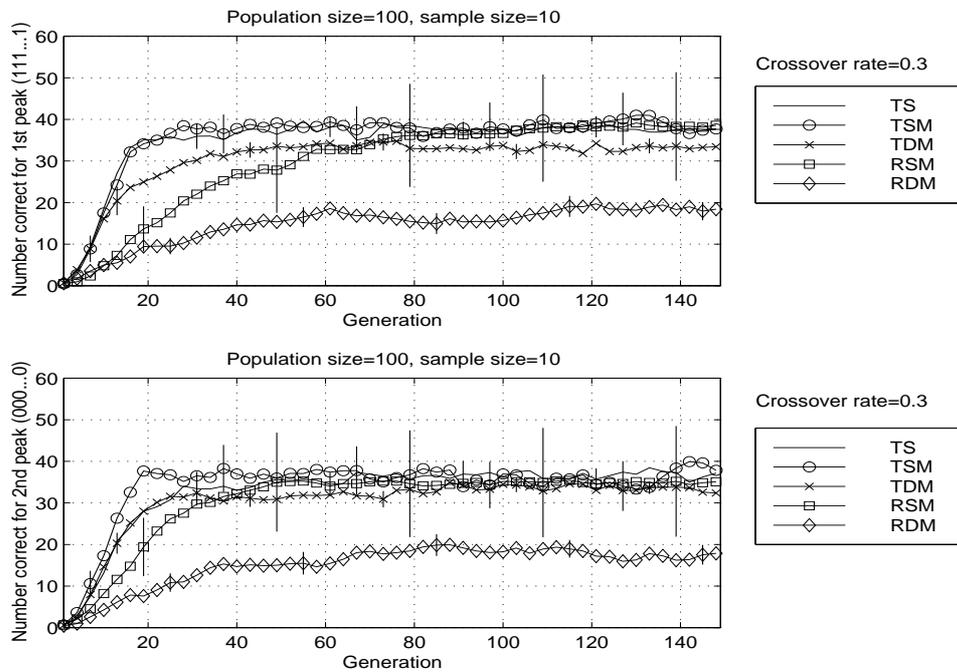


Figure 7.7: The number of antibodies that correctly recognize antigens (crossover rate = 0.3).

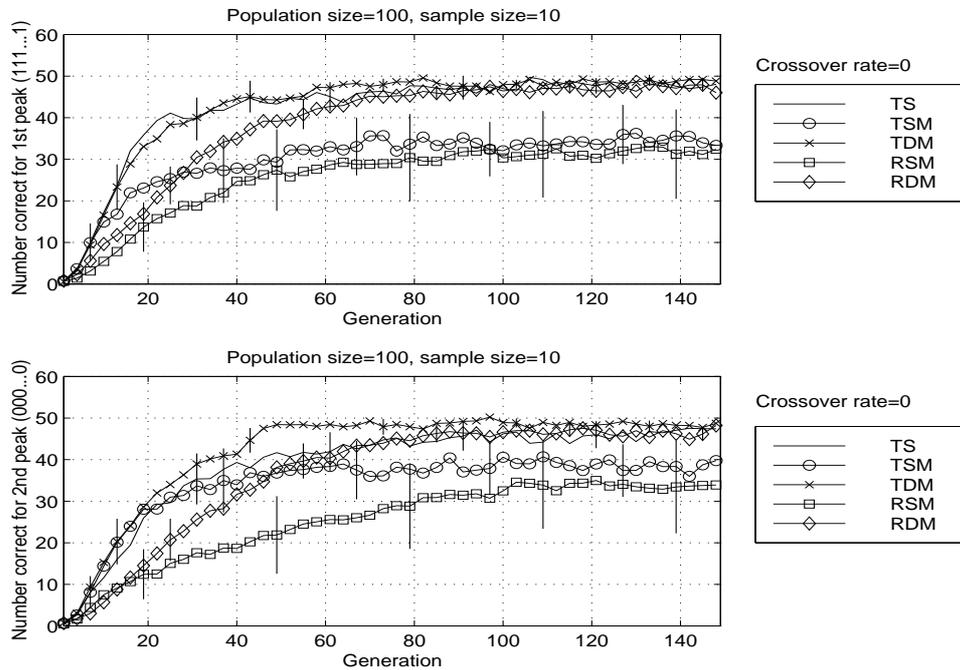


Figure 7.8: The number of antibodies that correctly recognize antigens (crossover rate = 0).

previous subsection. Given an individual $000\dots 0$ and two candidates $000\dots 0$ and $111\dots 1$, the similar mating GAs always choose $000\dots 0$ for mating with $000\dots 0$. Thus with or without crossover the mating between two identical individuals will not generate any difference. This explains why changing crossover rates does not affect results corresponding to the similar mating cases.

7.5.3 Effects of Mate Selection on the Discovery of Peaks

In the immune system problem considered, thus far I have been concerned with maintaining desired antibody subpopulations. However, there is another relevant issue I have not yet studied: the formation of the antibody subpopulations requires these antibodies to be discovered first. This is equivalent to the problem of finding multiple peaks. As has been shown in Chapter 3, the dissimilarity-based mate selection facilitates locating a single, best-so-far solution. Thus I am interested in investigating if dissimilar mating is also more beneficial in finding multiple peaks than

Table 7.3: The mean function evaluations of discovering antibodies 111...1 and 000...0 (over 50 runs).

Antibody	TS	TDM	RDM	TSM	RSM
111...1	2180 (255)	2300 (322)	2340 (243)	2140 (258)	49 runs reached
000...0	1860 (212)	1940 (188)	2320 (205)	1920 (240)	2420 (277)

Table 7.4: The number of runs (out of 50) in which antibodies 111...1 and 000...0 are discovered.

Antibody	TS	TDM	RDM	TSM	RSM
111...1	22	38	42	21	22
000...0	30	38	40	18	24

traditional selection schemes.

Table 7.3 displays the averaged mean function evaluations (over 50 runs) of discovering 111...1 and 000...0 based on population size 100, sample size 10, and crossover rate 1 (other parameter values are the same as those used previously). These results show no obvious difference between various mate selection schemes for finding the two peaks, except that there is one run in which one peak was not found before the maximum function evaluations 150,000 are reached. A closer inspection again shows the selection pressure toward similar individuals led that particular run of the GA to converge on 000...0, thereby precluding the discovery of 111...1. However, as population size decreases, the discrepancies between these mating schemes become more obvious. Table 7.4 illustrates the results for the number of runs (out of 50) in which antibodies 111...1 and 000...0 are discovered, respectively, based on population size 20, sample size 2 and crossover rate 1. (Other parameter values are the same as those used previously. The number of the total function evaluations in each run is equal to generation number \times cycles \times sample size (i.e., 6000).^{||}) It is clear that

^{||}Note that if the GA discovered the peak(s), the discovery usually occurs at very early generations. Thus 6,000 function evaluations are enough for comparing effects of various mate selection schemes.

the dissimilarity-based mating preferences facilitate locating two peaks. On the other hand, as discussed previously, similar mating introduces a selection pressure strong enough that the GAs perform worse than the other schemes in finding the two peaks. All this confirms with our expectation that the dissimilarity-based mate selection is beneficial in locating peaks.

7.6 Summary and Discussions

In this chapter, I have described Smith et al.'s immune system model in which subpopulations can be maintained through specific interactions among the strings. I have emphasized the performance of the GA in the binary immune system model, investigating how mate selection affects the GA's subpopulation-maintaining ability and the effects of mate selection on the discovery of multiple peaks. Both of these issues are important in the setting of multimodal function optimization, engineering and machine learning.

In studying the subpopulation-maintaining problem, the results illustrate that the dissimilar mating schemes are harmful in the sense of producing more lethal offspring. Consequently, the proportion of individuals that are representatives of different antibodies is reduced. I then showed that reducing the probability of dissimilar matings in the traditional selection can remedy this problem. I also hoped to improve the GA's performance by further increasing similar mating rates. However, as shown by the results obtained for TSM and RSM, they introduce a selection pressure strong enough that the population converges on only one peak. In addition, I have showed that reducing crossover rates can improve the GA's performance in maintaining subpopulations when the dissimilar mating and traditional selection schemes are employed, but this is not the case for the similarity-based mate selection schemes.

In studying the peaks-identifying problem, I showed that the dissimilarity-based mate selection schemes facilitate locating multiple peaks of the fitness landscape. This

is an extension of the results obtained in the previous chapters in which dissimilar mating is shown to be more likely to generate better performance in finding a single, best-so-far solution.

Since the pattern-recognition strategy in our approach was based on schema detection, it is worth further exploration because in real problems when there are many more antigens than antibodies, we need to detect common regions. In future work, I also hope to combine the solutions to the schema detection and multiple peaks problems in order to scale up our conclusions to realistic ratios of antigens and antibodies. In addition, I would like to develop an analytical analysis to enhance our understanding for mate selection in the context of the immune-GA-based system.

Finally, I would like to apply the lesson I learned to real applications. For example, in the control problem mentioned earlier, the Immunized Computational Systems in (Krishnakumar and Neidhoefer, 1999) consist of a hybrid structure whose building blocks, mimicking the immune building blocks, are composed of artificial neural networks, fuzzy systems and evolutionary algorithms. I hope to incorporate the understanding for mate selection into the evolutionary-algorithm part to see if I can improve the system's performance.

CHAPTER 8

CONCLUSIONS

8.1 Summary

As I discuss in Chapter 1, the process of information exchange among the GA's individuals involves two key components: crossover and mate selection. The central theme of this thesis concentrates on the investigation of mate selection in GAs. The goal of this thesis is to propose a framework that facilitates a systematic exploration of this subject.

Chapter 2 discusses, in depth, my motivation for conducting research on mate selection. Through a simple example in the context of GA, I indicate potential problems that occur to a simple implementation of GA. Then in Chapter 3, I propose the framework, in which by allowing individuals to select mates, fitnesses of candidate mates are dynamically re-scaled by the individual who chooses them. This results in a system in which population members' fitnesses depend on other individuals, rather than being determined only by the environment. I demonstrate that some biological phenomena, such as hitchhiking and the founder effect, which generally have significant negative effects on the GA's search power, can be reduced by dissimilar mating choices. I then proceed to examine performance of GAs with tournament-based mate selection. The results obtained further illustrate that dissimilarity-based mating preferences indeed help discriminate individuals and improve the GA's search power. The results also show that the GA's performance difference through various mate selection

schemes can be suppressed by large population sizes.

I also adopt a Markov chain model to analyze GAs as a theoretical basis for similarity and dissimilarity-based mate selection schemes, where I introduce an explicit way of describing how population individuals' fitnesses depend on others. Useful insights are obtained by means of both visual and computational exploration of the models. I characterize effects of various factors interacting with mate selection schemes, such as mutation rate, crossover, and difficulty of test functions. The results obtained enhance our understanding of the GA's behavior with different mate choices.

In Chapter 5, I continue the study using more complicated test functions for larger population sizes and string lengths. The investigation concentrates on a class of idealized building-block-based test functions, and an important hypothesis is drawn—given three mate selection schemes, TS, TDM and RDM, if one's goal is to search for best-so-far solutions (or the global optimum), the GA with TDM exhibits a higher (or equivalent) performance level than the GA with TS on either unimodal or multimodal problems. If a higher degree of multimodality presents sufficient difficulty to the GA's search, the GA using RDM is expected to outperform that using the other two schemes.

This hypothesis is then empirically validated in Chapter 6, based on more realistic, non-building-block-based problems. These positive results are encouraging since it means that the ideas of mate selection proposed in this thesis can be applied to practical problems.

Afterwards, in Chapter 7, I discuss a more general setting in the context of multimodal function optimization, engineering and machine learning. Two important goals are addressed: (1) how fast can the GA discover one or several peaks (or best-so-fars)? And (2) can the GA maintain diverse subpopulations in different parts of the search space? Since traditional GAs are subject to strong convergence and cannot maintain diverse subpopulations, I employ an immune system model which is shown

to effectively identify multiple peaks and maintain subpopulations. Then my experimental results show that the dissimilarity-based mate selection schemes facilitate locating multiple peaks of the fitness landscape. This is an extension of the results obtained in the previous chapters in which dissimilar mating is shown to be more likely to generate better performance in finding a single, best-so-far solution. On the other hand, with respect to the second goal, the results illustrate that the dissimilar mating schemes are harmful in the sense of reducing the proportion of individuals that are representatives of different antibodies. I then show that reducing the probability of dissimilar matings in the traditional selection can remedy this problem. I also show that reducing crossover rates can improve the GA's performance in maintaining subpopulations when the dissimilar mating and traditional selection schemes are employed, but this is not the case for the similarity-based mate selection schemes. All the results obtained in this chapter shed more light on how the mate selection schemes compare to traditional selection schemes.

In short, the contribution of the work in this thesis is that the importance of mate selection in the context of GA is identified, and what mating preferences must be used to improve the GA's performance are proposed. These results are obtained through investigating basic properties, testing an important hypothesis, and employing an immune-based model to further the understanding of the effects of mate selection on the GA's search power.

8.2 Future Work

In Chapter 4, I have discussed that the variance of the waiting times is also an important metric that can be derived from the mate selection Markov chain models. In addition, since only the mating schemes based on fitness proportionate selection are studied in that chapter, in future work I hope to develop a Markov model using tournament selection to further investigate the effects of tournament-based mate

selection schemes.

In the hypothesis testing of Chapter 6, although the empirical validation is positive, I would like to develop an analytic framework in which I can study the degree of the difficulty a problem presents to the GA's search. For example, it is worth further investigating if on unimodal problems, the GA with TDM always exhibits a higher (or equivalent) performance level than the GA with TS, and if on multimodal problems, the RDM GA can always achieve higher-level performance in terms of locating best-so-far solutions. Such a framework is important, since it would facilitate developing a guideline by which one can decide a better mating strategy for improving the GA's search power.

In Chapter 7, since the pattern-recognition strategy in my proposed approach was based on schema detection, it is worth further exploration because in real problems when there are many more antigens than antibodies, we need to detect common regions. In future work, I also hope to combine the solutions to the schema detection and multiple peaks problems in order to scale up our conclusions to realistic ratios of antigens and antibodies. In addition, I would like to develop an analytical analysis to enhance our understanding for mate selection in the context of the immune-GA-based system.

There are several ways to define the degree of similarity between individuals. In this thesis, I focus on using the Hamming distance as a similarity metric. The literature review presented in Section 2.3 illustrates several alternatives for calculating similarity between population individuals, such as the phenotypic distance used by Todd and Miller (1991). In future work, I thus hope to study mate selection based on different similarity metrics.

Since this thesis focuses on fixed-length, linear chromosomes, one of the other obvious extensions is to variable-length, nonlinear representations, such as those used in Genetic Programming (Koza, 1992). It is clear that proper similarity metrics

depend on problem domains and the algorithms used. Therefore, I would also need to select suitable similarity metrics for different problem domains.

So far the research of mate selection has focused on simple GAs. Since parallel GAs (PGAs) have been recognized to outperform simple GAs in general, I hope to continue the study of mate selection for PGAs in future.

In addition, I am also interested in other mate selection schemes; for instance, self-adaptive mate selection is a promising research line, including the existing tag-added, template-added mechanisms, and others. Since dissimilar mating schemes in general provide advantages in searching for the best-so-far solutions, I expect that, with respect to best-so-far performance, allowing mate selection to evolve would yield mating schemes that exhibit characteristics of dissimilar mate selection.

Finally, I would like to apply the results obtained in this thesis to practical problems, such as a manufacturing framework proposed in (Jacobs, 1995), and the other applications discussed in Section 7.2.

APPENDICES

APPENDIX A

Description of HDF H_1

First level (elementary) schemata: $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8$.

Second level schemata:

$$\begin{aligned} s_9 &= \{s_3, s_5\}, \\ s_{10} &= \{s_5, s_6\}, \\ s_{11} &= \{s_6, s_7\}, \\ s_{12} &= \{s_7, s_8\}. \end{aligned}$$

Third level schemata:

$$\begin{aligned} s_{13} &= \{s_2, s_3, s_5\}, \\ s_{14} &= \{s_3, s_5, s_6\}, \\ s_{15} &= \{s_6, s_7, s_8\}. \end{aligned}$$

Pothole schemata: $s_{16}, s_{17}, s_{18}, s_{19}, s_{20}, s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}, s_{27}, s_{28}, s_{29}$.

All schemata: $s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_8, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, s_{16}, s_{17}, s_{18}, s_{19}, s_{20}, s_{21}, s_{22}, s_{23}, s_{24}, s_{25}, s_{26}, s_{27}, s_{28}, s_{29}$.

Loci of all schemata: $s_1:\{7,9,10,12,13\}, s_2:\{15,17,19\}, s_3:\{30,31,32,33,34,36\}, s_4:\{41,42,43,44,45\}, s_5:\{45,46,47,48\}, s_6:\{50,52,53,54\}, s_7:\{53,54,55,56,57,58,59,60\}, s_8:\{56,57,58,60,61\}, s_9:\{30,31,32,33,34,36,45,46,47,48\}, s_{10}:\{45,46,47,48,50,52,53,54\}, s_{11}:\{50,52,53,54,55,56,57,58,59,60\}, s_{12}:\{53,54,55,56,57,58,59,60,61\}, s_{13}:\{15,17,19,30,31,32,33,34,36,45,46,47,48\}, s_{14}:\{30,31,32,33,34,36,45,46,47,48,50,52,53,54\}, s_{15}:\{50,52,53,54,55,56,57,58,59,60,61\}, s_{16}:\{7,9,10,12,13,15\}, s_{17}:\{15,17,19,32\}, s_{18}:\{30,31,32,33,34,36,45\}, s_{19}:\{41,42,43,44,45,48\}, s_{20}:\{45,46,47,48,50\}, s_{21}:\{50,52,53,54,59\}, s_{22}:\{53,54,55,56,57,58,59,60,61\}, s_{23}:\{12,15,17,19\}, s_{24}:\{17,30,31,32,33,34,36\}, s_{25}:\{34,41,42,43,44,45\}, s_{26}:\{42,45,46,47,48\}, s_{27}:\{46,50,52,53,54\}, s_{28}:\{50,53,54,55,56,57,58,59,60\}, s_{29}:\{54,56,57,58,60,61\}$.

Alleles of all schemata: $s_1:\{1,0,0,0,1\}, s_2:\{1,0,0\}, s_3:\{0,0,0,1,1,1\}, s_4:\{0,1,1,1,0\}, s_5:\{1,0,1,0\}, s_6:\{1,1,1,0\}, s_7:\{0,1,1,0,0,1,1,0\}, s_8:\{1,1,1,0,1\}, s_9:\{0,0,0,1,1,1,0,1,0\}, s_{10}:\{1,0,1,0,1,1,1,0\}, s_{11}:\{1,1,1,0,1,0,0,1,1,0\}, s_{12}:\{0,1,1,0,0,1,1,0,1\}, s_{13}:\{1,0,0,0,0,0,$

$1,1,1,1,0,1,0\}$, $s_{14}:\{0,0,0,1,1,1,0,1,0,1,1,1,0\}$, $s_{15}:\{1,1,1,0,1,0,0,1,1,0,1\}$, $s_{16}:\{1,0,0,0,1,1\}$, $s_{17}:\{1,0,0,0\}$, $s_{18}:\{0,0,0,1,1,1,0\}$, $s_{19}:\{0,1,1,1,0,0\}$, $s_{20}:\{1,0,1,0,1\}$, $s_{21}:\{1,1,1,0,1\}$, $s_{22}:\{1,1,1,0,1\}$, $s_{23}:\{0,1,1,0,0,1,1,0,1\}$, $s_{24}:\{0,1,0,0\}$, $s_{25}:\{0,0,0,0,1,1,1\}$, $s_{26}:\{1,0,1,1,1,0\}$, $s_{27}:\{1,1,0,1,0\}$, $s_{28}:\{1,0,1,1,0,0,1,1,0\}$, $s_{29}:\{1,1,1,1,0,1\}$.

Values of all schemata: $\{u(s_1)=4, u(s_2)=4, u(s_3)=5, u(s_4)=6, u(s_5)=4, u(s_6)=6, u(s_7)=5, u(s_8)=5, u(s_9)=11, u(s_{10})=10, u(s_{11})=11, u(s_{12})=12, u(s_{13})=10, u(s_{14})=12, u(s_{15})=10, u(s_{16})=-1, u(s_{17})=-1, u(s_{18})=-1, u(s_{19})=-1, u(s_{20})=-1, u(s_{21})=-1, u(s_{22})=-1, u(s_{23})=-1, u(s_{24})=-1, u(s_{25})=-1, u(s_{26})=-1, u(s_{27})=-1, u(s_{28})=-1, u(s_{29})=-1\}$.

APPENDIX B

Description of HDF H_2

Elementary schemata: $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}$.

Corresponding starting loci of elementary schemata : $\{17, 22, 55, 74, 101, 126, 147, 156\}$.

Corresponding lengths of elementary schemata: $\{10, 7, 9, 6, 8, 10, 9, 10\}$.

Combinant schemata (s_9 – s_{35}):

$\{\{s_1, s_2\}, \{s_1, s_3\}, \{s_2, s_3\}, \{s_2, s_4\}, \{s_3, s_4\}, \{s_3, s_5\}, \{s_4, s_5\}, \{s_4, s_6\}, \{s_6, s_8\}, \{s_7, s_8\}, \{s_1, s_2, s_3\}, \{s_2, s_4, s_5\}, \{s_2, s_4, s_6\}, \{s_4, s_6, s_8\}, \{s_6, s_7, s_8\}, \{s_3, s_4, s_5, s_6\}, \{s_4, s_5, s_6, s_8\}, \{s_4, s_6, s_7, s_8\}, \{s_5, s_6, s_7, s_8\}, \{s_2, s_3, s_4, s_5, s_6\}, \{s_2, s_4, s_6, s_7, s_8\}, \{s_4, s_5, s_6, s_7, s_8\}, \{s_2, s_3, s_4, s_6, s_7, s_8\}, \{s_2, s_4, s_5, s_6, s_7, s_8\}, \{s_3, s_4, s_5, s_6, s_7, s_8\}, \{s_1, s_2, s_3, s_4, s_6, s_7, s_8\}, \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8\}\}$.

Pothole Schemata: s_{36} – s_{49} .

Corresponding loci for each schema s_i , $i = 1 \dots 49$: $\{\{18, 19, 20, 21, 22, 24, 27\}, \{23, 24, 25, 29\}, \{56, 57, 58, 59, 60, 61\}, \{75, 77\}, \{102, 103, 105, 106, 107, 109\}, \{127, 128, 129, 130, 131, 133, 135\}, \{149, 150, 152, 153, 154, 155, 156\}, \{157, 159, 161, 162, 164, 165, 166\}, \{18, 19, 20, 21, 22, 24, 27, 23, 25, 29\}, \{18, 19, 20, 21, 22, 24, 27, 56, 57, 58, 59, 60, 61\}, \{23, 24, 25, 29, 56, 57, 58, 59, 60, 61\}, \{23, 24, 25, 29, 75, 77\}, \{56, 57, 58, 59, 60, 61, 75, 77\}, \{56, 57, 58, 59, 60, 61, 102, 103, 105, 106, 107, 109\}, \{75, 77, 102, 103, 105, 106, 107, 109\}, \{75, 77, 127, 128, 129, 130, 131, 133, 135\}, \{127, 128, 129, 130, 131, 133, 135, 157, 159, 161, 162, 164, 165, 166\}, \{149, 150, 152, 153, 154, 155, 156, 157, 159, 161, 162, 164, 165, 166\}, \{18, 19, 20, 21, 22, 24, 27, 23, 25, 29, 56, 57, 58, 59, 60, 61\}, \{23, 24, 25, 29, 75, 77, 102, 103, 105, 106, 107, 109\}, \{23, 24, 25, 29, 75, 77, 127, 128, 129, 130, 131, 133, 135\}, \{75, 77, 127, 128, 129, 130, 131, 133, 135, 157, 159, 161, 162, 164, 165, 166\}, \{127, 128, 129, 130, 131, 133, 135, 149, 150, 152, 153, 154, 155, 156, 157, 159, 161, 162, 164, 165, 166\}\}$.

APPENDIX C

Description of HDF H_3

Elementary schemata: $\{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}\}$.

Corresponding starting loci of elementary schemata : $\{33, 73, 111, 194, 239, 308, 315, 325, 338, 344\}$.

Corresponding lengths of elementary schemata: $\{10, 7, 11, 11, 10, 7, 5, 7, 10, 8\}$.

Combinant schemata (s_{11} – s_{65}):

$\{\{s_2, s_3\}, \{s_2, s_4\}, \{s_3, s_4\}, \{s_3, s_5\}, \{s_4, s_5\}, \{s_4, s_6\}, \{s_5, s_7\}, \{s_6, s_7\}, \{s_7, s_8\}, \{s_7, s_9\}, \{s_8, s_9\}, \{s_8, s_{10}\},$
 $\{s_9, s_{10}\}, \{s_1, s_2, s_3\}, \{s_1, s_3, s_4\}, \{s_1, s_3, s_5\}, \{s_2, s_3, s_4\}, \{s_2, s_3, s_5\}, \{s_3, s_4, s_6\}, \{s_3, s_5, s_7\}, \{s_4, s_5, s_7\},$
 $\{s_5, s_6, s_7\}, \{s_5, s_7, s_8\}, \{s_5, s_7, s_9\}, \{s_6, s_7, s_8\}, \{s_6, s_7, s_9\}, \{s_6, s_8, s_9\}, \{s_6, s_8, s_{10}\}, \{s_7, s_8, s_9\}, \{s_7, s_8, s_{10}\},$
 $\{s_7, s_9, s_{10}\}, \{s_2, s_3, s_4, s_6\}, \{s_2, s_4, s_5, s_7\}, \{s_3, s_4, s_5, s_7\}, \{s_3, s_5, s_7, s_8\}, \{s_3, s_5, s_7, s_9\}, \{s_4, s_6, s_7, s_8\},$
 $\{s_4, s_6, s_8, s_9\}, \{s_5, s_6, s_8, s_9\}, \{s_6, s_7, s_8, s_{10}\}, \{s_6, s_7, s_9, s_{10}\}, \{s_1, s_2, s_3, s_4, s_6\}, \{s_2, s_4, s_6, s_8, s_9\}, \{s_3, s_4,$
 $s_5, s_6, s_7\}, \{s_3, s_4, s_5, s_7, s_8\}, \{s_4, s_5, s_6, s_7, s_8\}, \{s_4, s_5, s_6, s_8, s_9\}, \{s_6, s_7, s_8, s_9, s_{10}\}, \{s_1, s_2, s_4, s_6, s_8, s_9\},$
 $\{s_2, s_3, s_4, s_6, s_8, s_9\}, \{s_4, s_5, s_6, s_7, s_8, s_9\}, \{s_1, s_2, s_3, s_4, s_6, s_8, s_9\}, \{s_2, s_3, s_4, s_5, s_6, s_7, s_8\}, \{s_2, s_3, s_4, s_5,$
 $s_6, s_8, s_9\}, \{s_1, s_2, s_3, s_4, s_5, s_6, s_8, s_9\}\}$.

Pothole Schemata: s_{66} – s_{115} .

Corresponding loci for each schema s_i , $i = 1 \dots 115$: $\{\{34, 35, 36, 38, 39, 41, 43\}, \{74, 76, 78, 79, 80\}, \{113, 114, 115, 116, 117, 119, 121\}, \{195, 196, 197, 199, 200, 201, 203, 204, 205\}, \{240, 241, 242, 243, 244, 247, 248, 249\}, \{310, 312, 313, 314, 315\}, \{317, 318, 319, 320\}, \{326, 327, 330, 331, 332\}, \{339, 340, 341, 343, 345, 346, 347\}, \{345, 346, 347, 349, 350, 351, 352\}, \{74, 76, 78, 79, 80, 113, 114, 115, 116, 117, 119, 121\}, \{74, 76, 78, 79, 80, 195, 196, 197, 199, 200, 201, 203, 204, 205\}, \{113, 114, 115, 116, 117, 119, 121, 195, 196, 197, 199, 200, 201,$

203, 204, 205}, {113, 114, 115, 116, 117, 119, 121, 240, 241, 242, 243, 244, 247, 248, 249}, {195, 196, 197, 199, 200, 201, 203, 204, 205, 240, 241, 242, 243, 244, 247, 248, 249}, {195, 196, 197, 199, 200, 201, 203, 204, 205, 310, 312, 313, 314, 315}, {240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320}, {310, 312, 313, 314, 315, 317, 318, 319, 320}, {317, 318, 319, 320, 326, 327, 330, 331, 332}, {317, 318, 319, 320, 339, 340, 341, 343, 345, 346, 347}, {326, 327, 330, 331, 332, 339, 340, 341, 343, 345, 346, 347}, {326, 327, 330, 331, 332, 345, 346, 347, 349, 350, 351, 352}, {339, 340, 341, 343, 345, 346, 347, 349, 350, 351, 352}, {34, 35, 36, 38, 39, 41, 43, 74, 76, 78, 79, 80, 113, 114, 115, 116, 117, 119, 121}, {34, 35, 36, 38, 39, 41, 43, 113, 114, 115, 116, 117, 119, 121, 195, 196, 197, 199, 200, 201, 203, 204, 205}, {34, 35, 36, 38, 39, 41, 43, 113, 114, 115, 116, 117, 119, 121, 240, 241, 242, 243, 244, 247, 248, 249}, {74, 76, 78, 79, 80, 113, 114, 115, 116, 117, 119, 121, 195, 196, 197, 199, 200, 201, 203, 204, 205}, {74, 76, 78, 79, 80, 113, 114, 115, 116, 117, 119, 121, 240, 241, 242, 243, 244, 247, 248, 249}, {113, 114, 115, 116, 117, 119, 121, 195, 196, 197, 199, 200, 201, 203, 204, 205, 310, 312, 313, 314, 315}, {113, 114, 115, 116, 117, 119, 121, 240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320}, {195, 196, 197, 199, 200, 201, 203, 204, 205, 240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320}, {240, 241, 242, 243, 244, 247, 248, 249, 310, 312, 313, 314, 315, 317, 318, 319, 320}, {240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320, 326, 327, 330, 331, 332}, {240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320, 339, 340, 341, 343, 345, 346, 347}, {310, 312, 313, 314, 315, 317, 318, 319, 320, 326, 327, 330, 331, 332}, {310, 312, 313, 314, 315, 317, 318, 319, 320, 339, 340, 341, 343, 345, 346, 347}, {310, 312, 313, 314, 315, 326, 327, 330, 331, 332, 339, 340, 341, 343, 345, 346, 347}, {310, 312, 313, 314, 315, 326, 327, 330, 331, 332, 345, 346, 347, 349, 350, 351, 352}, {317, 318, 319, 320, 326, 327, 330, 331, 332, 339, 340, 341, 343, 345, 346, 347}, {317, 318, 319, 320, 326, 327, 330, 331, 332, 345, 346, 347, 349, 350, 351, 352}, {317, 318, 319, 320, 339, 340, 341, 343, 345, 346, 347}, {317, 318, 319, 320, 326, 327, 330, 331, 332, 339, 340, 341, 343, 345, 346, 347, 349, 350, 351, 352}, {74, 76, 78, 79, 80, 113, 114, 115, 116, 117, 119, 121, 195, 196, 197, 199, 200, 201, 203, 204, 205, 310, 312, 313, 314, 315}, {74, 76, 78, 79, 80, 195, 196, 197, 199, 200, 201, 203, 204, 205, 240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320}, {113, 114, 115, 116, 117, 119, 121, 195, 196, 197, 199, 200, 201, 203, 204, 205, 240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320, 326, 327, 330, 331, 332}, {113, 114, 115, 116, 117, 119, 121, 240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320, 326, 327, 330, 331, 332}, {113, 114, 115, 116, 117, 119, 121, 240, 241, 242, 243, 244, 247, 248, 249, 317, 318, 319, 320, 339, 340, 341, 343, 345, 346, 347}, {195, 196, 197, 199, 200, 201, 203, 204, 205, 310, 312, 313, 314, 315, 317, 318, 319, 320, 326, 327, 330, 331, 332}, {195, 196, 197, 199, 200, 201, 203, 204, 205, 310, 312, 313, 314, 315, 326, 327, 330, 331, 332, 339, 340, 341, 343, 345, 346, 347}, {240, 241, 242, 243, 244, 247, 248, 249, 310, 312, 313, 314, 315, 326,

5, 6, 6, 10, 8, 8, 10, 10, 8, 9, 10, 10, 10, 9, 10, 9, 8, 9, 10, 8, 10, 10, 9, 10, 10, 9, 10, 8, 8, 9, 9, 10, 8, 9, 9, 10, 9, 9, 9,
8, 8, 10, 9, 10, 8, 9, 8, 8, 8, 10, 9, 8, 8, 8, 10, 8, 9, 10, -2, -2, -1, -2, -2, -2, -1, -2, -2, -1, -2, -2, -2, -2, -2, -2, -2, -2, -2,
-2, -2, -2, -2, -2, -2, -1, -1, -2, -1, -2, -1, -2, -2, -2, -1, -2, -1, -2, -2, -1, -1, -1, -2, -1, -1, -2, -2, -2, -2, -1}.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [Bäck, 1994] Bäck, T. (1994). Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pp. 57-62.
- [Baker, 1985] Baker, J. E. (1985). Adaptive selection methods for genetic algorithms. *Proc. 1st International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 101-111.
- [Baker, 1987] Baker, J. E. (1987). Reducing bias and inefficiency in the selection algorithm. *Genetic Algorithms and Their Applications: Proceedings of the 2nd International Conference on Genetic Algorithms*, pp. 14-21.
- [Bagchi, Uckun, Miyabe, and Kawamura, 1991] Bagchi, S., Uckun, S., Miyabe, Y., and Kawamura, K. (1991). Exploring problem-specific recombination operators for job shop scheduling. *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 10-17.
- [Bean, 1994] Bean, J. C. (1994). Genetics and random keys for sequencing and optimization, *ORSA J Comput* 6, 1994, pp. 154-160.
- [Belding, 2001] Belding, T. (2001). Potholes on the royal road. *Proc. of 2001 Genetic and Evolutionary Computation Conference (GECCO-2001)*, Morgan Kaufmann, pp. 211-218.
- [Booker, 1982] Booker, L. B. (1982). Intelligent behavior as an adaptation to the task environment (Doctoral dissertation, University of Michigan). *Dissertation Abstracts International*, 43(2), 469B. (University Microfilms No. 8214966)
- [Booker, 1985] Booker, L. B. (1985). Improving the performance of genetic algorithms in classifier systems. *Proc. of International Conference on Genetic Algorithms and Their Applications*, pp. 80-92.
- [Booker, 1987] Booker, L. B. (1987). Improving search in genetic algorithms. In L. Davis (Ed.), *Genetic algorithms and simulated annealing*, pp. 61-73. London: Pitman.
- [Box, 1957] Box, G. E. P. (1957). Evolutionary operation: a method of increasing industrial productivity. *Applied Statistics* 6, pp. 81-101.

- [Brindle, 1981] Brindle, A. (1981). *Genetic Algorithms for Function Optimization*. Unpublished doctoral dissertation, University of Alberta, Edmonton.
- [Cantú-Paz, 1997] Cantú-Paz, E. (1997). A survey of parallel genetic algorithms. IlliGAL Report No. 97003. Urbana, IL: University of Illinois at Urbana-Champaign.
- [Chen, 1999] Chen, S. (1999). *Is the Common Good? A New Perspective Developed in Genetic Algorithms*, Ph. D. diss., Robotics institute, Carnegie Mellon University.
- [Collins and Jefferson, 1991] Collins, R. J., and Jefferson, D. R. (1991). Selection in massively parallel genetic algorithms. *Proceedings of the Fourth International Conference on Genetic Algorithms*, pp. 249-256.
- [Cook, 1991] Cook, L. M. (1991). *Genetic and ecological diversity: The sport of nature*. London: Chapman & Hall.
- [Costa, Tavares, and Rosa, 1999] Costa, J., Tavares, R., Rosa, A.C. (1999). An experimental study on dynamic random variation of population size. *Proc. of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 607-612.
- [Craighurst and Martin, 1995] Craighurst, R. and Martin, W. (1995). Enhancing GA performance through crossover prohibitions based on ancestry. *Proc. 6th International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 130-135.
- [Dallery, David, and Xie, 1989] Dallery, Y., David, R., and Xie, X.-L. (1989). Approximate Analysis of Transfer Lines with Unreliable Machines and Finite Buffers. *IEEE Trans. Automat. Contr.*, 34: pp. 943-953.
- [Das and Whitley, 1991] Das, R., and Whitley, L. D. (1991). The only challenging problems are deceptive: Global search by solving order 1 hyperplanes. *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 166-173.
- [Davis and Principe, 1991] Davis, T. E., and Principe, J. C. (1991). A simulated annealing like convergence theory for the simple genetic algorithm. *Proceedings of the 4th International Conference on Genetic Algorithms*, San Diego, 174-181.
- [Deb and Goldberg, 1989] Deb, K. and Goldberg, D. E. (1989). An investigation of niche and species formation in genetic function optimization. , *Proc. 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 42-50.
- [De Castro and Von Zuben, 2000] De Castro, L. N. and Von Zuben, F. J. (2000). Artificial Immune Systems: Part II A Survey of Applications, *Technical Report RT DCA 02/00*.

- [De Jong, 1975] De Jong, K. (1975). *Analysis of the behavior of a class of genetic adaptive systems*. Ph. D. thesis, University of Michigan, Ann Arbor.
- [De Jong, Spears, and Gordon, 1994] De Jong, K., Spears, W., and Gordon, D. (1994). Using Markov chains to analyze GAFOs. *Foundations of Genetic Algorithms*, Volume 3, Morgan Kaufmann, pp. 115-137.
- [Dobzhansky, 1937] Dobzhansky, T. (1937). *Genetics and the Origin of Species*. (Reprinted 1982). New York: Columbia University Press.
- [Eshelman and Schaffer, 1991] Eshelman, L. J. and Schaffer, J. D. (1991). Preventing premature convergence in genetic algorithms by preventing incest. *Proc. 4th International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 115-122.
- [Farmer, Packard and Perelson, 1986] Farmer, J. D., Packard, N. H., and Perelson, A. S. (1986). The immune system, adaptation, and machine learning. *Evolution, Games and Learning*. NorthHolland. (Reprinted from *Physica*, 22D, pp. 187-204)
- [Fernandes, Tavares, Munteanu, and Rosa 2001] Fernandes, C., Tavares, R., Munteanu, C. and Rosa, A. (2001). Assortative mating in genetic algorithms for vector quantization problems. *Proceedings of the 2001 ACM Symposium of Applied Computing*, pp. 361-365.
- [Fisher, 1930] Fisher, R. A. (1930). *The Genetical Theory of Natural Selection*. Oxford University Press.
- [Fogel, Owens, and Walsh, 1966] Fogel, L., Owens, A., and Walsh, M. (1966). *Artificial Intelligence Through Simulated Evolution*. John Wiley.
- [Fogel, 1993a] Fogel, D. B. (1993). Evolving behaviours in the iterated prisoner's dilemma. *Evolutionary Computation*, 1(1): pp. 77-97.
- [Fogel, 1993b] Fogel, D. B. and Atmar, W. (1993). *Proceedings of the Second Annual Conference on Evolutionary Programming*, La Jolla, CA, Evolutionary Programming Society.
- [Fogel, 1995] Fogel, D. B. (1995). *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ.
- [Forrest, 1985] Forrest, S. (1985). Scaling fitnesses in the genetic algorithm. In *Documentation for PRISONERS DILEMMA and NORMS Programs That Use the Genetic Algorithm*. Unpublished manuscript.
- [Forrest and Mitchell, 1993] Forrest, S., and Mitchell, M. (1993). Relative building block fitness and the building block hypothesis. *Foundations of Genetic Algorithms 2*, pp. 109-126.

- [Forrest, Javornik, Smith, and Perelson, 1993] Forrest, S., Javornik, B., Smith, R. E., and Perelson, A. S. (1993). Using Genetic Algorithms to Explore Pattern Recognition in the Immune System. *Evolutionary Computation*, 1(3): pp. 191-211.
- [Forrest, Perelson, Allen, and Cherukur, 1994] Forrest, S., Perelson, A. S., Allen, L. and Cherukur, R. (1994) Self-Nonself Discrimination in a Computer. *Proceedings of IEEE Symposium on Research in Security and Privacy*, pp. 202-212, Oakland, CA, 16-18.
- [Fraser, 1957] Fraser, A. (1957). Simulation of genetic systems by automatic digital computers I: introduction. *Australian Journal of Biological Science* 10, pp. 484-491.
- [Freeman and Herron, 1998] Freeman, S. and Herron, J. C. (1998). *Evolutionary Analysis*, Prentice Hall, 1998.
- [Gershwin, 1987] Gershwin, S. B. (1987). An Efficient Decomposition Algorithm for the Approximate Evaluation of Tandem Queues with Finite Storage Space and Blocking. *Operation Research*, 35: pp. 291-305.
- [Goldberg, 1989] Goldberg, D. E. (1989). *Genetic Algorithms in search, Optimization, and Machine Learning*. Reading, MA: Addison Wesley.
- [Goldberg and Richardson, 1987] Goldberg, D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41-49.
- [Goldberg and Segrest, 1987] Goldberg, D. E. and Segrest, D. (1987). Finite Markov chain analysis of genetic algorithms. *International Conference on Genetic Algorithms*, Volume 2, Morgan Kaufmann, pp. 1-8.
- [Goldberg and Deb, 1991] Goldberg, D. E. and Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundation of Genetic Algorithms*, Morgan Kaufmann, pp. 69-93.
- [Hager and Pardalos, 1998] Hager, W. W., and Pardalos, P. M. (1998). *Optimal Control: Theory, Algorithms and Applications*. Kluwer Academic Publishers.
- [Hillis, 1992] Hillis, W. D. (1992). Coevolving parasites improve simulated evolution as an optimization procedure. *Artificial Life II*, volume X of Santa Fe Institute Studies in the Sciences of Complexity, Addison-Wesley, pp. 313-324.
- [Hofmeyr and Forrest, 2000] Hofmeyr, S. A., and Forrest, S. (2000). Architecture for an artificial immune system. *Evolutionary Computation*, 8(4): pp. 443-473.
- [Holland, 1975] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press.

- [Holland, 1992] Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. Cambridge, MA: MIT Press.
- [Holland, 2000] Holland, J. H. (2000). Building blocks, cohort genetic algorithms, and hyperplane-defined functions. *Evolutionary Computation*, 8(4): pp. 373-391.
- [Holland, Holyoak, Nisbett, and Thagard, 1986] Holland, J. H., Holyoak, K., Nisbett, R., and Thagard, P. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: MIT Press.
- [Horn, Goldberg, and Deb, 1994] Horn, J., Goldberg, D. E., and Deb, K., (1994) Implicit niching in a learning classifier system: nature's way. *Evolutionary Computation*, 2(1): pp. 37-66.
- [Huang, 1997] Huang, C-F. (1997). *Properties of Invariant Measures of Markov Processes That Describe Serial Production Lines*. Qual II Report. Ann Arbor, MI: University of Michigan, Electrical Engineering and Computer Science.
- [ICEO, 1997] Second International Contest on Evolutionary Optimization, held in the IEEE-ICEC 97 conference. URL <http://iridia.ulb.ac.be/~aroli/ICEO/2ndICEO.html>.
- [Ishida and Mizessyn, 1992] Ishida, Y. and Mizessyn, F. (1992) Learning Algorithms on an Immune Network Model: Application to Sensor Diagnosis. *Proceedings of International Joint Conference on Neural Networks*, vI, pp. 33-38.
- [Ishida, 1993] Ishida, Y. (1993). An Immune Network Model and its Applications to Process Diagnosis. *Systems and Computers in Japan*, 24(6):38-45.
- [Jacobs, 1995] Jacobs, D. and Meerkov, S. M. (1995). A system-theoretic property of serial production lines: improvability, *International Journal of System Science*, 26, pp. 755-785.
- [Jerne, 1974] Jerne, N. K. (1974) Towards a network theory of the immune system. *Ann. Immunol. (Inst. Pasteur)*, 125C:373-389.
- [Koza, 1992] Koza, J. (1992). *Genetic Programming*. MIT Press.
- [KrishnaKumar, Satyadas, and Neidhoefer, 1995] KrishnaKumar, K., Satyadas, A., and Neidhoefer, J. (1995), An immune system framework for integrating computational intelligence paradigms with applications to adaptive control, *Computational Intelligence A Dynamic System Perspective*, (Eds.) M. Palaniswami, Y. Attikiouzel, R. J. Marks II, D. Fogel and T. Fukuda, IEEE Press, pp. 32-45.
- [KrishnaKumar and Neidhoefer ,1997] KrishnaKumar, K. and Neidhoefer, J. (1997), immunized neurocontrol, *Expert Systems with Applications*, 13(3), pp. 201-214.

- [KrishnaKumar and Neidhoefer ,1999] KrishnaKumar, K. and Neidhoefer, J. (1999), Immunized adaptive critics for an autonomous aircraft control application, *Artificial Immune Systems and Their Applications*, Springer-Verlag, pp. 221-241.
- [Levy, 1993] Levy, S. (1993). *Artificial Life — A Report from the Frontier Where Computers Meet Biology*. New York: Vintage Books.
- [Mahfoud, 1995] Mahfoud, S. W. (1995). *Niching Methods For Genetic Algorithms*. Ph. D. thesis, IlliGAL Report No. 95001. Urbana, IL: University of Illinois at Urbana-Champaign.
- [Mayr, 1942] Mayr, E. (1942). *Systematics and the Origin of Species*. (Reprinted 1982). New York: Columbia University Press.
- [McDonnell, Reynolds, and Fogel, 1995] McDonnell, J. R., Reynolds, R. G., and Fogel, D. B. (1995). *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, The MIT Press.
- [Michalewicz, 1994] Michalewicz Z. (1994) *Genetic Algorithms + Data Structures = Evolution Programs* (second, extended edition). Springer-Verlag.
- [Miller and Todd, 1993] Miller, G. F. and Todd, P. M. (1993). Evolutionary wanderlust: Sexual selection with directional mate preferences. *From animals to animats 2: Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pp. 21-30.
- [Miller, 1986] Miller, R. G. (1986). *Beyond ANOVA, basics of applied statistics*. John Wiley & Sons.
- [Mitchell, 1996] Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.
- [Mitchell, Forrest, and Holland, 1992] Mitchell, M., Forrest, S., & Holland, J. H. (1992). The royal road for genetic algorithms: fitness landscapes and GA performance. *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pp. 245-254.
- [Mühlenbein and Schlierkamp-Voosen, 1993] Mühlenbein, H. and Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithms I. Continuous Parameter Optimization. *Evolutionary Computation*, 1:25-49, 1993.
- [Nix and Vose, 1992] Nix, A. E. and Vose, M. D. (1992). Modelling genetic algorithms with Markov chains. *Annals of Mathematics and Artificial Intelligence* #5, pp. 79-88.

- [Ootsuki and Sekiguchi, 1999] Ootsuki, J. T. and Sekiguchi, T. (1999), Application of the Immune System Network Concept to Sequential Control, *Proc. IEEE SMC'99*, 3, pp. 869-874.
- [Percus, Percus, and Perelson, 1993] Percus, J. K., Percus, O. E., and Perelson, A. S. (1993). Predicting the size of the antibody-combining region from consideration of efficient self/nonself discrimination. *Proceedings of the National Academy of Science* 90, pp. 1691-1695.
- [Perelson, 1989] Perelson, A. S. (1989). Immune network theory. *Immunol. Rev.*, 110, pp. 5-36.
- [Potter, De Jong, and Grefenstette, 1995] Potter, M. A., De Jong, K. A., and Grefenstette, J. J. (1995). A coevolutionary approach to learning sequential decision rules. *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 366-372.
- [Price, 1996] Price, P. W. (1996). *Biological Evolution*. Saunders College Publishing, Harcourt Brace College Publishers, New York, NY.
- [Rechenberg, 1965] Rechenberg, I. (1965). Cybernetic Solution Path of an Experimental Problem. Ministry of Aviation. Royal Aircraft Establishment.
- [Rechenberg, 1973] Rechenberg, I. (1973). *Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart.
- [Riopka and Bock, 2000] Riopka, T. P. and Bock, P. (2000). Intelligent recombination using individual learning in a collective learning genetic algorithm. *Proc. of 2000 Genetic and Evolutionary Computation Conference (GECCO-2000)*, Morgan Kaufmann, pp. 104-111.
- [Rosin and Belew, 1997] Rosin, C. D. and Belew, R. K. (1997). New methods for competitive coevolution. *Evolutionary Computation*, 5(1): pp. 1-29.
- [Roughgarden, 1979] Roughgarden, J. (1979). *Theory of Population Genetics and Evolutionary Ecology*, Prentice-Hall, 1979.
- [Rudolph, 1994] Rudolph, G. (1994). Massively parallel simulated annealing and its relation to evolutionary algorithms. *Evolutionary Computation*, 1(4): pp. 361-383.
- [Russel, 1998] Russel, P. J. (1998). *Genetics*. Benjamin/Cummings, 1998.
- [Schaffer and Eshelman, 1991] Schaffer, J. D. and Eshelman, L. J. (1991). On crossover as an evolutionarily viable strategy. *Proc. 4th International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 61-68.

- [Schaffer and Morishima, 1987] Schaffer, J. D. and Morishima, A. (1987). An adaptive crossover distribution mechanism for genetic algorithms. , *Genetic Algorithms and Their Applications: Proceedings of the 2nd International Conference on Genetic Algorithms*, NJ: Lawrence Erlbaum, pp. 36-40.
- [Schaffer, Caruana, Eshelman, and Das, 1989] Schaffer, J. D., Caruana, R. A., Eshelman, L. J., and Das, R. (1989). A study of control parameters affecting online performance of genetic algorithms for function optimization. *Proc. 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 51-60.
- [Schwefel, 1981] Schwefel, H.-P. (1981). *Numerical Optimization of Computer Models*. John Wiley and Sons.
- [Schwefel, 1995] Schwefel, H.-P. (1995). *Evolution and Optimum Seeking*, John Wiley, Chichester, UK.
- [Sebald and Fogel, 1994] Sebald, A. V. and Fogel, L. J. (1994). *Proceedings of the Third Annual Conference on Evolutionary Programming*, San Diego, CA, World Scientific.
- [Sevastyanov, 1962] Sevastyanov, B. A. (1962). Influence of Storage Bin Capacity on the Average Standstill Time of a Production Line. *Theory Prob. Appl.*, 7, pp: 429-438.
- [Smith, Forrest, and Perelson, 1993] Smith, R., Forrest, S., and Perelson, A. S. (1993). Searching for diverse, cooperative populations with genetic algorithms. *Evolutionary Computation*, 1(2): pp. 127-149.
- [Spears, 1994] Spears, M. W. (1994). Simple subpopulation schemes, *Proc. of the 1994 Evolutionary Programming Conference*, World Scientific. pp. 296-307.
- [Spears, 1995] Spears, M. W. (1995). Adapting crossover in evolutionary algorithms, *Proc. 4th Annual Conference on Evolutionary Programming*, MIT Press. pp. 367-384.
- [Spears and De Jong, 1991a] Spears, M. W. and De Jong, K. A. (1991). On the virtues of parameterized uniform crossover, *Proc. 4th International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 230-236.
- [Spears and De Jong, 1991b] Spears, M. W. and De Jong, K. (1991) An analysis of multi-point crossover. *Foundation of Genetic Algorithms*, Morgan Kaufmann, pp. 301-315.
- [Stadnyk, 1987] Stadnyk, I. (1987). Schema recombination in pattern recognition problems. In *Proc. 2nd International Conference on Genetic Algorithms and their Applications*, Hillsdale, N.J. Lawrence Erlbaum Assoc.

- [Suzuki, 1993] Suzuki, J. (1993). A Markov chain analysis on a genetic algorithm. *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 146-153.
- [Syswerda, 1989] Syswerda, G. (1989). Uniform crossover in genetic algorithms, *Proc. 3rd Int'l Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 2-9.
- [Todd and Miller, 1991] Todd, P. and Miller, G. (1991). On the sympatric origin of species: mercurial mating in the quicksilver model. , *Proc. 4th International Conference on Genetic Algorithms*, Morgan Kaufmann, pp. 547-554.
- [Vafaie and De Jong, 1996] Vafaie, H. and De Jong, K. A. (1996). Genetic algorithms as a tool for restructuring feature space representations. *Proceedings of the International Conference on Tools with AI*, Los Alamitos, CA. Institute of Electrical and Electronics Engineers, IEEE Computer Society Press.
- [Vlachos, Williams and Gomm, 1999] Vlachos, C., Williams, D., and Gomm, J. B. (1999). Genetic approach to decentralized PI controller tuning for multivariable processes. *IEE Proc. Control Theory and Applications*, vol. 146, pp. 58-64.
- [Vose and Liepins, 1991] Vose, M. D. and Liepins, G. E. (1991). Punctuated equilibria in genetic search. *Complex Systems 5*, pp. 31-44.
- [Watson and Pollack, 2000] Watson, R. A. and Pollack, J. (2000). Recombination without respect: schema combination and disruption in genetic algorithm crossover. *Proc. of 2000 Genetic and Evolutionary Computation Conference (GECCO-2000)*, Morgan Kaufmann, pp. 112-119.
- [Whitley, Rana, Dzubera, and Mathias, 1996] Whitley, D., Rana, S., Dzubera, J., and Mathias, K. E. (1996). Evaluating evolutionary algorithms. *Artificial Intelligence*, 85(1-2):245-276.