Artificial evolution on fitness landscapes with neutral networks

Katada, Yoshiaki

(Degree)
博士（工学）

(Date of Degree)
2004-09-30

(Date of Publication)
2013-03-26

(Resource Type)
doctoral thesis

(Report Number)
甲3202

(URL)
https://hdl.handle.net/20.500.14094/D1003202

ARTIFICIAL EVOLUTION ON FITNESS LANDSCAPES WITH NEUTRAL NETWORKS

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL OF SCIENCE AND TECHNOLOGY
AND THE COMMITTEE ON GRADUATE STUDIES
OF KOBE UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF ENGINEERING

Yoshiaki Katada
August 2004

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Engineering.

Toshiharu Taura
(Principal Co-Advisor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Engineering.

Kazuhiro Ohkura
(Principal Co-Advisor)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Engineering.

Kanji Ueda

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Engineering.

Fumio Kojima

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Engineering.

Hidefumi Sawai

Approved for the University Committee on Graduate Studies.

Preface

Neutral networks, which occur in fitness landscapes containing neighboring points of equal fitness, have attracted much research interest in the evolutionary computation community in recent years. Neutral networks have been found in many real-world applications of artificial evolution, such as the evolution of neural network controllers in robotics and on-chip electronic circuit evolution. Although around ten years have passed since the importance of neutral networks was pointed out, the progress in this research field is stagnating due to the difficulties of applying the traditional theories of artificial evolution. Therefore, several researchers have been attempting to develop new theories of artificial evolution on fitness landscapes with neutral networks.

Following these research trends, this thesis has established some theoretical guidelines of artificial evolution on fitness landscapes with neutral networks. This thesis can be divided into foundations and applications; In foundations, evolutionary dynamics were investigated in test functions of fitness landscapes with neutral networks as well as efficient genetic algorithms for them were proposed. The measure of neutrality in fitness landscapes, which originates from population genetics, was also proposed. In applications, it was investigated whether the results obtained in the test functions are applicable to real-world problems, particularly evolution of artificial neural networks for robot control. It is confirmed that their fitness landscapes includes neutral networks and the evolutionary dynamics obtained in these problems are consistent with the dynamics in the test functions. Finally, some conclusions are given.

Acknowledgements

I would like to express my gratitude to all the people who helped me to accomplish this dissertation.

I am very grateful to Professor Kanji Ueda of The University of Tokyo for his instructive guidance and encouragement in my work.

I would like to send my special thanks to my supervisor Dr. Kazuhiro Ohkura of Kobe University for his valuable support in my research work and for his great credit in me.

I also would like to thank Professor Toshiharu Taura of Kobe University for his valuable help in completing my Ph.D. study, and to the members of the committee, Professor Fumio Kojima and Professor Hidefumi Sawai of Kobe University for their helps and constructive advices.

Many thanks goes to Mr. Yutaka Yasuda, Mr. Yuzo Hirotsu and Mr. Shinya Takasaki for their great contributions and helps to this work.

I am also much indebted to a number of people for their heartful encouragements: my senior students Dr. Kazuaki Yamada of The University of Tokyo, Dr. Yoshiyuki Matsumura of Shinshu University, Dr. Zlatan Car of Rijeka University, my junior student Mr. Toshiyuki Yasuda, and all the other members of the Research Group for Robotics and Collective Intelligence in Intelligent Machines and Manufacturing Systems Laboratory at Kobe University.

Finally, I owe an enormous thanks to my parents Toyozou and Masae for their unstinting supports, helps and advices in my life.

Yoshiaki Katada

August, 2004.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Open-ended Evolution

In the evolutionary algorithm (EA) community, evolutionary dynamics of genetic algorithms (GA) have been described based on the schema theory and the building block hypothesis (Holland, 1975; Goldberg, 1989); The utilization of the initial diversity in a population generated randomly can make a global search. Through the successive application of selection, recombination and mutation, the population gradually loses its variation and focuses its genetic search into a small region of high fitness, that is, shifts to local search. At last, the population converges completely to a certain point in the genotype space, which would be a good solution. This is an ideal end of artificial evolution considered in the EA community. Especially in problems that include so-called GA difficulty, GA researchers and practioners have been trying to make the convergence speed slow with keeping genetic materials gained before in order to have enough generations for finding appropriate building blocks without being stuck at local optima. Within this framework, crossover is a major genetic operator because it generates innovation by combining building blocks. On the other hand, mutation would be a local search operator.

While, when people shift the focus of attention to natural evolution, they suspect whether artificial evolution should be as the canonical GA, because natural evolution never ceases or converges. It does not seem to be stuck on local optima at all. How

can an evolving population avoid being trapped on them? From this point of view, there are recently becoming novel research trends where it is considered that artificial evolution should be open-ended as natural evolution is (Harvey, 1997; Barnett, 1997; Nolfi and Floreano, 2000). It might be claimed that open-ended evolution is caused by several factors; The first candidate would be environmental change. If the environment changes and the individual is not selected according to the predefined environment, the trail of evolution is dictated by environmental changes and its adaptations. The second one would be coevolution. In the case of competitive coevolution, competing individuals interact each other then the environment is not static but dynamic for an individual due to the opposite one. Consequently, a population does not cling to a certain point in the genotype space due to their dynamics. However, it would be passive or not efficient for an evolving population to wait an innovation until environmental change occurs. In addition, it is likely that artificial evolution might be initialized by environmental changes and coevolution due to that the genetic materials obtained before would not be reusable in new environments. Thus, another possibilities must be considered. One of them would be neutral evolution. This will be discussed by considered as a necessary condition for open-ended evolution in the remainder of this thesis.

1.2 Landscape Structure and Problem Difficulties

The *fitness landscape*, introduced by Wright (1932)[1], is described as the multidimensional space which is composed of the set of several dimensions per genotype locus and an additional dimension, or "height" of the landscape, for the fitness value of the particular genotype. Each point on it corresponds to a single genotype and the fitness value. Thus, a genetic operation means a movement from an original point to neighbors of it, and selection means the fixation or elimination of the points. The process of evolution is described as the transition of points on a fitness landscape.

In early works of the theoretical GA community, problem difficulties for a GA

[1]Wright viewed evolution as optimization process on a fitness landscape.

have been discussed in terms of the geography of a fitness landscape: *isolation*, *deception* and *multimodality*. Their factors affecting the performance of the GA to solve optimization problems are still contentious issues. However, several counterexamples on the latter two factors were found, showing that they are neither necessary nor sufficient to make a problem difficult (Jones and Forrest, 1995; Horn and Goldberg, 1995; Kallel, 1998).

Another attempt to characterize difficulty has been done by measuring the feature of a fitness landscape, *epistasis* or *ruggedness*. Most of the works in this area are based on the fitness correlation between the original individual and its mutant to describe a fitness landscape (Kauffman, 1993; Weinberger, 1990; Manderick et al., 1991; Hordijk, 1994; Jones and Forrest, 1995; Stadler, 1996). Therefore, they are derived from the average of fitness correlations between parents and offspring or the fitness distance autocorrelation function obtained by using a random walk. In the GA community, the majority of fitness landscape descriptions have based on ruggedness since then. However, the existence of problem domains has been reported where the traditional theories based on the schema theory and the building block hypothesis does not hold (Mitchell et al., 1992; van Nimwegen et al., 1999). Those kinds of problem mainly show equilibrium period, *neutral evolution* in their evolutionary dynamics. If equilibrium periods seem dominative in the process of evolution, ruggedness is not enough to measure the search difficulty.

1.3 Redundant Representations and Neutrality

Selective neutrality has been found in many real-world applications of artificial evolution, such as the evolution of neural network controllers in robotics (Harvey, 1997; Smith et al., 2001b,a, 2002b) and on-chip electronic circuit evolution (Thompson, 1996; Vassilev et al., 2000; Vassilev and Miller, 2000). This characteristic is caused by highly redundant mappings from genotype to phenotype or phenotype to fitness. With these kinds of problems, redundancy is inevitable although it is customary among GA practioners deliberately to avoid redundancy in the genetic coding of artificial evolution problems. Even for problems where redundancy is largely absent,

however, it may be useful to introduce it. A number of researchers have been trying to improve the performance of artificial evolution on more traditional problems by incorporating redundancy in genotype to phenotype mappings(Ohkura and Ueda, 1999; Knowles and Watson, 2002; Shipman et al., 2000; Shackleton et al., 2000; Ebner et al., 2001; Rothlauf and Goldberg, 2003).

Some formal definitions about redundant mappings are given according to the notation in (Rothlauf and Goldberg, 2003) in the following; All fitness landscapes in this thesis are based on fixed length binary bit-string genotypes with the length, N. Φ_g is defined as the set of genotypes, the genotypic search space where the genetic operations are applied. Thus, the size of the genotypic search space is $|\Phi_g| = 2^N$. Φ_p is defined as the phenotypic search space. The size of the phenotypic search space is $|\Phi_p|$. A mapping $f_g : \Phi_g \rightarrow \Phi_p$ determines which genotypes $x^g \in \Phi_g$ are assigned to which phenotypes $x^p \in \Phi_p$. A mapping f_g is called redundant if f_g is surjective[2] and not injective. Then, there exists $\Phi'_p (\neq \phi)$ s.t.

$$\Phi'_p = \{x^p | \exists x^g \exists y^g \in \Phi_g s.t. x^p = f_g(x^g), x^p = f_g(y^g), x^g \neq y^g\}. \tag{1.1}$$

Ohkura and Ueda (1999) designed the mapping motivated by the genetic mechanisms of *operon* found in microbial organisms: A genotype includes a large amount of redundancy with respect to a phenotype. In addition, any locus is always included in a certain cluster called an operon to make a hierarchal structure of two layers, i.e., an operon layer and a locus layer. For the redundant genotype and hierarchal structure, new genetic operators were developed (see the details in Appendix B). Shackleton et al. (2000), Shipman et al. (2000) and Ebner et al. (2001) proposed to use the cellular automaton mapping, the boolean network mapping and the trivial voting mapping for designing redundant mappings from genotype to phenotype.

Rothlauf and Goldberg (2003) proposed the necessary condition for efficient evolutionary search which a redundant mapping should satisfies by introducing the concept, "synonymity." A mapping f_g is called a "synonymously redundant" if the genotypic distances between all x^g which represent the same phenotype x^p are small for all

[2]It can be assumed that each phenotype x^p is assigned to at least one genotype x^g.

different x^p. Then, there exists $\Phi'_{p,syn}$ s.t.

$$\Phi'_{p,syn} = \{x^p | \ \exists x^g \exists y^g \in \Phi_g s.t. x^p = f_g(x^g), x^p = f_g(y^g), \\ y^g = E(x^g), \ h(x^g, y^g) \leq d, d \neq 0\}, \tag{1.2}$$

where E is a genetic operator, h is Hamming distance, precisely, $h(x^g, y^g) = \sum_{i=1}^{N} \delta(x^g, y^g)$ and d should be reasonably small. Here, a genotype y^g is called, "a (d-bit) neighbour to the genotype x^g or a (d-bit) mutation of x^g." Note that the genetic operators define the neighborhood in landscapes. Rothlauf and Goldberg (2003) classified the cellular automaton mapping and the boolean network mapping as non-synonymously redundant mappings and the trivial voting mapping as a synonymously redundant mapping.

These definitions can also be extended to redundant mappings from phenotype to fitness. Let Φ_F be defined as the set of fitness values, the fitness search space where the genotype x^g is evaluated. The size of the fitness search space is $|\Phi_F|$. A mapping $f_p : \Phi_p \rightarrow \Phi_F$ determines which phenotypes $x^p \in \Phi_p$ are assigned to which fitness values $x^F \in \Phi_F$. For the explanation, it is assumed that f_g is bijective, that is, there is no redundant mapping from genotype to phenotype. Here, a mapping f_p is called redundant if f_p is surjective and not injective. Then, there exists $\Phi'_F (\neq \phi)$ s.t.

$$\Phi'_F = \{x^F | \ \exists x^g \exists y^g \in \Phi_g s.t. x^F = (f_p \circ f_g)(x^g), x^F = (f_p \circ f_g)(y^g), x^g \neq y^g\}. \tag{1.3}$$

Neutrality is also found in natural systems. In molecular biology, it is clear that there is often a high degree of redundancy in the coding from genotype to phenotype. Thus, there are redundancy on several levels; e.g. many nucleotide sequences may code for the same amino acid, while many amino acid sequences may code for functionally equivalent proteins. Therefore, neutrality has been of particular interest to evolutionary theorists (Kimura, 1983; Ohta, 1992) and molecular biologists (Schuster et al., 1994; Forst et al., 1995; Gruner et al., 1996; Huynen et al., 1996; Reidys et al., 1997; Babajide et al., 1997).

The concept of neutrality in artificial evolution originates from Kimura's neutral

theory in population genetics. Kimura (1983) developed the neutral theory in order to give the explanation for the various observations in the amino acid substitution rate. Ohta (Ohta, 1992, 1998) extended the neutral theory under selective constraints as the nearly neutral theory. According to Kimura's neutral theory and Ohta's nearly neutral theory, the following five assertions have been made (Kimura and Ohta, 1974):

1. For each protein, the rate of evolution in terms of amino acid substitutions is approximately constant per year per site for various lines, as long as the function and tertiary structure of the molecule remain essentially unaltered.

2. Functionally less important molecules or parts of a molecule evolve (in terms of mutant substitutions) faster than more important ones.

3. Those mutant substitutions that disrupt less the existing structure and function of a molecule (conservative substitutions) occur more frequently in evolution than more disruptive ones.

4. Gene duplication must always precede the emergence of a gene having a new function.

5. Selective elimination of definitely deleterious mutants and random fixation of selectively neutral or very slightly deleterious mutants occur far more frequently in evolution than positive Darwinian selection of definitely advantageous mutants.

1.4 Neutral Networks

Fitness landscapes which include neutrality have been conceptualized as containing *neutral networks* (Harvey and Thompson, 1996; Smith et al., 2002a; Ebner et al., 2001). This concept is central to the majority of research in this field. Harvey (1996) first introduced the concept of neutral networks into the GA community. His definition is as follows:

> A neutral network of a fitness landscape is defined as a set of connected points of equivalent fitness, each representing a separate genotype: here connected means that there exists a path of single (neutral) mutations which can traverse the network between any two points on it without affecting fitness.

The word, "connected" in this definition corresponds to a synonymously redundant mapping from genotype to phenotype. Therefore, a neutral network of a fitness landscape is defined more formally by extending Equation 1.2 to redundant mappings from phenotype to fitness as follows:

$$\Phi_F^* = \{x^F |\ \exists x^g \exists y^g \in \Phi_g s.t. x^F = (f_p \circ f_g)(x^g), x^F = (f_p \circ f_g)(y^g), \\ y^g = E(x^g), h(x^g, y^g) \leq d, d \neq 0\}, \tag{1.4}$$

where d should be reasonably small. Here, a genotype y^g is called a (d-bit) mutation of x^g. Also note that the genetic operators, E, define the neighborhood in the landscapes. This relationship partitions the fitness space Φ_F, as a result, the genotypic space Φ_g. Barnett (1998) defined Φ'_F (Equation 1.3) as *neutral sets*. Thus, the neutral networks are the connected components of the neutral sets, $\Phi_F^* \subseteq \Phi'_F$.

The evolutionary dynamics on neutral networks is similar to *punctuated equilibria* described by Gould and Eldredge (1977). That is, it can be classified into transient periods and equilibrium periods (Fig. 1.1) (van Nimwegen et al., 1999; Barnett, 2001). During an equilibrium period, the population is clustered in genotype space around the *dominant phenotype*, analogously to *quasi-species* (Eigen et al., 1989), and moves around until it finds a portal to a neutral network of higher fitness. The discovery of a portal leads to a transient period, which is expected to be very short in comparison to an equilibrium period[3].

The evolutionary dynamics on neutral networks is clearly different from the traditional explanations based on the schema theory. First, the convergence of a population

[3]These concepts originate from molecular evolution (Schuster et al., 1994; Forst et al., 1995; Gruner et al., 1996; Huynen et al., 1996; Reidys et al., 1997; Babajide et al., 1997).

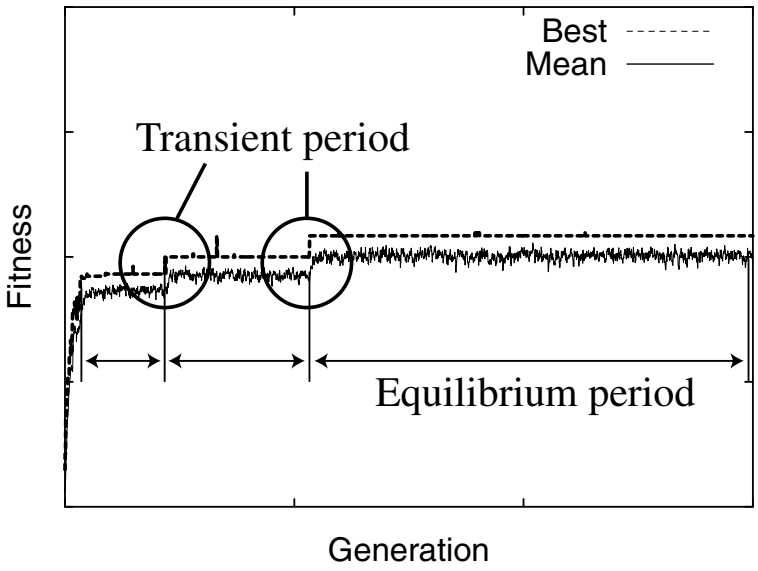


Figure 1.1: Typical evolutionary dynamics on a fitness landscape featuring neutral networks, which can be classified into transient periods and equilibrium periods: Instead of being stuck in a local optimum, a population is exploring in the genotype space during the equilibrium period.

imply not an end of artificial evolution but a formation of a cluster, that is, a beginning of open-ended evolution (Harvey, 1997). Secondly, the building block hypothesis does not hold (Mitchell et al., 1992) and crossover does not make sense in equilibrium periods (van Nimwegen et al., 1999) in the *Royal Road Functions*, which were designed to demonstrate the building block hypothesis. This is because Royal Road Functions includes neutral networks (van Nimwegen et al., 1999). Thus, in fitness landscapes with neutral networks, crossover would not be a major genetic operator even if it is useful as a complementary operator. Mutation should be considered as a major genetic operator in such fitness landscapes.

1.5 Problems and Approaches

It has been shown that there is a clear transition in evolutionary dynamics for populations on neutral networks over the mutation rate range. At a very low mutation

rate, the population is maintained in a cluster on the neutral network. As the mutation rate increases, the population gradually loses the current network. That is, some individuals fall to lower neutral networks. At a certain critical mutation rate, the whole population will lose the current neutral network. Then the search will become random. This mutation rate is called the *phenotypic error threshold* (Huynen et al., 1996; Reidys et al., 1997; Barnett, 1997; van Nimwegen and Crutchfield, 1998; van Nimwegen et al., 1999).

Generally, the error threshold sets the upper limit for a mutation rate that will enable efficient search. This implies that if a constant mutation rate strategy is adopted, a relatively low mutation rate should be set so as to avoid any error threshold effects during the process. From a practical point of view, however, it would be efficient to shorten the equilibrium periods which dominate the whole computation (Fig. 1.1).

Additionally, in landscapes which include ruggedness, individuals can easily get trapped on local optima if there is a low mutation rate and high selection pressure. Since the concept of neutrality or neutral networks was introduced into the EA community, EA researchers have expected that increasing neutrality makes the surface of rugged landscape smoother to prevent a population from getting trapped on local optima. However, it has been demonstrated in tunably neutral NK landscapes(Barnett, 1997; Newman and Engelhardt, 1998) that increasing neutrality does not affect the ruggedness, although it does reduce the number of local optima (Barnett, 1997; Newman and Engelhardt, 1998; Smith et al., 2002a). This means that the effects of ruggedness must be taken into account even if landscapes include neutral networks. Using a high mutation rate can shorten equilibrium periods and help a population avoid becoming trapped on local optima. However, as noted above, using a high mutation rate can be counterproductive because of the effects of error thresholds. One approach to overcoming these problems would be to adopt *variable mutation rate strategies*, which change the effective mutation rate adaptively during the process of evolution.

In this thesis, a standard GA[4] (Goldberg, 1989), which employs a constant mutation rate, and the operon-GA (Ohkura and Ueda, 1999), which can change its effective mutation rate are employed. The influences of GA parameters, especially, the mutation rate and selection pressure on the speed of population movement on very simple test functions with different levels of neutrality are investigated. The performance of both GAs are also examined in more complex test functions with different levels of ruggedness.

One of another research directions to advance in this field would be establishing the measure of neutrality of fitness landscapes as well as ruggedness. The above discussions demonstrate that ruggedness alone might be inadequate as a principal feature of a fitness landscape, especially with neutrality. However, the majority of such descriptions has focused solely on ruggedness of landscapes. Neutrality in landscapes has not been taken into account. If equilibrium periods seem dominative in the process of evolution, ruggedness is not enough to measure the search difficulty. Another measure, i.e., neutrality is required.

To the best of my knowledge, statistical measurements with respect to neutrality are found only in the references (Vassilev et al., 2000; Smith et al., 2002a). Vassilev et al. (2000) studied the structure of on-chip electronic circuit evolution landscapes. They proposed the *information analysis* of fitness landscapes, which is defined over a time series obtained by a walk on a landscape. Their entropic measure of the time series makes it possible to confirm the existence of neutrality[5] in a landscape and which feature, neutrality or ruggedness is dominant in the landscape. Smith et al. (2002a) proposed a method for measuring neutrality of a landscape as one of the *fitness evolvability portraits*. This is defined as the probability that an offspring fitness is equal to the parent fitness. In their measurement, a certain distinctive difference, ε, between two fitness values considered to be neutral must be set by GA practioners, that is, $|(f_p \circ f_g)(x^g) - (f_p \circ f_g)(y^g)| \leq \varepsilon$. It has been reported by Vassilev et al. (2000) that Smith's measure is very sensitive to the value, ε. A great influence of the value on the measure of neutrality would be predicted especially in the case

[4]The details are described briefly in Appendix A.

[5]Vassilev et al. (2000) call it as plateaus.

that the fitness is evaluated as a real value or in a noisy environment. For these problems, Smith proposed the use of *neutral fitness band* (Smith et al., 2001b) or the significance level for the Student t-test (Smith, 2002) as the value. However, he declared that no significant difference between two fitness landscapes was detected by introducing such statistical neutrality in his thesis (Smith, 2002). This implies the difficulty to use fitness data for measuring neutrality. Therefore, this thesis focuses on the genotype data for measuring it.

According to the assertions (Kimura and Ohta, 1974) on the neutral theory (1983) and the nearly neutral theory (Ohta, 1992, 1998) mentioned earlier in this chapter, the number of gene substitutions of each genotype increases with the increase of neutrality. The number of gene substitutions is a scale of genetic distance, which shows the degree of difference between populations or species. From the viewpoint of measuring neutrality, a genetic distance could be an index of neutrality. In population genetics, there are several statistical methods for measuring the genetic distance between populations or species because population geneticists have been trying to explain evolutionary change quantitatively, that is, the change of gene frequency in the population. This idea would be beneficial in artificial evolution because there is no need to treat any fitness values. Thus, this thesis proposes the use of the *Nei's standard genetic distance* (Nei, 1972), which is one of such statistical methods for measuring genetic distances, for estimating the degree of neutrality in fitness landscapes.

1.6 Thesis Overview

This thesis can be divided roughly into two parts: foundations (Chapter 2 and Chapter 3) and applications (Chapter 4).

Chapter 1 introduced the theoretical background and methodology necessary to read this thesis.

Chapter 2 investigates the evolutionary dynamics in simple test functions and complex test functions for obtaining some guidelines for tuning the performance of GAs. Two types of GA are used. One is the standard GA, where the mutation rate is constant, and the other is the operon-GA, whose effective mutation rate at each

locus changes independently according to the history of genetic search. Section 2.2 investigates the effect of selection pressure and the variable mutation rate strategy on the evolutionary dynamics on an abstract model of very simple neutral networks, called the *Balance Beam Function*. Then, Section 2.3 examines the evolutionary dynamics on a more complex test function, called the terraced NK landscape, which incorporates both neutrality and ruggedness. Section 2.3.3 discusses the effect of error thresholds and the variable mutation rate strategy on smooth and rugged landscapes.

Chapter 3 introduces the second main direction of this thesis, describing fitness landscapes with neutral networks. The use of the Nei's standard genetic distance is proposed for estimating the degree of neutrality in fitness landscapes. Section 3.2 describes the Nei's standard genetic distance and modifies it in order to apply it to GAs. Section 3.3 investigates the characteristics of the Nei's genetic distance in a tunably neutral landscape to validate the use of the Nei's standard genetic distance for estimating the degree of neutrality in fitness landscapes. Section 3.4 examines the measure of ruggedness as well as the genetic distance, considering real-world problems. Section 3.5 discusses the error threshold on the population size and the mutation rate based on the obtained results.

Chapter 4 explores how well the obtained results in Chapter 2 and Chapter 3 apply to several real-world problems, particularly evolution of artificial neural networks for robot control. Section 4.2 and Section 4.3 investigate simulated real-world problems, a discrimination problem and a pursuit problem. Section 4.4 analyses a single evolutionary run for each GA in a real-world problem, a goal reach problem. In these problems, the Nei's genetic distance is applied to estimate and compare the neutrality of the fitness landscape among the neural network controllers. Then, based on these features of the fitness landscape, both GAs are applied for examining whether the evolutionary dynamics obtained in these problems are consistent with the dynamics in the test functions. Finally, the best evolved controllers of the neural networks are evaluated in both the simulated and the real environment.

Conclusions are given in the last chapter.

Chapter 2

Foundation: Neutral Evolution

2.1 Introduction

Considering that equilibrium periods dominate the whole computation in the evolutionary dynamics on neutral networks, A hypothesis is generated that tuning GAs for maximizing the speed of a population's movement on a neutral network is very effective for accelerating the evolution. Based on this, computer simulations are conducted using simple neutral networks, called the Balance Beam Function in Section 2.2. Additionally, it is interesting to investigate whether the observations on the Balance Beam Function are consistent with more complex problems with neutrality and ruggedness. This is because this research effort aims at solving complex real-world problems. However, because of their complexity, investigating such problems is unlikely to lead to generalizable conclusions. Therefore, Section 2.3 investigates the effect of ruggedness by employing a more complex and popular test function, the terraced NK landscape (Newman and Engelhardt, 1998).

2.2 Simple Test Function

2.2.1 Balance Beam Function

The Balance Beam Function, proposed by Yasuda *et al.* (2001), is employed as the test function. It is relatively easy to investigate evolutionary dynamics with this function, because it has a landscape composed of only two neutral networks, each of a different fitness—one of which has the shape of a "balance beam". The BBF is formulated as follows:

$$W = \begin{cases} 1.0 & if(A \leq \sum_{i=1}^{N} s_i \leq B) \\ \varepsilon \ \ (\ll 1) & otherwise, \end{cases} \tag{2.1}$$

where W, N, $s_i \in \{0, 1\}$ and ε are the fitness value, the length of the genotype, the value of the i-th locus and the positive real number, respectively. A and B $(0 < A < B \leq N)$ are constant integers. The abstract width, w, of a neutral pathway is calculated as $B - A$. Based on the hypothesis mentioned in the beginning of this chapter, this experiment focuses on how to maximize the moving speed of individuals on the neutral network with $W = 1.0$.

As the initial setting, all individuals in the population are placed on a particular point[1] on the higher neutral network, for instance, $s_i = 1, i = 1, \ldots, d$ and $s_i = 0, i = d+1, \ldots, l$, where d is a constant integer between A and B. This would be a situation similar to a population just after a transition period. In the process of evolution, each individual moves along the pathway by flipping the values of loci. Distance is defined as the number of 0s between the 1st locus and the d-th locus if the individual is on the higher neutral network. Distance is not defined for an individual on the lower neutral network. Therefore, the moving speed of a population can be measured as the number of generations (for generational GAs), or the number of evaluations (for steady state GAs) for one of individuals to reach the furthest distance, d. For the BBF, the optimal mutation rate, q_o, is identified as the mutation rate which results in the fastest speed over the mutation rate range. The phenotypic error threshold, q_{err},

[1]The similar initial setting is found in (Ochoa et al., 1999)

is identified as the mutation rate at which no individual reaches the furthest distance by the final generation.

2.2.2 Computer Simulations

Simulation Conditions

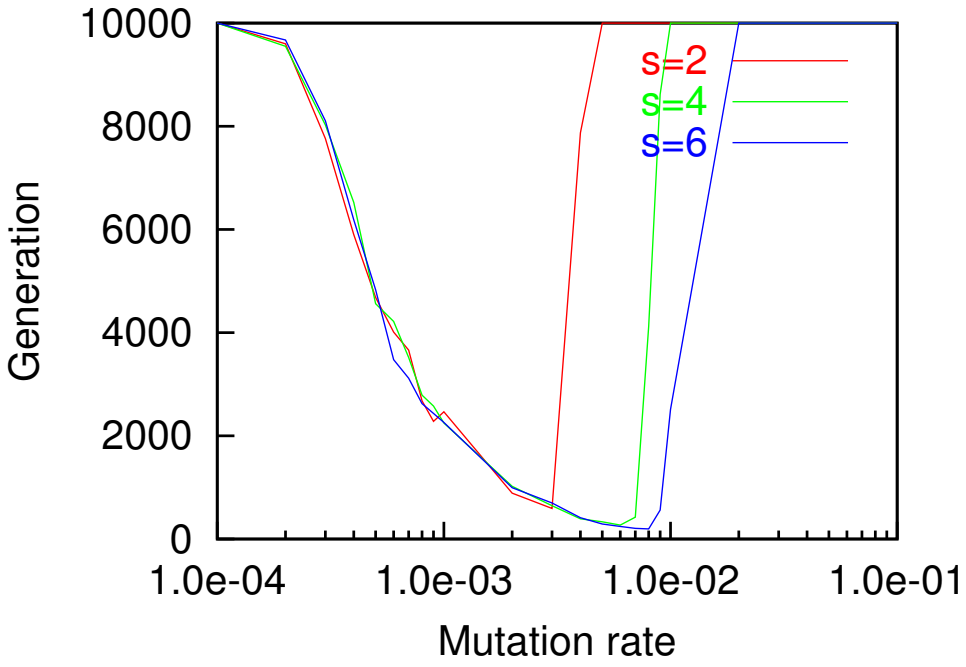
Computer simulations were conducted by setting the population size to 50 and the length of the genotype to 200. The standard GA (SGA) and the extended GA, called the operon-GA (OGA)(Ohkura and Ueda, 1999) were employed to evolve a population. The genetic operation for the SGA was standard bit mutation. For both GAs, the per-bit mutation rate, q, was set between 0.0001 and 0.1 with step sizes of $\Delta q = 0.0001$ (if $q < 0.001$), 0.001 (if $0.001 \leq q \leq 0.01$) and 0.01 (if $0.01 < q \leq 0.1$). Crossover was not used for either GA, following Nimwegen's suggestion (van Nimwegen et al., 1999). The OGA uses not only standard bit mutation but also five additional genetic operators: *connection*, *division*, *duplication*, *deletion* and *inversion* (The details are described briefly in Appendix B). The probabilities for genetic operations were set at 0.3 for *connection* and *division*, 0.1 for *duplication* and 0.05 for *deletion* and *inversion*, as recommended by Ohkura and Ueda (Ohkura and Ueda, 1999). The length of the value list in a locus was 6. Tournament selection was adopted. *Elitism*[2] was optionally applied. The tournament size was set at $s = 2, 4$ and 6. A generational GA was used. Each run lasted 10,000 generations[3]. 50 independent runs were conducted for each problem with the parameters $(A, B) = (19, 20), (19, 24), (19, 29)$ and $(19, 39)$; that is, with neutral pathways of width $w = 1, 5, 10$ and 20. d was set at 19. All results were averaged over 50 runs.

Simulation Results

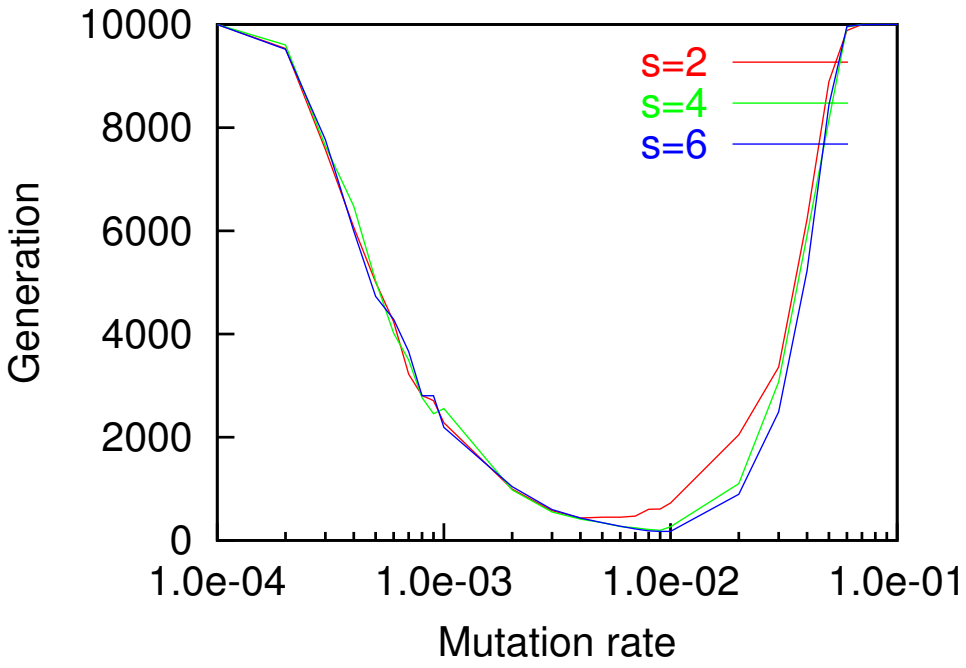
Fig. 2.1 and 2.2 show the average generations to reach the furthest distance, d, for $s = \{2, 4, 6\}$ and $w = 1$, with and without elitism, for the SGA and the OGA,

[2]An individual, randomly selected from those individuals whose fitness value is 1.0, is passed unmutated to the next generation.

[3]If no individual reaches the furthest distance by the final generation, the number of generations to reach the furthest distance is taken to be $10,000$.

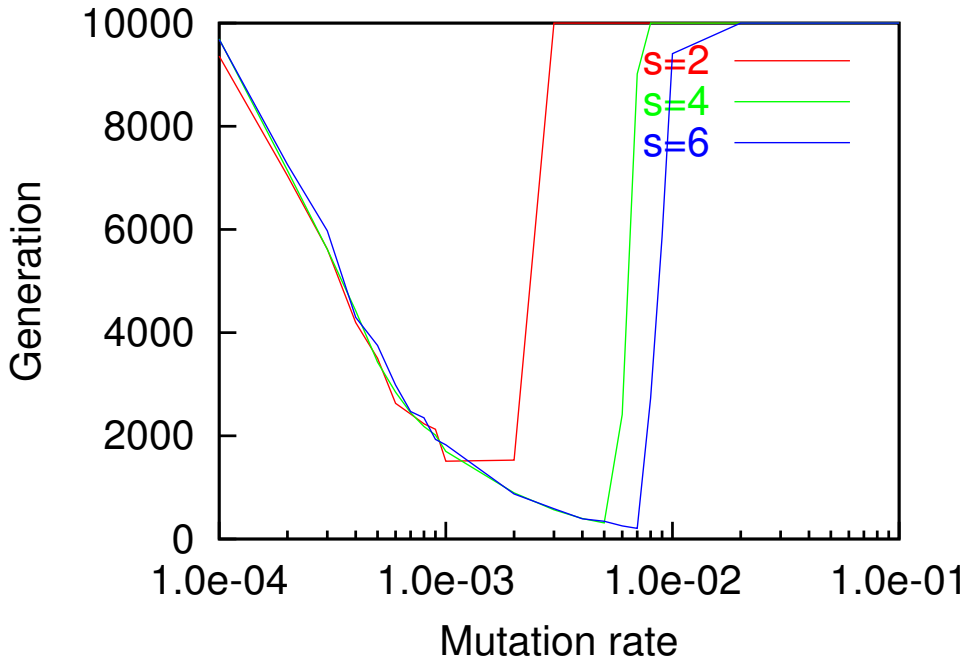


(a) without elitism

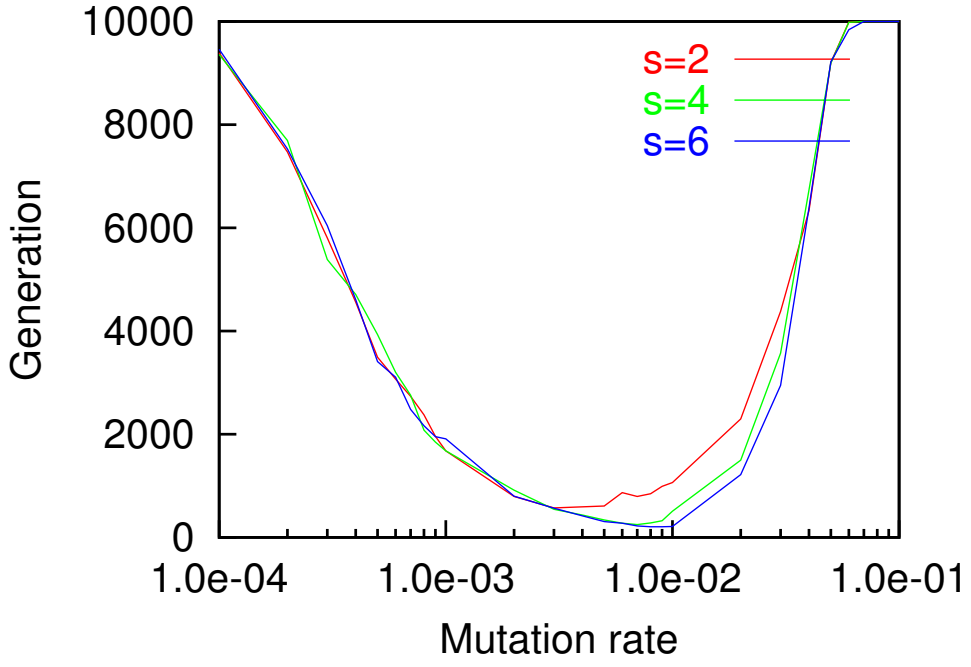


(b) elitism

Figure 2.1: Average generations to reach the furthest distance in 50 runs for the SGA with $w = 1$



(a) without elitism



(b) elitism

Figure 2.2: Average generations to reach the furthest distance in 50 runs for the OGA with $w = 1$

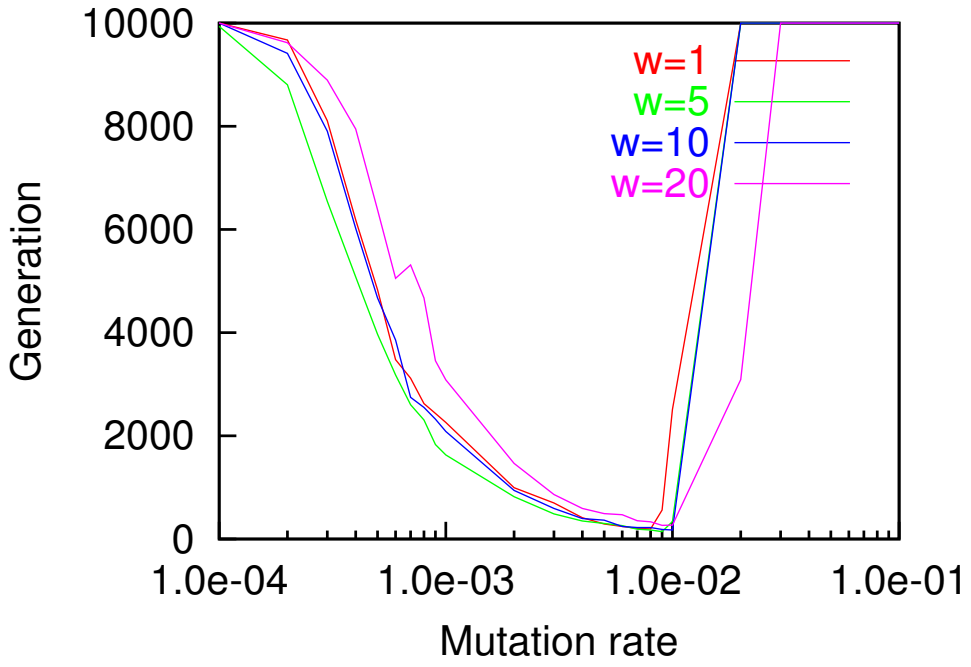
respectively. The curves are U-shaped as a function of the mutation rate, q, and have an optimal mutation rate and an error threshold.

Without elitism, the speed falls sharply when q exceeds q_o. Moreover, the optimal mutation rate is just below the error threshold (Fig. 2.1(a), 2.3(a) and Fig. 2.2(a), 2.4(a)). With elitism, both the optimal mutation rate and the error threshold increase, and so does the distance between them (Fig. 2.1(b), 2.3(b) and Fig. 2.2(b), 2.4(b)).

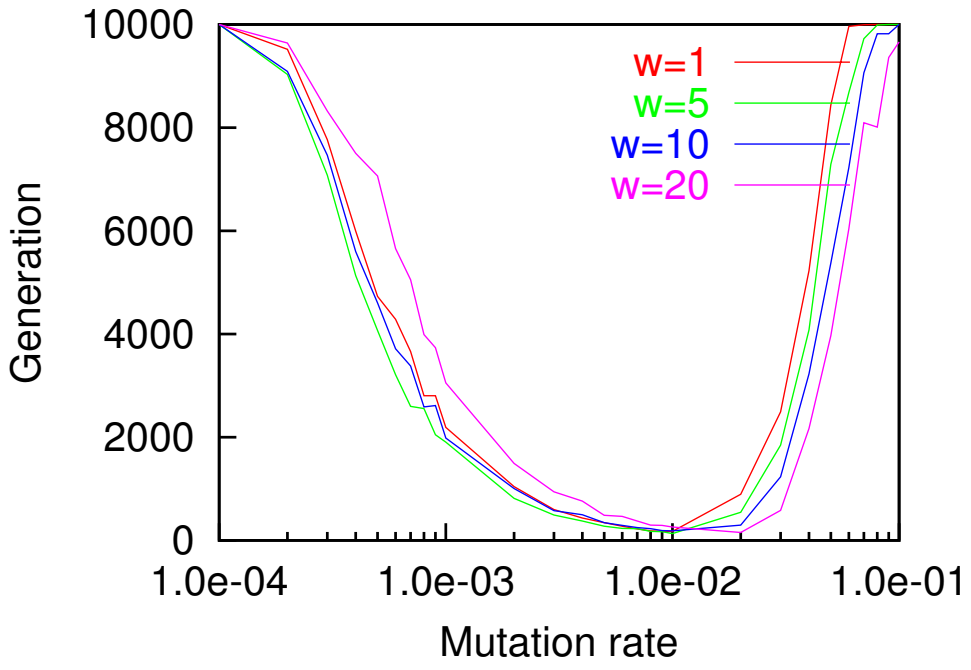
Increasing the tournament size increases the optimal mutation rate and the error threshold. It also results in improved speeds over the mutation rate range $q_o < q < q_{err}$ (Fig. 2.1(a), 2.1(b) and Fig. 2.2(a), 2.2(b)). This had the same effect for all the widths (Appendix C).

Fig. 2.3 and 2.4 show the average generations to reach the furthest distance, d, for $s = 6$ and $w = \{1, 5, 10, 20\}$, with and without elitism, for the SGA and the OGA, respectively. Increasing the width of the neutral pathway improves speeds over the mutation rate range $q_o < q < q_{err}$, and $q < q_o$ except $w = 10$ and 20 (Fig. 2.3(a), 2.3(b) and Fig. 2.4(a), 2.4(b)). This had the same effect for all the tournament sizes (Appendix C). This may be considered as follows; With the larger width, individuals move more freely on a neutral network. If the width exceeds a certain threshold, it takes more time for individuals to reach the furthest distance.

Fig. 2.5 shows the average generations to reach the furthest distance for the SGA and the OGA. The SGA produced higher speeds than the OGA when q was larger than q_o. However, the OGA produced higher speeds than the SGA when q was below q_o. This would be due to the effect of the variable mutation rate strategy.

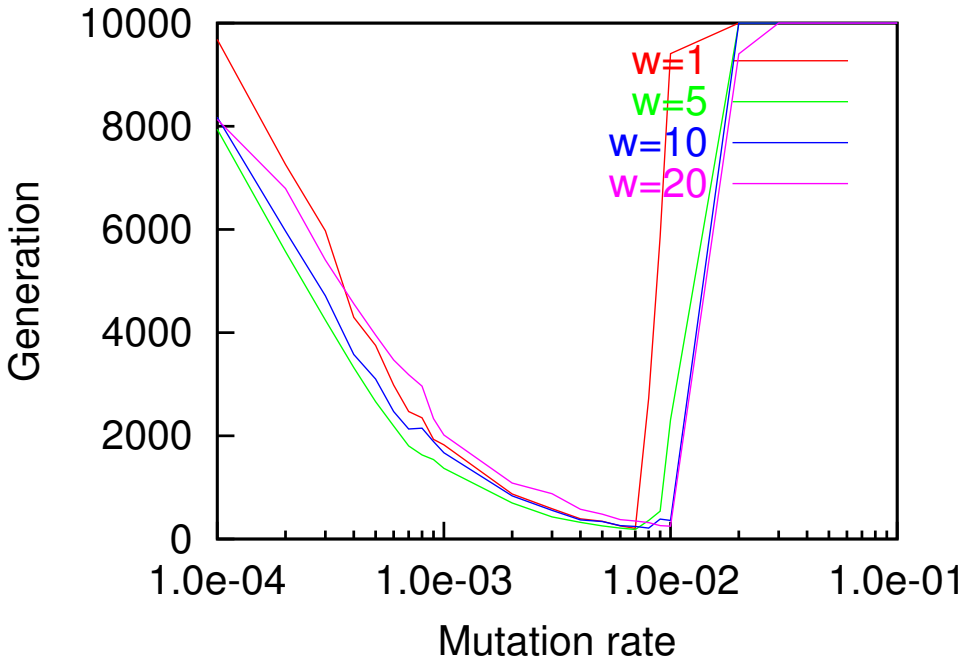


(a) without elitism

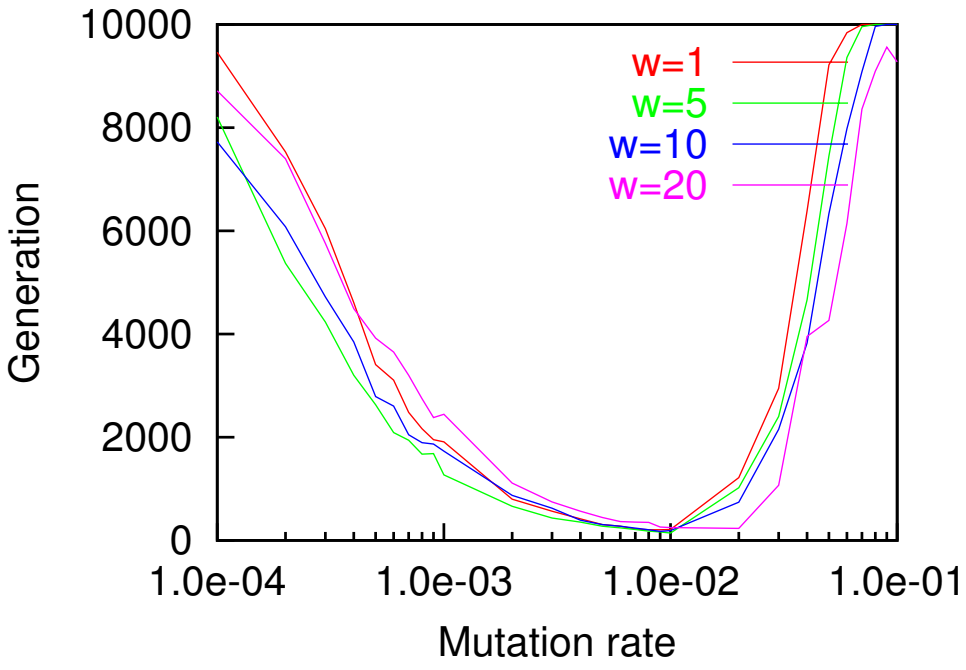


(b) elitism

Figure 2.3: Average generations to reach the furthest distance in 50 runs for the SGA for $s = 6$

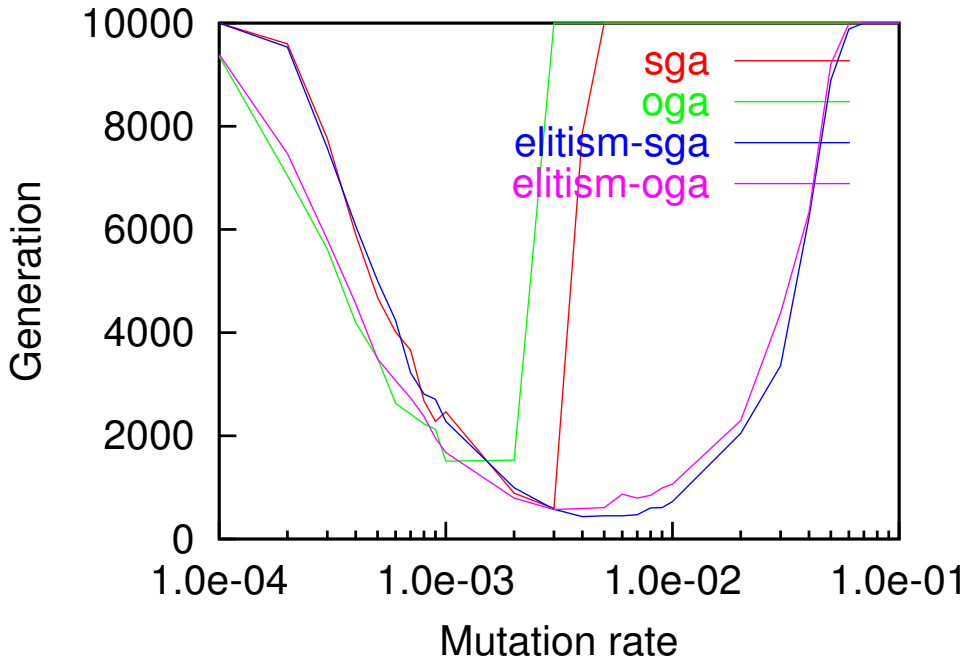


(a) without elitism

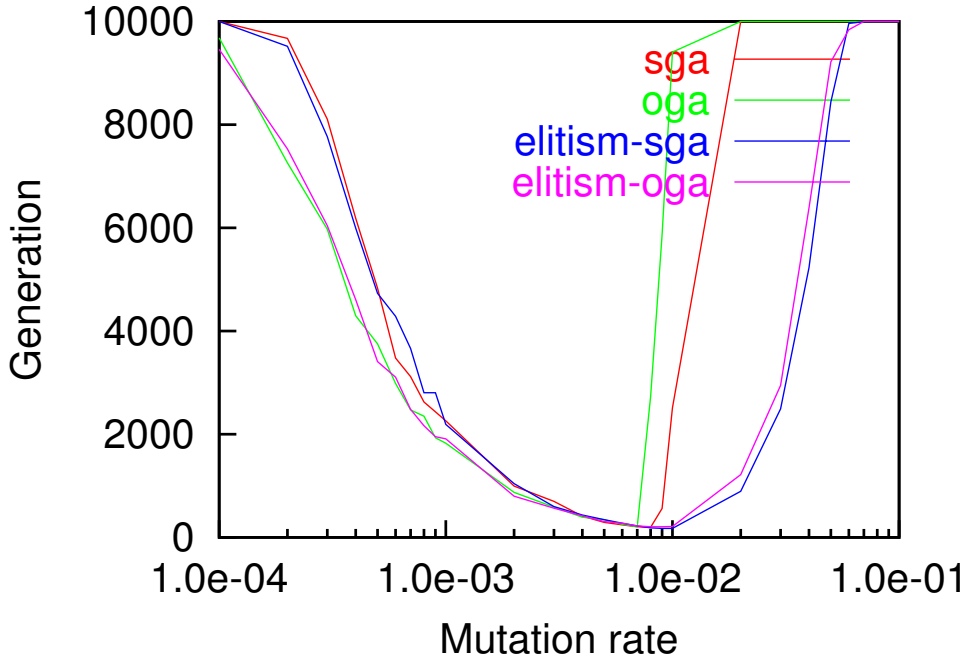


(b) elitism

Figure 2.4: Average generations to reach the furthest distance in 50 runs for the OGA for $s = 6$



(a) $s = 2$



(b) $s = 6$

Figure 2.5: Average generations to reach the furthest distance in 50 runs for the SGA and the OGA ($w = 1$)

2.2.3 Error Threshold and Selection Pressure

The results obtained in the previous subsection suggest that optimal mutation rates and error thresholds are strongly correlated with selection pressure. Without elitism, the optimal mutation rate will be just below the error threshold. If the mutation rate is set by estimating the optimal mutation rate, a misjudgement could result in the error threshold being exceeded, and thus lead to poor performance. With higher selection pressure, the optimal mutation rate and error threshold both increase. Elitism improves the moving speed when q is larger than q_o. Note that extreme elitism (e.g. 50% of offspring are identical to their parents) will reduce the moving speed, since the population will tend to repeatedly sample the same genotypes (Bullock, 2002). This may be related to the effects of sampling fluctuation. Nimwegen *et al.* (1999) have suggested that there is an appreciable chance that all individuals on higher neutral networks will be lost through sampling fluctuations[4]. These results demonstrate that high selection pressure decreases the probability with which the individuals are lost through sampling fluctuation, with the result that the optimal mutation rate and error threshold are shifted to higher mutation rates. This would explain why, when q exceeds q_o, the moving speed improves if (relatively) high selection pressure is applied.

[4]Higher neutral networks will tend to have narrower neutral pathways, with more bits on the genotype being subject to *selective constraints*.

2.3 Complex Test Function

2.3.1 Terraced NK Landscape

A terraced NK landscape is the tunably neutral NK landscape proposed by Newman *et al.* (1998). A terraced NK landscape has three parameters: N, the length of the genotype; K, the number of epistatic linkages between genes; and w, the contribution of a locus to the fitness of the entire genotype.

The fitness value is calculated as follows: The fitness contribution of the i-th locus, w_i, is an integer, generated randomly in the range $0 \leq w_i < F,\ i = 1, \cdots, N$. To calculate the fitness, W, of a genotype, the fitness contribution of each locus is averaged, and then divided by $F - 1$, normalizing W to the range 0.0 to 1.0. More formally:

$$W = \frac{1}{N(F-1)} \sum_{i=1}^{N} w_i. \tag{2.2}$$

The neutrality of the landscape can be tuned by changing the value of F. The neutrality of the landscape is maximized when $F = 2$, and is effectively non-existent as $F \to \infty$.

2.3.2 Computer Simulations

Simulation Conditions

Computer simulations were conducted using a population size of 50. The OGA uses the same genetic operators as in Section 2.2: *connection*, *division*, *duplication*, *deletion* and *inversion*. The probabilities for genetic operations were set at 0.3 for *connection* and *division*, 0.6 for *duplication* and 0.3 for *deletion* and *inversion* (in the cases that $K \geq 5$ for $s = 6$ with elitism), and 0.3 for *connection* and *division*, 0.2 for *duplication* and 0.05 for *deletion* and *inversion* (in the other cases). The length of the value list in a locus was 6. The genetic operation for the SGA was standard bit mutation. For both GAs, the per-bit mutation rate, q, was set at 0.01, according to Mühlenbein's suggestion (Mühlenbein, 1993). Crossover was not used with either GA. Tournament

selection was adopted. *Elitism*[5] was optionally applied. The tournament size, s, was set at $\{2, 6\}$ because low selection pressure is generally preferable with the SGA, whereas high selection pressure is preferable with the OGA. A generational GA was used. 50 independent runs were conducted for each problem under the landscape parameters, $N = 100$, $K = \{0, 5, 15, 25\}$, $F = 2$. Each run lasted 3,000 generations. All results were averaged over 50 runs.

Simulation Results

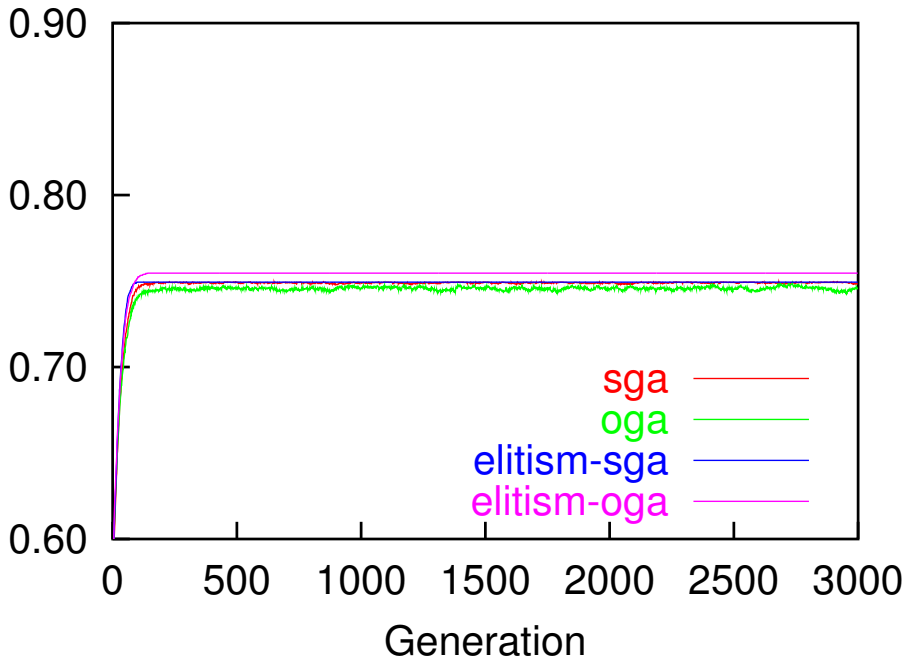
Fig. 2.6 to 2.9 show the maximum fitness at each generation for the SGA and OGA, with and without elitism, and for different values of K and s.

For $K = 0$, there is no ruggedness, and the landscape is similar to simple neutral networks of the BBF. Fig. 2.6(a) and 2.6(b) show the results for the four GA conditions for $s = 2$ and 6 respectively. No significant differences in performance were observed. This is consistent with the results obtained with simple neutral networks using the BBF in Chapter 2.2.

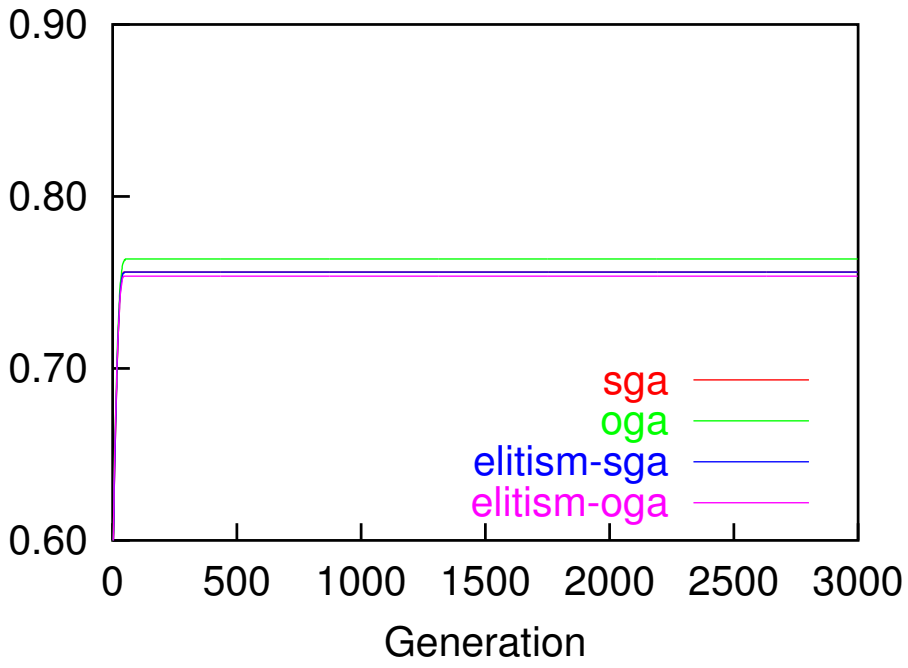
For $K = 5$, both GAs performed better with elitism than without it for $s = 2$ (Fig. 2.7(a)). However, for $s = 6$ there was no significant difference between the four GA conditions (Fig. 2.7(b)). Fitness increased faster for $s = 6$ than for $s = 2$.

For $K = 15$ and $K = 25$, differences between the SGA and the OGA were much more pronounced than at $K = 0$ and $K = 5$. As with $K = 5$, both GAs performed better with elitism than without it for $s = 2$ (Fig. 2.8(a), 2.9(a)). The OGA was outperformed by the SGA for $s = 2$, however, the OGA outperformed the SGA for $s = 6$ (Fig. 2.8(b), 2.9(b)). A closer examination reveals that the OGA performed better for $s = 6$ than the SGA did for $s = 2$ with elitism.

[5] An individual, randomly selected from those individuals with the highest fitness value, is passed unmutated to the next generation.

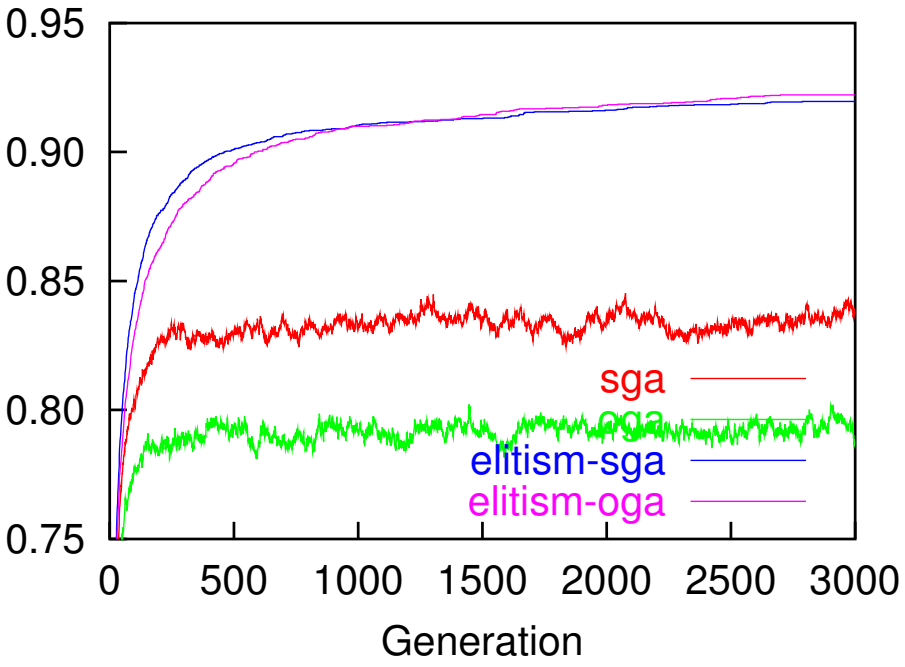


(a) $(K, s) = (0, 2)$

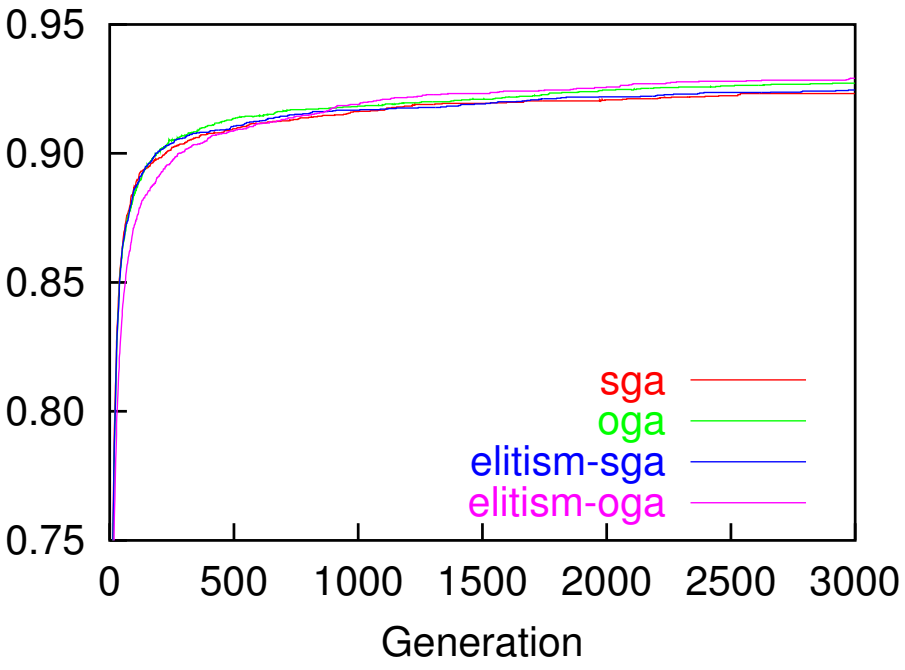


(b) $(K, s) = (0, 6)$

Figure 2.6: Maximum fitness at each generation for $K = 0$

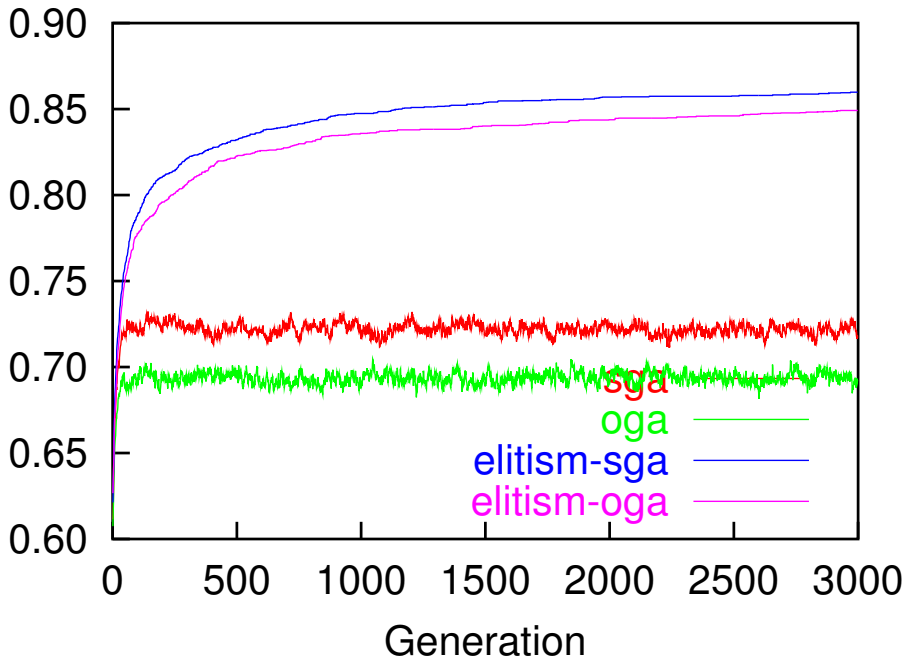


(a) $(K, s) = (5, 2)$

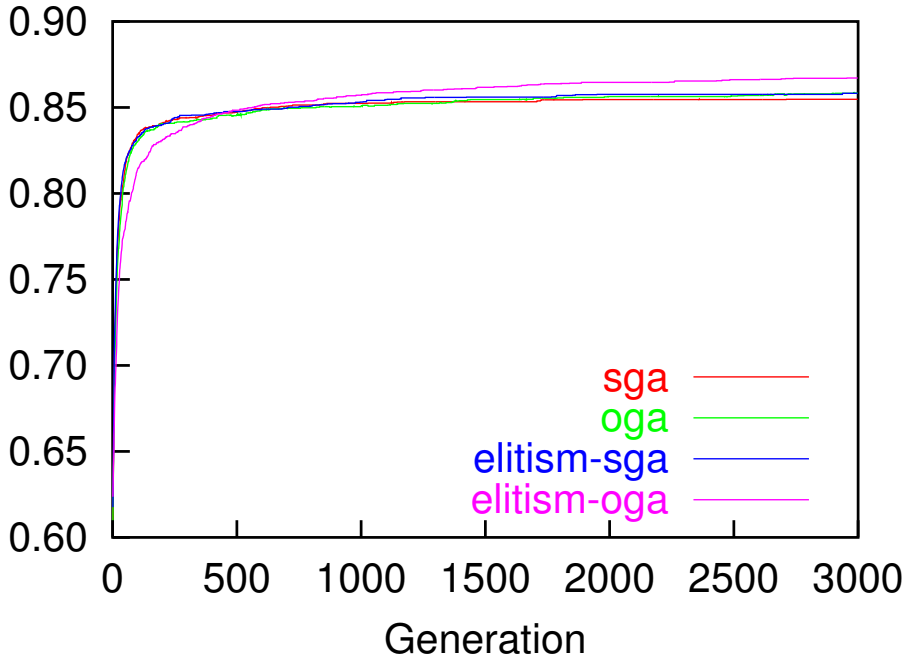


(b) $(K, s) = (5, 6)$

Figure 2.7: Maximum fitness at each generation for $K = 5$

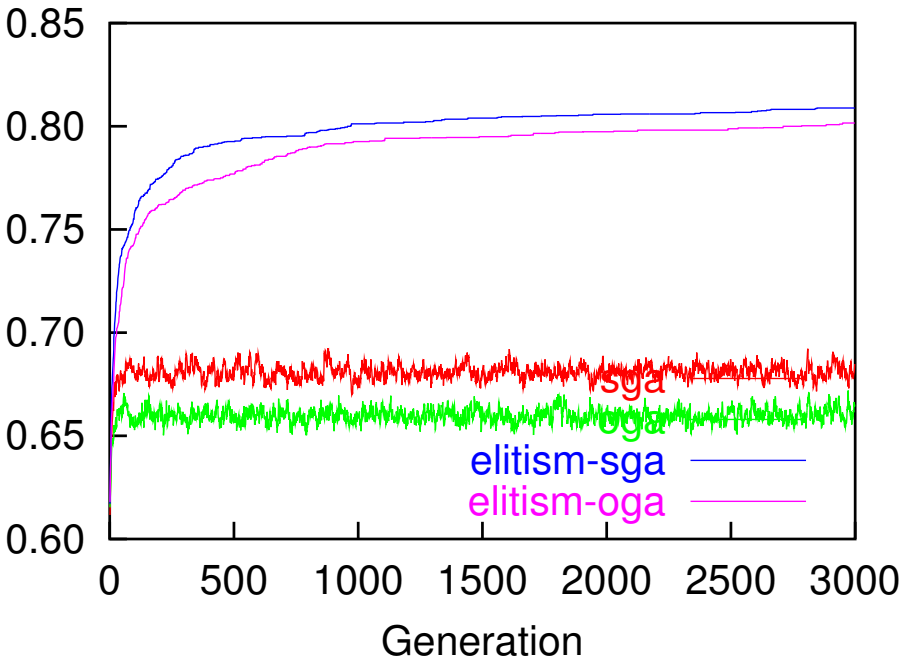


(a) $(K, s) = (15, 2)$

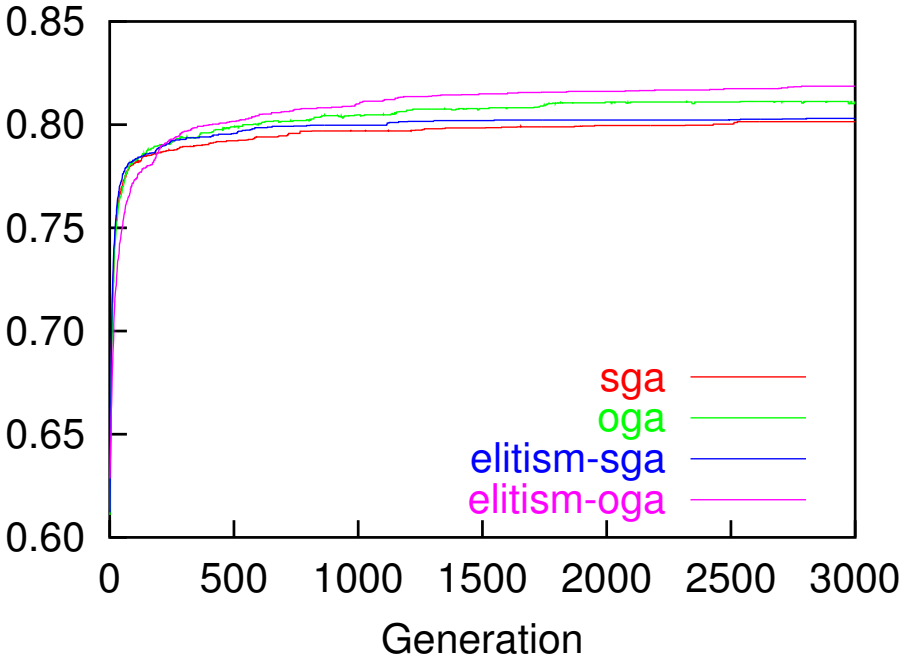


(b) $(K, s) = (15, 6)$

Figure 2.8: Maximum fitness at each generation for $K = 15$



(a) $(K, s) = (25, 2)$



(b) $(K, s) = (25, 6)$

Figure 2.9: Maximum fitness at each generation for $K = 25$

2.3.3 Variable Mutation Rate Strategy against Ruggedness

The obtained results can be summarized as follows:

- *the effect of the error threshold*

  Without elitism, both types of GA search inefficiently when landscapes are rugged and selection pressure is low. The OGA performs slightly worse than the SGA, as shown in Fig. 2.7(a), 2.8(a) and 2.9(a). This is due to the fact that the effective mutation rate is more likely to exceed the error threshold when selection pressure is low (van Nimwegen and Crutchfield, 1998). This is consistent with the results obtained using the BBF (Fig. 2.5).

- *the variable mutation rate strategy*

  The variable mutation rate strategy of the OGA is a better approach on highly rugged landscapes when selection pressure is high. For $s = 6$ and, with elitism for $s = 2$, the SGA is easily trapped on local optima when the landscape is highly rugged ($K = 15$ and $K = 25$); in contrast, on the same landscapes, for $s = 6$, the OGA continues to find better regions. This improvement is due to the online adaptation of mutation rates during the process of evolution.

2.4 Summary

This chapter investigated the effect of selection pressure and mutation rate on the speed of population movement on very simple neutral networks with different levels of neutrality. Then, examined the performance of GAs in the terraced NK landscapes with different levels of ruggedness and different selection pressures. Two types of GA were used. The results can be summarized as follows:

- For a given population size, the speed of a population plotted as a function of the mutation rate yields a concave curve with an optimal mutation rate and an error threshold.
- Increasing the selection pressure will improve the speed at which a population moves on a neutral network.
- The benefits of the variable mutation rate strategy become increasingly clear as the ruggedness of the landscapes increases.

These results suggest some guidelines for tuning the performance of GAs. Chapter 4 will investigate how well these tuning guidelines apply to real-world problems, such as the evolution of artificial neural networks for robot control.

Chapter 3

Foundation: Estimating Neutrality

3.1 Introduction

This chapter investigates systematically the characteristics of the Nei's standard genetic distance in fitness landscapes with neutrality in various conditions to validate the use of the standard genetic distance for estimating the degree of neutrality in fitness landscapes.

Section 3.2 describes the Nei's standard genetic distance. Section 3.3 applies the Nei's genetic distance to a tunably neutral landscapes and shows the results. Section 3.4 examines the measure of ruggedness as well as the genetic distance, considering real-world problems. Finally, Section 3.5 discusses the *error threshold* on the population size and the mutation rate based on the obtained results and the chapter closes with a summary of the main points.

3.2 The Nei's Standard Genetic Distance

Genetic distance is a term of population genetics used for estimating gene differences per locus between populations. Although there are several definitions for this, the Nei's standard genetic distance (Nei, 1972), which is often used for that sake in population genetics, is adopted in this thesis.

The Nei's standard genetic distance is defined as follows. Consider two populations, X and Y. Let x_{ik} and y_{ik} be the frequencies[1] of the k-th alleles ($i = 1, \cdots, N$, $k \in \{1, 2\}$ in a binary coded GA) in X and Y, respectively. The probability of identity of two randomly chosen genes is $j_{xi} = x_{i1}^2 + x_{i2}^2$ in the population X, while it is $j_{yi} = y_{i1}^2 + y_{i2}^2$ in the population Y. The probability of identity of a gene from X and a gene from Y is $j_{xyi} = x_{i1}y_{i1} + x_{i2}y_{i2}$. The normalized identity of genes between X and Y with respect to a locus is defined as

$$I_i = \frac{j_{xyi}}{\sqrt{j_{xi}}\sqrt{j_{yi}}} \tag{3.1}$$

where, $I_i = 1.0$ if the two populations have the same alleles in identical frequencies, and $I_i = 0.0$ if they have no common alleles. The normalized identity of genes between X and Y with respect to the average in all loci is defined as

$$I = \frac{J_{XY}}{\sqrt{J_X}\sqrt{J_Y}} \tag{3.2}$$

where, $J_X = \sum_{i=1}^{N} j_{xi}/N$, $J_Y = \sum_{i=1}^{N} j_{yi}/N$ and $J_{XY} = \sum_{i=1}^{N} j_{xyi}/N$. The genetic distance between X and Y is defined as

$$D = -\log_e I \tag{3.3}$$

under the assumption that the mutation rate per locus is sufficiently small. However, the above definition cannot be applied to the standard GA directly, because population genetics assumes that a new allele always appears on a locus when a mutation occurs, while "back mutations (Ohta, 1992)" frequently occur in the standard GA, due to the binary coding scheme. Therefore, the genetic distance for GAs is extended as follows:

$$D = \sum_{1}^{T-1} D_{t,t+1} \tag{3.4}$$

where T is the number of the final generation and $D_{t,t+1}$ is the genetic distance between the populations in the t-th and the $(t+1)$-th generation.

[1]The term, "frequency" often means a proportion in population genetics.

3.3 The Genetic Distance in a Test Function

3.3.1 Simulation Conditions

Two genetic algorithms were employed: the SGA and the (*random-sampling*, q)-algorithm. The (*random-sampling*, q)-algorithm employs standard bit mutation at the rate of q as the genetic operation and random sampling as a selection method where M offsprings are sampled from M ancestors with replacements. This model was used to investigate the effect of random sampling with mutation on a genetic distance. This is approximately equivalent with Kimura's stochastic genetic models to study random genetic drift and the expected time of fixation of a mutant gene (Kimura, 1983).

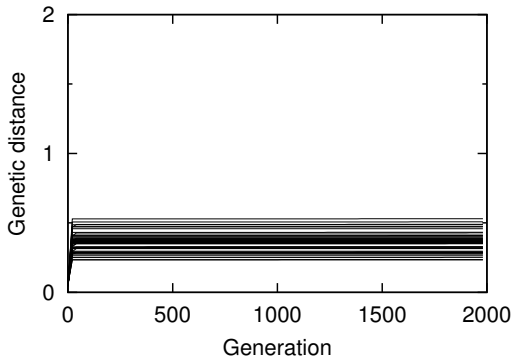
Computer simulations were conducted by varying the population size, M, and the mutation rate, q. The SGA was applied to terraced NK landscapes (Section 2.3) varying the landscape parameters. The SGA used standard bit mutation as the genetic operation. Crossover was not employed. Tournament selection was adopted for the SGA. The tournament size was set at 2 for the SGA. Each run lasted 2,000 generations. 50 independent runs were conducted for each problem under the landscape parameters, $N = 20$, $K \in \{0, 2, 6, 12, 19\}$, $F \in \{2, 3, 4, 6, \infty^2\}$. By using the genotype data obtained in 50 runs, the Nei's standard genetic distance were calculated.

3.3.2 Simulation Results

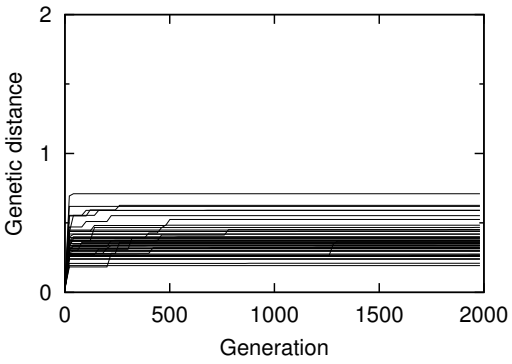
Existence and Non-existence of Neutrality

The first experiments were conducted to investigate the effect of the existence of neutrality on the transition of the genetic distance. Fig. 3.1 shows the genetic distance of the SGA for $F = \infty$, where q was set at 0.008 based on the assumption of Equation 3.3. They leveled off in the very early generations. This means that the population converged to a certain point in the genotype space then the genetic distance between the generations ($D_{t,t+1}$ in Equation 3.4) became zero. In contrast, the genetic distance

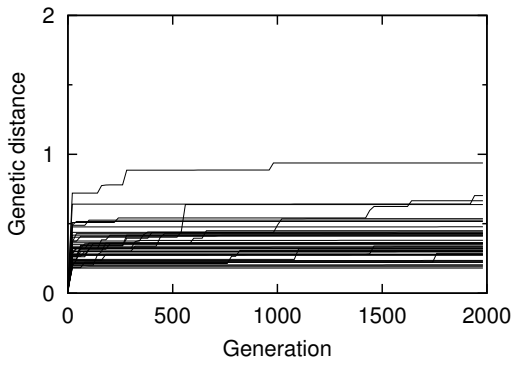
[2]For $F = \infty$, the NK fitness landscape (Kauffman, 1993) was employed instead of the terraced NK landscape as (Smith et al., 2002a), which results in practically non-existence of neutrality.



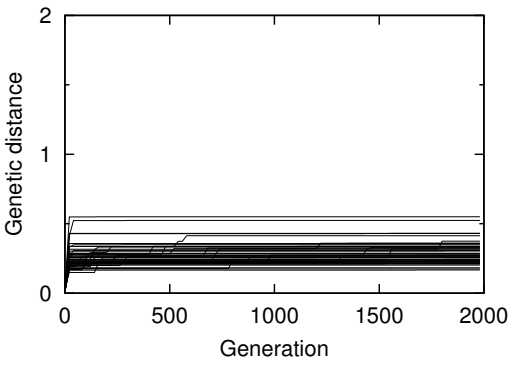
(a) $K = 0$



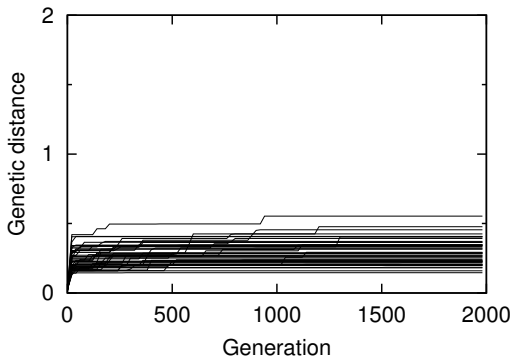
(b) $K = 2$



(c) $K = 6$

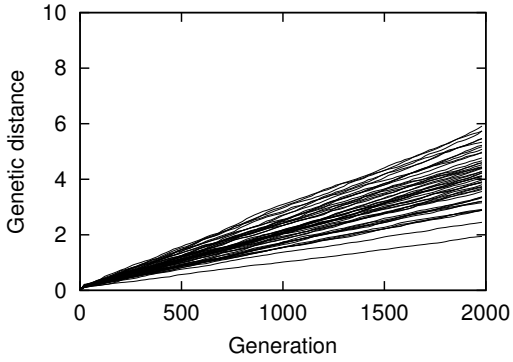


(d) $K = 12$

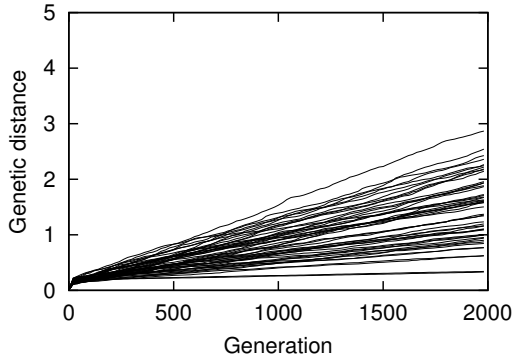


(e) $K = 19$

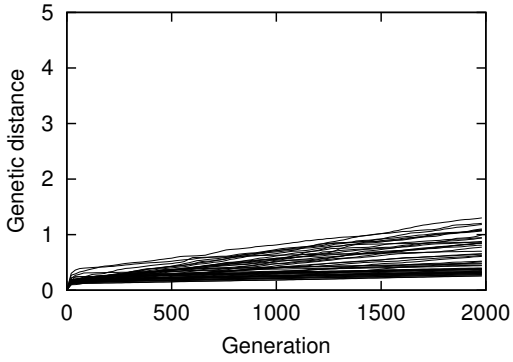
Figure 3.1: Genetic distance at each generation for the SGA with $q = 0.008$ and $M = 50$ for $F = \infty$ in 50 runs



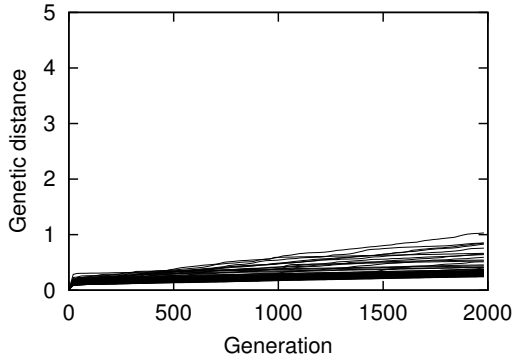
(a) $K = 0$



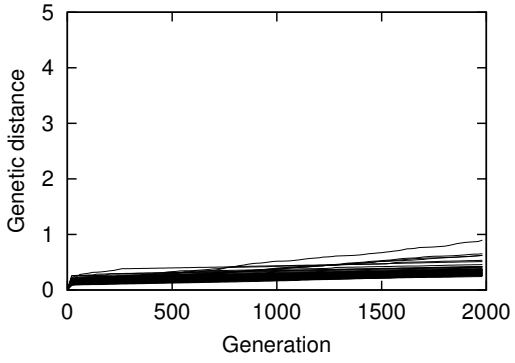
(b) $K = 2$



(c) $K = 6$



(d) $K = 12$



(e) $K = 19$

Figure 3.2: Genetic distance at each generation for the SGA with $q = 0.008$ and $M = 50$ for $F = 2$ in 50 runs

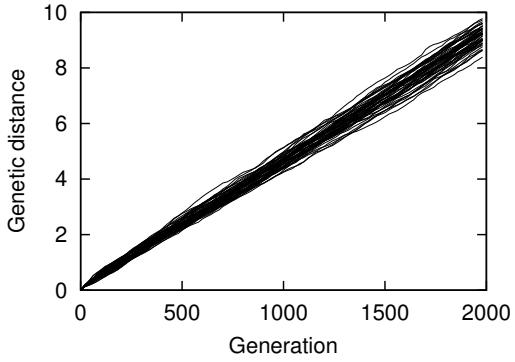


Figure 3.3: Genetic distance at each generation for the (*random-sampling*, q)-algorithm with $q = 0.008$ and $M = 50$ in 50 runs

of the SGA for $F \neq \infty$ (for instance, the results for $F = 2$ are shown in Fig. 3.2.) and the (*random-sampling*, q)-algorithm (Fig. 3.3) with $q = 0.008$ increased approximately linearly over generations in all runs. This differentiates between the existence and the non-existence of neutrality in a fitness landscape. That is, the increase of the genetic distance over generations indicates the presence of neutrality in a fitness landscape.

In the remainder of this experiment, the gradient of the genetic distance over generations, α, for the (*random-sampling*, q)-algorithm and the SGA for $F \neq \infty$ are shown by using the method of least squares on the results of all the runs.

Neutrality and Selective Constraint

With respect to the assertions 1, 2 and 3 mentioned in Chapter 1, Kimura (1983) has suggested that the rate of gene substitution is largest when the selective advantage of a new mutation over the original allele is zero except that the new mutation is deleterious in a small population. Thus, it seems likely that the genetic distance increases with the increase of neutrality and that for the (*random-sampling*, q)-algorithm is largest, because *random-sampling* can be considered completely neutral for selection. In addition to this, according to Ohta's nearly neutral theory (Ohta, 1992, 1998), the stronger the selective constraint on the molecule is, the lower its rate of evolution becomes. That is, the genetic distance is likely to decrease with the increase of selective constraint, K.

Table 3.1 and Fig. 3.4(a) show α for the SGA with $q = 0.008$ and $M = 50$. Notice first that α increased with the decrease of F for all Ks. This means that the genetic distance increases with the increase of neutrality as predicted. Secondly, α decreased with the increase of K for all Fs. This means that not only neutrality but also ruggedness has an influence on the genetic distance. This tendency is consistent with Ohta's results for NK landscapes with weak selection based on the nearly neutral theory, where the number of substitutions decreases with the increase of K (Ohta, 1997, 1998). Similar behavior to $M = 50$ is shown for each population size $M = \{100, 200, 400\}$ (Table 3.2, 3.3 and 3.4, and Fig. 3.4(b), 3.4(c) and 3.4(d)).

α for the (*random-sampling*, q)-algorithm with each M is shown in Table 3.5. It is confirmed that for each M, α for the (*random-sampling*, q)-algorithm was always larger than any others for the SGA with K and F(from Table 3.1 to 3.4). This agrees with the above prediction.

Varying the Population Size

In the next experiments, the analysis was extended by varying the population size. According to Ohta's nearly neutral theory (Ohta, 1992, 1998), population movement depends on the population size. That is, mutant dynamics becomes slower by increasing the population size. Fig. 3.5(a) to 3.5(d) show α for the SGA with each M. With the increase of the population size, α decreased for each K and F. Therefore, the larger the population size becomes, the slower the population moves. This tendency is also consistent with Ohta's results for NK landscapes with weak selection based on the nearly neutral theory, where the number of substitutions decreases with the increase of the population size (Ohta, 1997, 1998).

Table 3.5 shows α for the (*random-sampling*, q)-algorithm with each M. α also decreased with the increase of the population size.

Table 3.1: α for the SGA with $q = 0.008$ and $M = 50$

F \ K	0	2	6	12	19
2	0.002042	0.000666	0.000220	0.000131	0.000106
3	0.001307	0.000421	0.000173	0.000102	0.000104
4	0.001063	0.000318	0.000130	0.000099	0.000087
6	0.000738	0.000235	0.000128	0.000090	0.000087

Table 3.2: α for the SGA with $q = 0.008$ and $M = 100$

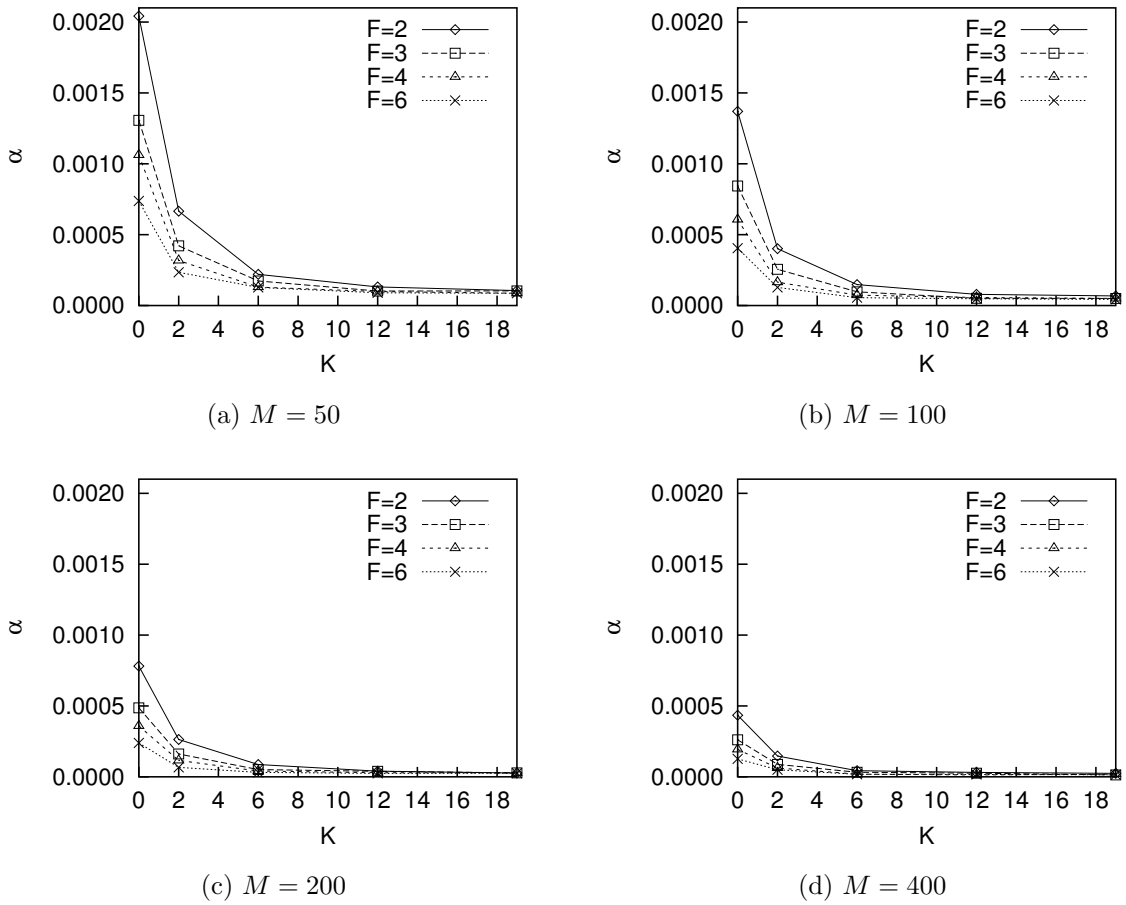
F \ K	0	2	6	12	19
2	0.001370	0.000401	0.000148	0.000079	0.000068
3	0.000843	0.000255	0.000096	0.000051	0.000050
4	0.000607	0.000164	0.000073	0.000057	0.000050
6	0.000405	0.000128	0.000056	0.000048	0.000044

Table 3.3: α for the SGA with $q = 0.008$ and $M = 200$

F \ K	0	2	6	12	19
2	0.000781	0.000263	0.000086	0.000040	0.000026
3	0.000486	0.000161	0.000051	0.000039	0.000028
4	0.000359	0.000119	0.000041	0.000029	0.000027
6	0.000238	0.000067	0.000031	0.000025	0.000026

Table 3.4: α for the SGA with $q = 0.008$ and $M = 400$

F \ K	0	2	6	12	19
2	0.000434	0.000146	0.000043	0.000032	0.000024
3	0.000261	0.000087	0.000032	0.000025	0.000013
4	0.000193	0.000062	0.000019	0.000014	0.000019
6	0.000127	0.000050	0.000017	0.000017	0.000015



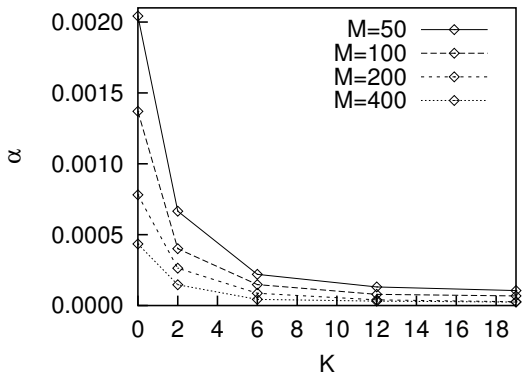
(a) $M = 50$

(b) $M = 100$

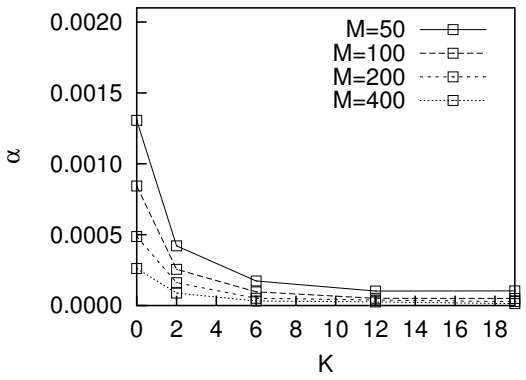
(c) $M = 200$

(d) $M = 400$

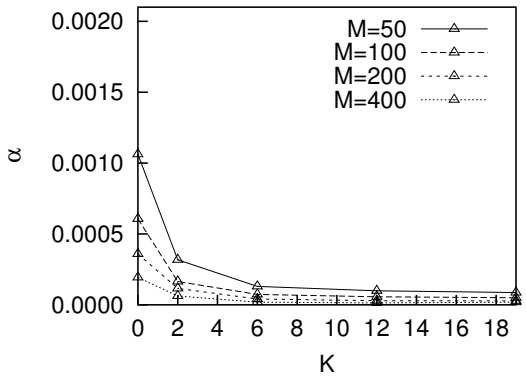
Figure 3.4: α for the SGA at $q = 0.008$ with each M



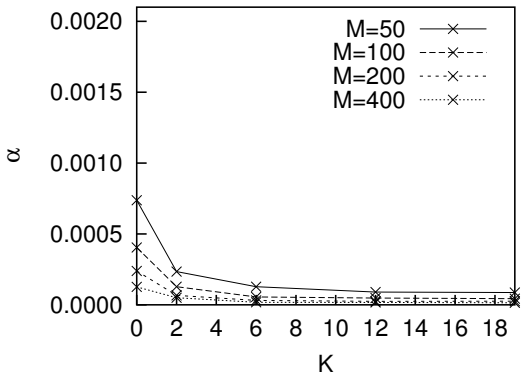
(a) $F = 2$



(b) $F = 3$



(c) $F = 4$



(d) $F = 6$

Figure 3.5: α for the SGA at $q = 0.008$ for each F

Table 3.5: α for the (*random-sampling*, q)-algorithm with $q = 0.008$ for each population size

M	50	100	200	400
$rate$	0.004576	0.003194	0.001971	0.001120

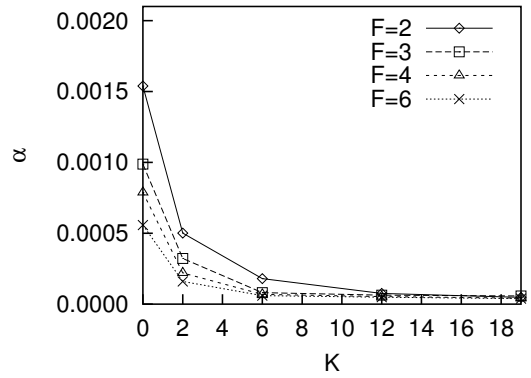
Varying the Mutation Rate

In population genetics, it is assumed that the mutation rate per locus is sufficiently small as mentioned in Section 3.2. In the last series of experiments, the transition of the Nei's genetic distance were observed by varying the mutation rate, $q = \{0.005, 0.006, \cdots, 0.010, 0.020, \cdots, 0.05, 0.07, 0.1\}$, from relatively small one to large one for the SGA with $M = 50$.

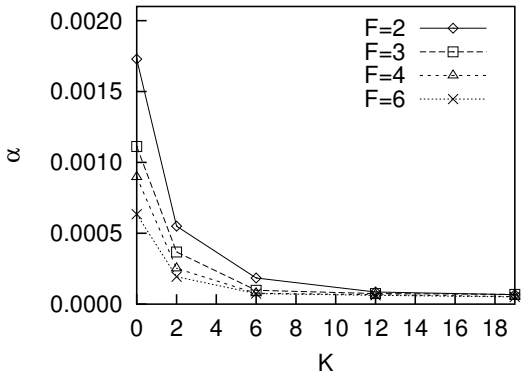
Fig. 3.6 shows the results with $q = \{0.005, \cdots, 0.010\}$. In this range, α increased with the increase of the mutation rate for each K and F. For each q, similar behaviors were observed to the results with $q = 0.008$ (mentioned in the previous subsections). In contrast, there became less significant differences between the graphs of different Fs with the increase of the mutation rate from $q = 0.02$ to 0.1 (Fig. 3.7). Surprisingly, at $q = 0.1$, α increased with the increase of K for all Fs (Fig. 3.7(f)).

α for the (*random-sampling*, q)-algorithm with $q = 0.1$ was 0.0135733. Thus, α for the SGA was higher than that for the (*random-sampling*, q)-algorithm for $K > 2$ and all Fs. This implies that artificial evolution has changed into random search, caused by the mutation rate which is larger than the error threshold (van Nimwegen et al., 1999).

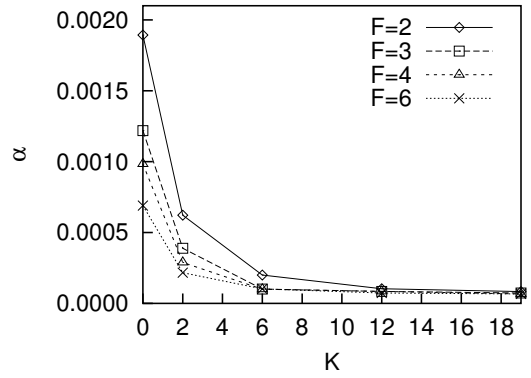
From the above, it was confirmed that the Nei's genetic distance depends on the mutation rate, and can be used as long as the mutation rate is sufficiently small compared with the error threshold.



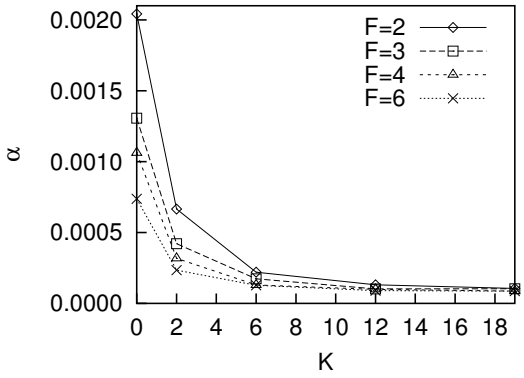
(a) $q = 0.005$



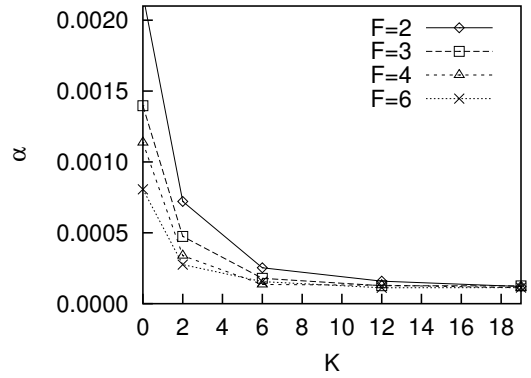
(b) $q = 0.006$



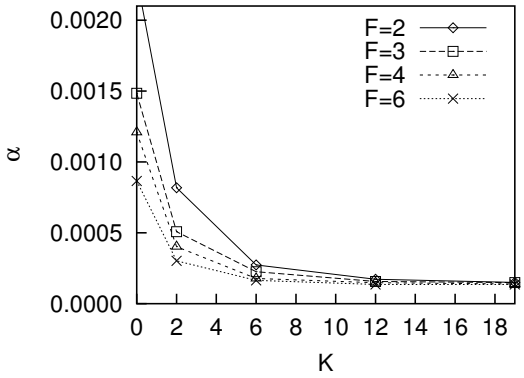
(c) $q = 0.007$



(d) $q = 0.008$

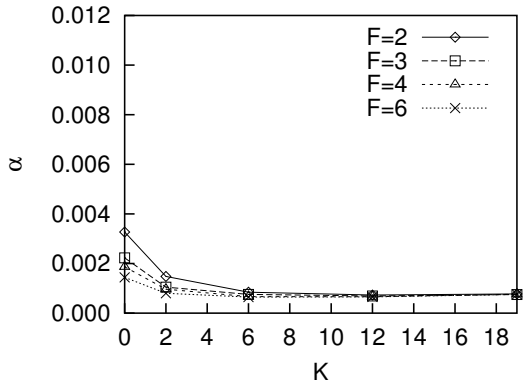


(e) $q = 0.009$

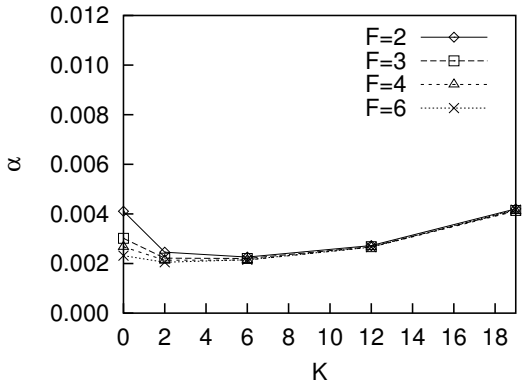


(f) $q = 0.010$

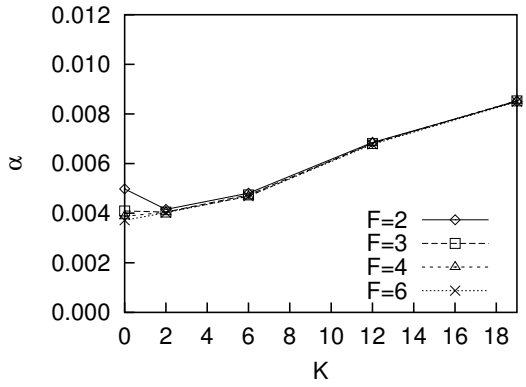
Figure 3.6: α for the SGA with $M = 50$ at $q = \{0.005, 0.006, \cdots, 0.010\}$



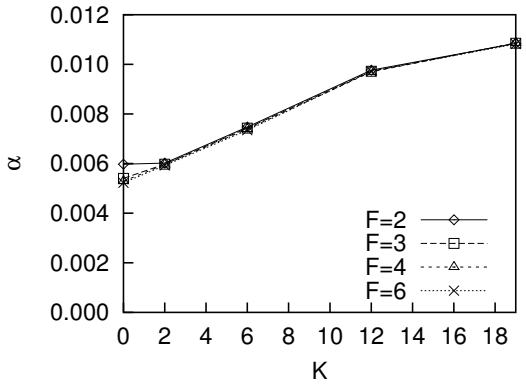
(a) $q = 0.02$



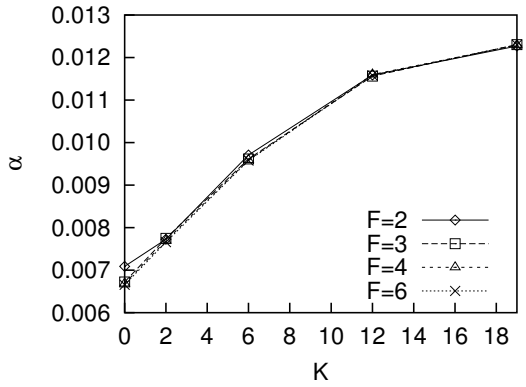
(b) $q = 0.03$



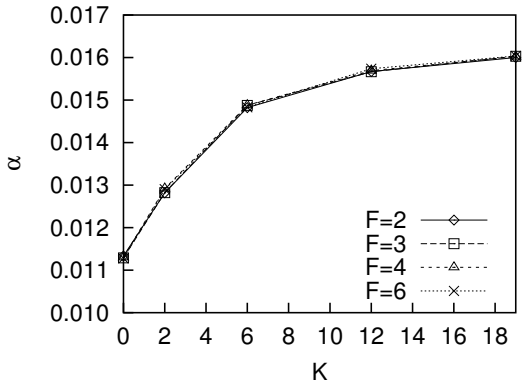
(c) $q = 0.04$



(d) $q = 0.05$



(e) $q = 0.07$



(f) $q = 0.10$

Figure 3.7: α for the SGA with $M = 50$ at $q = \{0.02, \cdots, 0.05, 0.07, 0.10\}$

3.4 Estimating Ruggedness

In real-world problems, ruggedness of a fitness landscape is predicted by the fitness correlation (Smith et al., 2002a; Weinberger, 1990). In this thesis, therefore, the Smith's measurement (Smith et al., 2002a) is employed for the measure of ruggedness because its fitness correlation can be expressed as one dimensional value.

The average fitness of the offspring solutions, called the Smith's E_b, is given by

$$E_b(k) = \frac{\sum_{g \in G(k)} F(g)}{|G(k)|} \tag{3.5}$$

where, $G(k)$ is the set of offspring from parents with the fitness k, g is an offspring genotype and $F(.)$ is the fitness function. For instance, on the same settings in the previous section, E_bs in TNK($N = 20$, $F = 2$, $K \in \{0, 2, 6, 12, 19\}$) are shown in Fig. 3.8 by using the method of least squares on 50 runs. It is found that the gradient, $\dot{E}_b$, decreases with the increase of K. It has been known that this gradient is proportional to the autocorrelation function for the mutation operator applied, and that $\dot{E}_b \simeq 1.0$ when $K = 0$ and $\dot{E}_b \simeq 0.0$ when $K = N - 1$.

Figure 3.9 shows α at each correlation ($\dot{E}_b$) corresponding to K with all Fs in TNK ($N = 20$, $K \in \{0, 2, 6, 12, 19\}$, $F \in \{2, 3, 4, 6\}$ and the other parameters were the same as in the previous section). It is confirmed that α increased with the increase of $\dot{E}_b$ for all Fs, and increased with the decrease of F for all $\dot{E}_b$s. As a result, a set of points $(\dot{E}_b, \alpha)$ composes a curve, which increases with the increase of the correlation when F is constant. A set of curves are also found with different levels of neutrality. This demonstrates that α would predict the increase of neutrality combined with the measure of ruggedness.

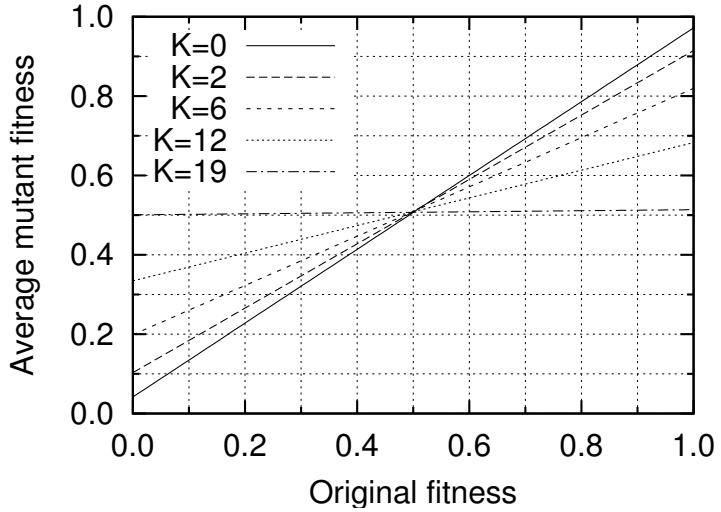


Figure 3.8: Average offspring fitness value over all parent fitness values

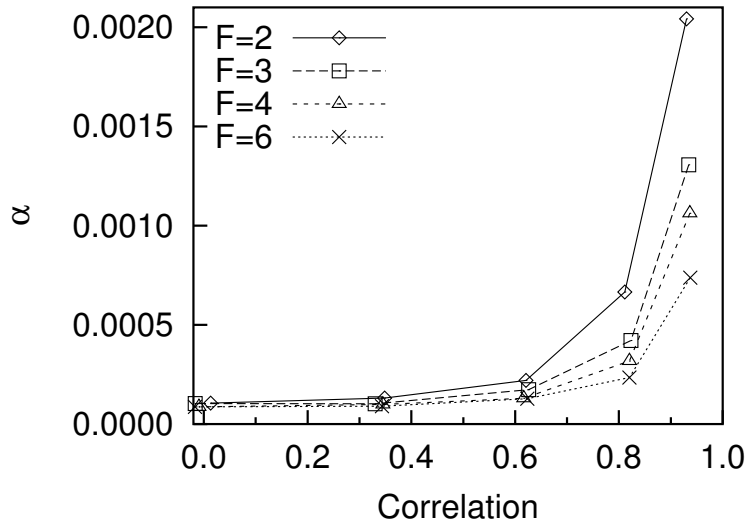


Figure 3.9: α for the SGA as a function of the correlation($\dot{E}_b$) with $q = 0.008$ and $M = 50$

3.5 Discussion

In the previous section, it has been shown that population movement depends on the population size. If the population size is small, the population for the SGA moves quickly. This would have the advantage of flexibility. As pointed out in (Ohta, 1998), evolution would become more flexible for a small population size than for a large population size, particularly when the environment is not static, that is, the fitness landscape changes occasionally. On the other hand, it has been reported that as the population size becomes too small, it becomes easier for the population to lose the current best individuals through random sampling or mutation and fall to lower neutral networks (van Nimwegen and Crutchfield, 1998). This phenomenon is due to the effects of the error threshold on the population size[3]. This would be more understandable by considering the population movement. Due to the small population size, the population on the neutral networks moves too quickly to keep the current neutral network. This implies that there exists the optimal population size that keeps the fastest speed as well as avoids the influence of the error threshold.

The same discussion can be applied to the error threshold on the mutation rate (van Nimwegen et al., 1999; Barnett, 2001) mentioned in Section 3.3.2.

[3]It has been known that there are two kinds of error threshold: on the mutation rate and on the population size (van Nimwegen and Crutchfield, 1998; van Nimwegen et al., 1999).

3.6 Summary

This chapter investigated the characteristics of the Nei's standard genetic distance by applying it to terraced NK landscapes, and confirmed the validity of the use of the Nei's standard genetic distance for estimating the degree of neutrality in fitness landscapes. It was confirmed that the results were consistent with the neutral theory and the nearly neutral theory in population genetics. The influence of the error threshold on the population size was discussed.

The characteristics of the Nei's standard genetic distance can be summarized as follows:

When the mutation rate per locus is sufficiently small,

- The genetic distance increases approximately linearly over generations in fitness landscapes with neutrality.
- Random sampling with mutation results in the largest genetic distance.
- The genetic distance increases with the increase of neutrality.
- The genetic distance decreases with the increase of ruggedness in landscapes with neutrality.
- The genetic distance decreases with the increase of the population size.

These results suggest that the genetic distance can provide a guideline for estimating the degree of neutrality combined with the measure of ruggedness. The next chapter will investigate how well the proposed method applies to describe real-world problems whose fitness landscape includes neutral networks, such as the evolution of artificial neural networks for robot control.

Chapter 4

Applications

4.1 Introduction

The previous chapters showed some guidelines based on the experimental results for tuning the performance of GAs on fitness landscapes with neutral networks and for estimating the degree of neutrality in fitness landscapes. The remainder of this thesis investigates how well those guidelines apply to real-world problems, especially to the evolution of artificial neural networks for robot control on several evolutionary robotics tasks.

Evolutionary robotics aims at providing autonomous robots with high adaptability by employing artificial evolution (Nolfi and Floreano, 2000). This is because autonomous robots should be adaptable to environments, which the system of a robot is developed through the interaction with (Pfeifer and Scheier, 1999). In evolutionary robotics, genetic algorithms and neural networks are the most used techniques. However, useful theories or methodologies for them have not been discussed enough. This would be due to landscapes with neutral networks in the problems where traditional theories are not useful, as mentioned in Chapter 1. If the proposed guidelines successfully apply to the problems, great breakthroughes are expected in that research field .

Section 4.2 and 4.3 investigate such simulated real-world problems, a discrimination problem and a pursuit problem. In both problems, it is investigated whether the

Nei's genetic distance can be used to estimate and compare the degree of neutrality in the fitness landscape among the neural network controllers (varying the number of hidden neurons in the discrimination problem) and whether the evolutionary dynamics observed in these problems are consistent with those of the test functions in Chapter 2. Section 4.4 investigates a single evolutionary run for each GA in a real-world problem, a goal reach problem. The features of the fitness landscape are estimated by using the proposed method. Based on these features, the SGA and the OGA are applied to the problem. The best evolved controllers of the neural networks are evaluated as the controller of an autonomous mobile robot in the real environment. Finally, the chapter closes with a summary of the main points.

4.2 Discrimination Problem

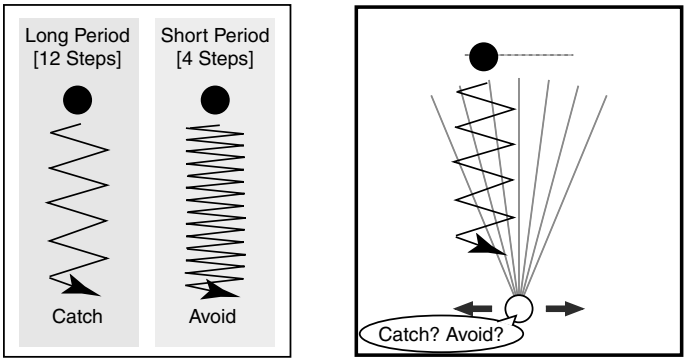


Figure 4.1: Experimental setup for a discrimination of the motion patterns. Two kinds of period used in the discrimination experiments (left) and the agent in the arena with its array of the proximity sensors (right).

For animals with vision, information of figures plays the main role in discriminating an object. It is also known that those animals discriminate an object not only by its figure but also by its motion pattern (von. Uexküll, 1926; Jörg-Peter, 1976). Thus, it is interesting to investigate whether an artificial agent can discriminate an object either by its figure or by its motion pattern, or both. Beer (1996) has demonstrated that an agent controlled by evolved dynamical neural networks can discriminate an object by its figure. Moreover, an agent must be able to discriminate an object by its motion pattern (Katada et al., 2003). For this experiment, the control task was motion pattern discrimination, and is based on a task originally implemented by Beer (1996). The agent must discriminate between two types of vertically falling object based on the object's period of horizontal oscillation; it must catch (i.e., move close to) falling objects that have a long period whilst avoiding those with a short period (see Fig. 4.1). An array of proximity sensors allow the agent to perceive the falling objects. If an object intersects a proximity sensor, the sensor outputs a value inversely proportional to the distance between the object and the agent. The agent can move horizontally along the bottom of the arena. For this experiment, the agent of diameter 30 had 7 proximity sensors of maximum range 220 uniformly distributed over a visual angle of 45 degrees. The horizontal velocity of the agent was proportional to the sum

of the opposing horizontal forces produced by a pair of effectors. It has maximum velocity of 8. Each falling object was circular, with diameter 30, and dropped from the top of the arena with a vertical velocity of 4, a horizontal amplitude of 30 and an initial horizontal offset of ±50. An object's horizontal velocity was ±10 (12 steps in a period) for a long period and ±30 (4 steps in a period) for a short period.

The performance measure to be maximized was as follows:

$$Fitness = 1000 \sum_{i=1}^{NumTrials} \frac{P_i}{NumTrials}, \tag{4.1}$$

where $P_i = 1 - d_i$ for a long period and $P_i = d_i$ for a short period, $d_i = 1$ when $hd_i > 60$ and $d_i = hd_i/60$ when $hd_i \leq 60$, hd_i is the final horizontal distance between the center of the agent and the object, and $NumTrials$ is the number of trials for an individual (8 trials for each period).

4.2.1 Describing the Fitness Landscapes

Simulation Conditions

For this experiment, the agent controller was a PNN with 7 sensory neurons, 2 fully interconnected motor neurons and N_h fully interconnected hidden neurons, where $N_h \in \{0, 1, 5, 10, 15\}$ in order to estimate and compare the features among the fitness landscape with each N_h. The network's connection weights and the firing threshold for each neuron were genetically encoded and evolved. The total number of parameters is equal to $\{20, 33, 105, 240, 425\}$ corresponding to each N_h. The parameters were mapped linearly with the following ranges: connection weights $\omega \in [-1.0, 1.0]$, thresholds $\theta \in [0.0, 3.9]$. The parameters of the neurons and synapses (see Appendix D) were set as follows: $\tau_m = 4$, $\tau_s = 10$, $\Delta^{ax} = 2$ for all neurons and all synapses in the network following the recommendations given in (Floreano and Mattiussi, 2001).

Computer simulations were conducted using populations of size 50. Each individual was encoded as a binary string with 10 bits for each parameter. Therefore, the total length of the genotype was $L = \{200, 330, 1050, 2400, 4250\}$. The SGA were adopted to evolve PNN parameters. The genetic operation for the SGA was standard

bit mutation. Based on the assumption of Equation 3.3 in Chapter 3, two types of mutation rate were set as follows:

a) $q = 1/L_{N_h=15}$, which is constant for each landscape, corresponding to $1/L$ for the longest length in Ls.

b) $q = 1/L$, following the recommendation in the evolutionary computation community.

According to the characteristics of the Nei's standard genetic distance investigated in Chapter 3, all the parameters of the SGA must be the same among fitness landscapes to compare the features. Thus, the other parameters were set as follows. Tournament selection was adopted. Elitism was applied. The tournament size was set at 2 because the SGA generally prefers low selection pressure. A generational model was used. Each run lasted 6,000 generations. 10 independent runs were conducted for each PNN controller.

As the genetic distance depends on both neutrality and ruggedness (Chapter 3), the features of test functions, which were obtained in the complementary computer simulations with the same parameters of the SGA as those of the EANNs, were used as baselines for estimating and comparing indirectly the features of the EANNs. This is because in the case of **a)** there are cases that it is not possible to estimate and compare the degree of neutrality between fitness landscapes even if combined with the measure of ruggedness (will be described in the next subsection), and in the case of **b)** there is no theoretical meaning to compare directly the features of the EANNs due to the different mutation rates. As the test functions, TNK and TNKp were adopted. TNKp is an extended form of TNK for $F = 2$ in order to increase neutrality of TNK. For TNKp, following the way to involve neutrality in NKp fitness landscapes (Barnett, 1997, 1998), the probability P ($0 \leq P \leq 1$) is introduced to make the fitness contribution of each locus (w_i in Equation 2.2) 0. In this experiment, the landscape parameters of TNK(p) were set as follows: $N = 20, K \in \{0, 2, 6, 12, 19\}, F \in \{2, 3, 4\}, (P \in \{0.9, 0.99\})$.

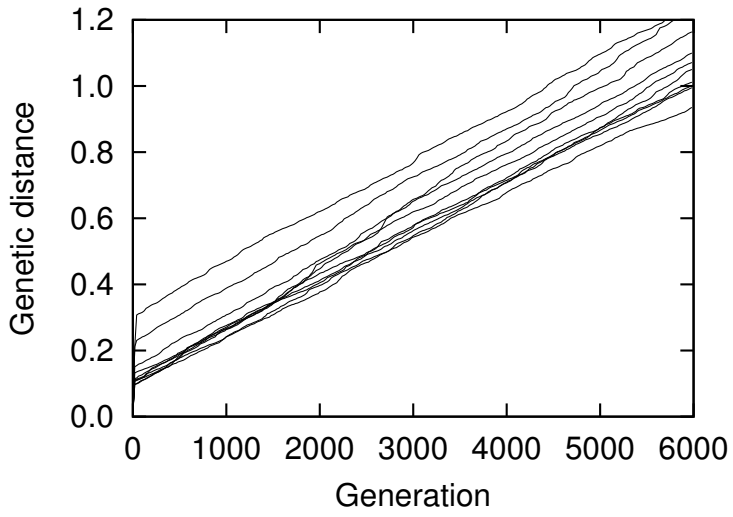


Figure 4.2: Genetic distance at each generation for the SGA with $q = 1/L$ for $N_h = 15$ in 10 runs

Simulation Results

a) $q = 1/L_{N_h=15}$

Fig. 4.2 shows the genetic distance at each generation for $N_h = 15$. The approximately linear increases were observed in all runs. For other N_hs, the same transition were observed. From the results in Chapter 3, this indicates the presence of neutrality in the fitness landscape of the EANNs. Thus, α is shown by using the method of least squares on the results of all the runs in the following parts.

Fig. 4.3, 4.4 and 4.5 show the features of the fitness landscape for each N_h. Fig. 4.3 plots the correlation ($\dot{E}_b$) for each N_h. In this experiment, the correlation increased with the increase of N_h except of $N_h = 15$. That is, ruggedness decreased with the increase of N_h. In Fig. 4.4, α increased with the increase of N_h. When the fitness landscape for $N_h = 1$ was compared with the one for $N_h = 0$, there was no significant differences in the correlation but α increased so much. For another instance, when the fitness landscape for $N_h = 15$ was compared with the one for $N_h = 10$, the correlation decreased but α increased. In these cases, the increase of neutrality can be estimated based on the results obtained in Chapter 3. Note that the increase of neutrality cannot be estimated when the fitness landscape for $N_h = 5$ was compared with the

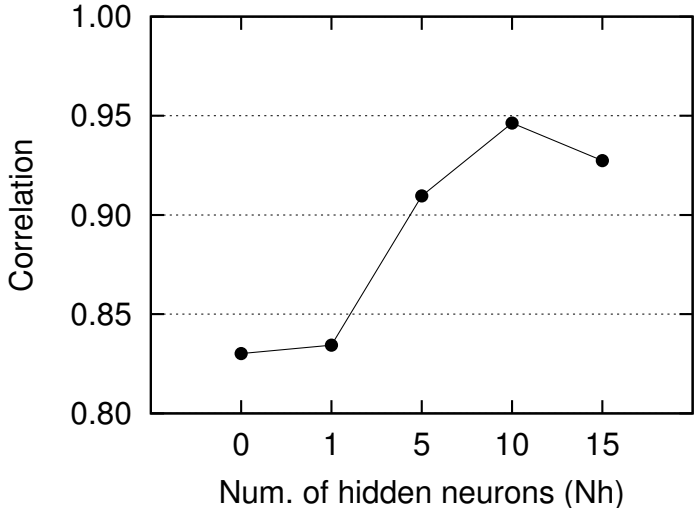


Figure 4.3: Correlation ($\dot{E}_b$) for each N_h at $q = 1/L_{N_h=15}$

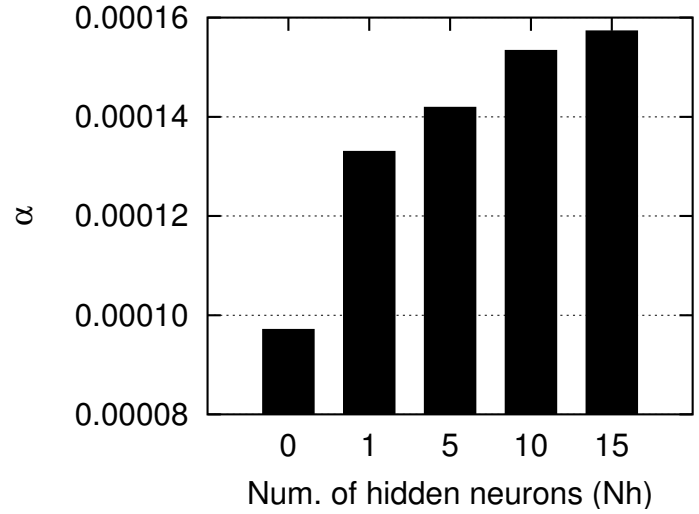


Figure 4.4: α for each N_h at $q = 1/L_{N_h=15}$

one for $N_h = 1$ or the fitness landscape for $N_h = 10$ was compared with the one for $N_h = 5$, because the increase of α coincided with the increase of the correlation.

By using the $\dot{E}_b$-α curves obtained in the TNK(p) landscape (F, P are constant) as baselines, the neutrality of all the EANNs were estimated indirectly (Fig. 4.5). It was observed that neutrality increased in order of $N_h = 10 \rightarrow \{5, 15\} \rightarrow 0 \rightarrow 1$ by using the curve for $P = 0.9$ as a baseline. The increase of neutrality was observed in the cases of $N_h = 0 \rightarrow 1$, $10 \rightarrow 15$, which are consistent with the results of the direct comparison between the landscapes mentioned above. Additionally, the decrease of neutrality was also observed in the cases of $N_h = 1 \rightarrow 5$, $5 \rightarrow 10$ although it was not possible to estimate them by the direct comparison.

Fig. 4.6 shows the maximum fitness at each generation for N_h. Except of $N_h = 1$, the fitness increased faster as the decrease of N_h, that is, the decrease of the genotypic search space. The poor performance in the SGA for $N_h = 1$ cannot be predicted by only the correlation. It would be predicted by the increase of neutrality observed in Fig. 4.5 where the genetic search would become inefficient due to large neutrality.

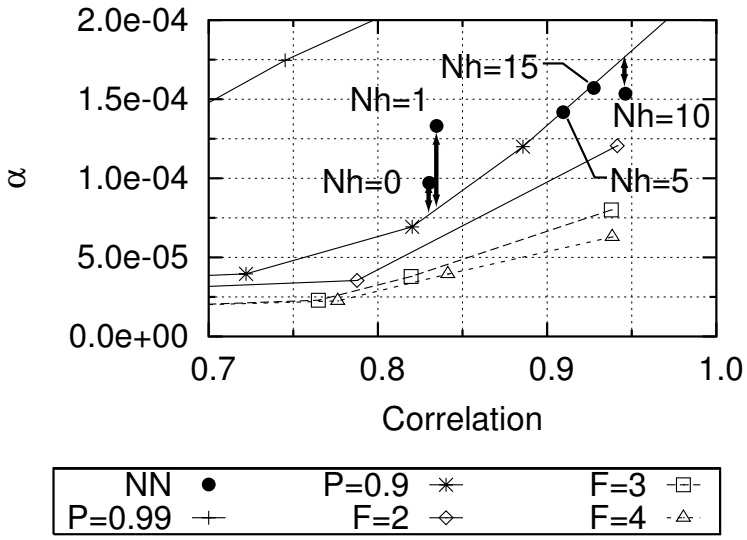


Figure 4.5: α as a function of the correlation ($\dot{E}_b$) at $q = 1/L_{N_h=15}$

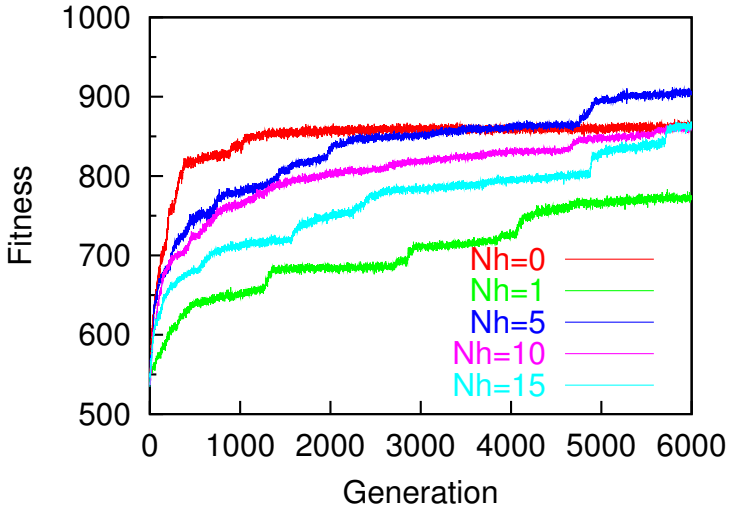


Figure 4.6: Maximum fitness at each generation at $q = 1/L_{N_h=15}$ for each N_h

b) $q = 1/L$

Fig. 4.7 plots the correlation ($\dot{E}_b$) for each N_h. As observed in **a)**, the correlation increased with the increase of N_h except of $N_h = 15$. Because of different q among the landscapes, neutrality of the EANNs were estimated indirectly by using the $\dot{E}_b$-α curves obtained in the TNK(p) landscape (F, P are constant) as baselines (Fig. 4.8). By using the curve for $P = 0.9$ as a baseline, it was observed that neutrality increased in order of $N_h = 10 \rightarrow 5 \rightarrow \{1, 15\} \rightarrow 0$.

Fig. 4.9 shows the maximum fitness at each generation for N_h. The fitness increased faster as the decrease of N_h, that is, the decrease of the genotypic search space. This would be explained as follows; In the process of evolution, no error threshold effects were observed. This implies that the effective mutation rate at $q = 1/L$ would be below the error threshold under each condition. Therefore, neutrality of the landscapes would not explicitly have any influences on the performances of the SGA but the size of the genotypic search space would have influences on them.

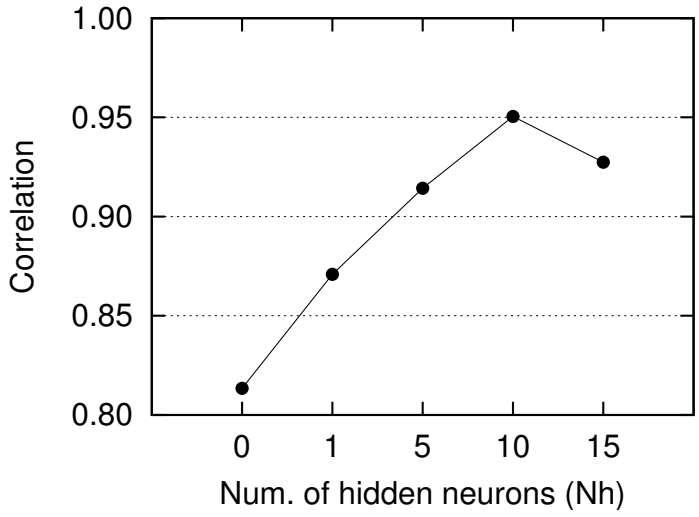
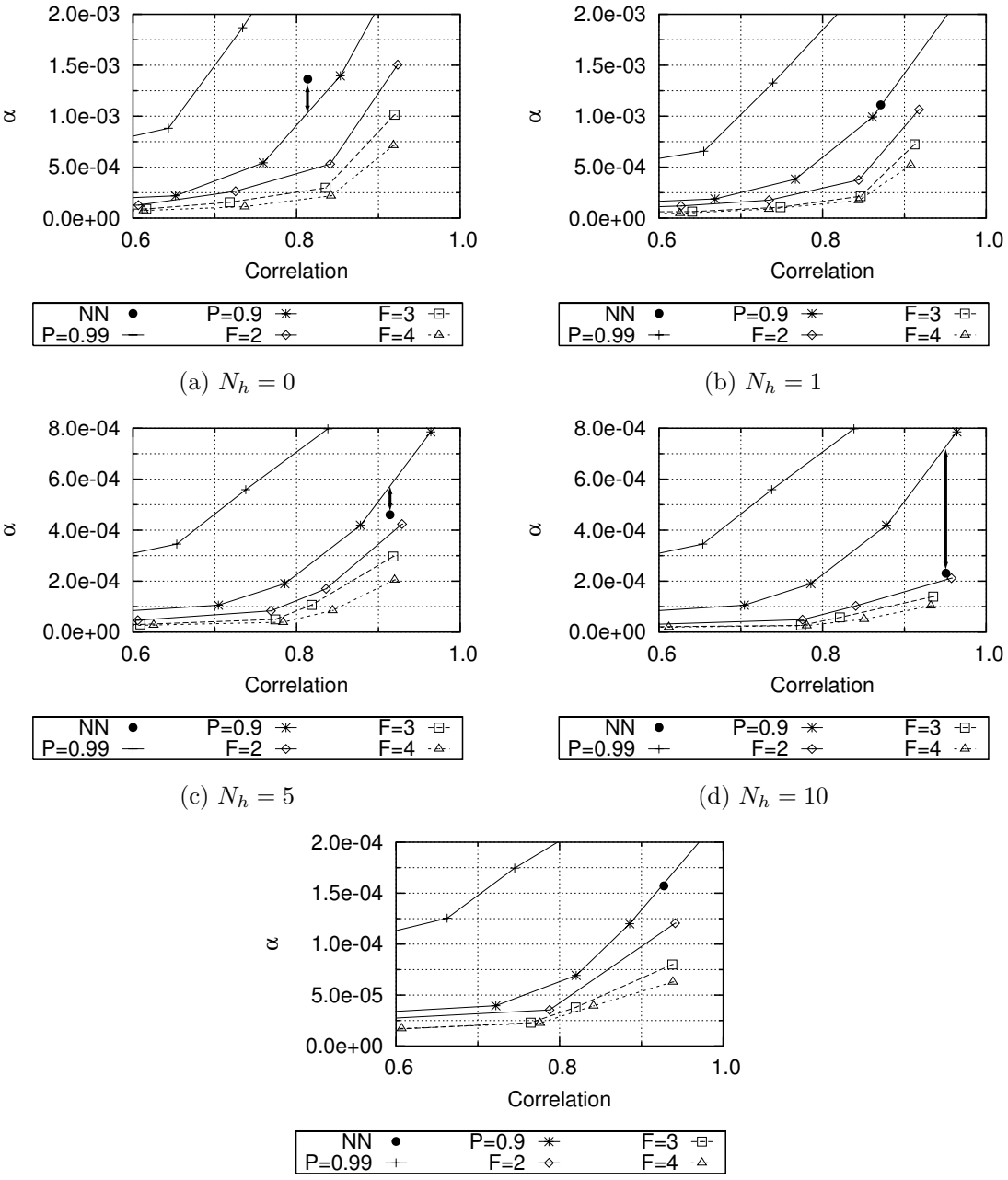


Figure 4.7: Correlation ($\dot{E}_b$) for each N_h at $q = 1/L$



(a) $N_h = 0$

(b) $N_h = 1$

(c) $N_h = 5$

(d) $N_h = 10$

(e) $N_h = 15$

Figure 4.8: α as a function of the correlation ($\dot{E}_b$) at $q = 1/L$

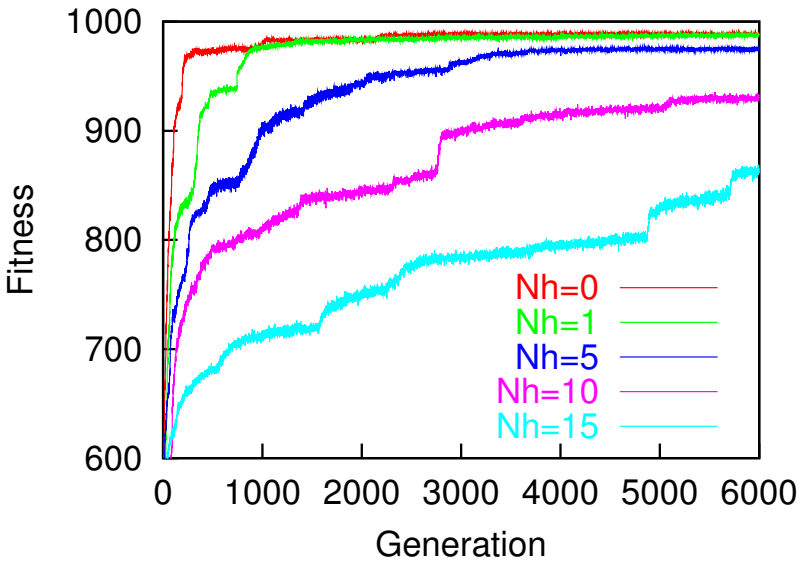


Figure 4.9: Maximum fitness at each generation at $q = 1/L$

4.2.2 Evolutionary Dynamics

Simulation Conditions

For this experiment, an agent's controller was a PNN with 7 sensory neurons, 2 fully interconnected motor neurons and N_h fully interconnected hidden neurons, where $N_h \in \{1, 10\}$. The network's connection weights and the firing threshold for each neuron were genetically encoded and evolved. The total number of parameters was either 33 ($N_h = 1$) or 240 ($N_h = 10$). The parameters were mapped linearly onto the following ranges: connection weights, $\omega \in [-1.0, 1.0]$, and thresholds, $\theta \in [0.0, 3.9]$. The parameters of the neurons and synapses (see Appendix D) were set as follows: $\tau_m = 4$, $\tau_s = 10$, $\Delta^{ax} = 2$ for all neurons and all synapses in the network following the recommendations given in (Floreano and Mattiussi, 2001).

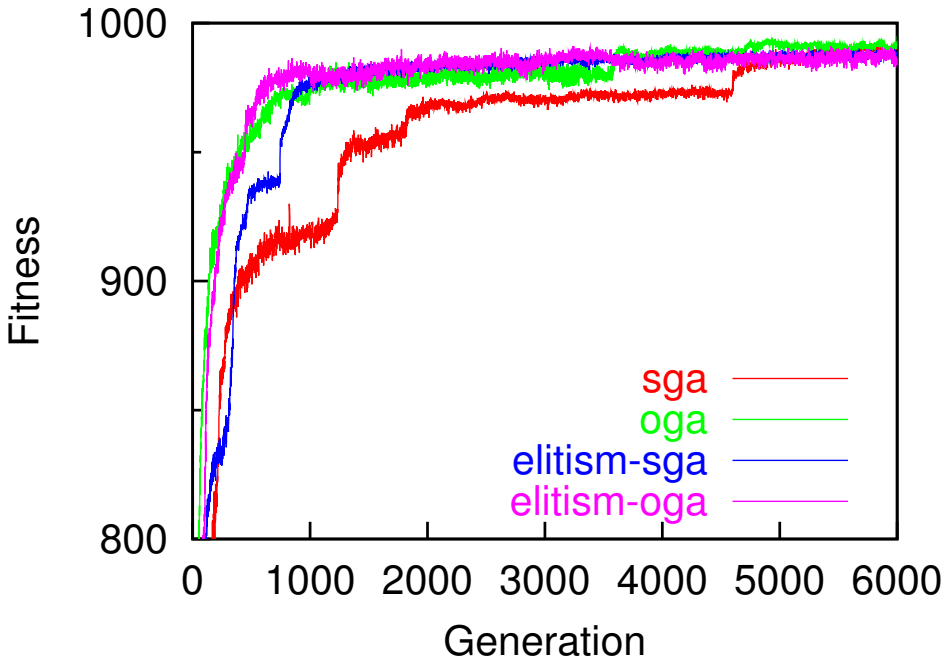
Computer simulations were conducted using populations of size 50. Each individual was encoded as a binary string with 10 bits for each parameter. Therefore, the total length of the genotype was either $L_1 = 330$ ($N_h = 1$) or $L_{10} = 2400$ ($N_h = 10$). The SGA and the OGA were employed to evolve the PNN parameters. The OGA uses standard bit mutation and five additional genetic operators: *connection*, *division*, *duplication*, *deletion* and *inversion*. The probabilities for genetic operations were set at 0.3 for *connection* and *division*, 0.2 for *duplication* and 0.05 for *deletion* and *inversion*, based on the previous results in Chapter 2. The length of the value list in a locus was 6. The genetic operation for the SGA was standard bit mutation. For both GAs, the per-bit mutation rate, q, was set at $1/L$ (0.003 for L_1 and 0.000416 for L_{10}). Crossover was not used for either GA, following Nimwegen's suggestion (van Nimwegen et al., 1999). Tournament selection was adopted. Elitism was optionally applied. The tournament size, s, was set at $\{2, 6\}$ because the SGA prefers low selection pressure while the OGA prefers high selection pressure. A generational model was used. Each run lasted 6,000 generations. 10 independent runs were conducted for each of the sixteen conditions (i.e., with each GA, with and without elitism, with each tournament size, and for each of the two PNN configurations). All results were averaged over 10 runs.

Simulation Results

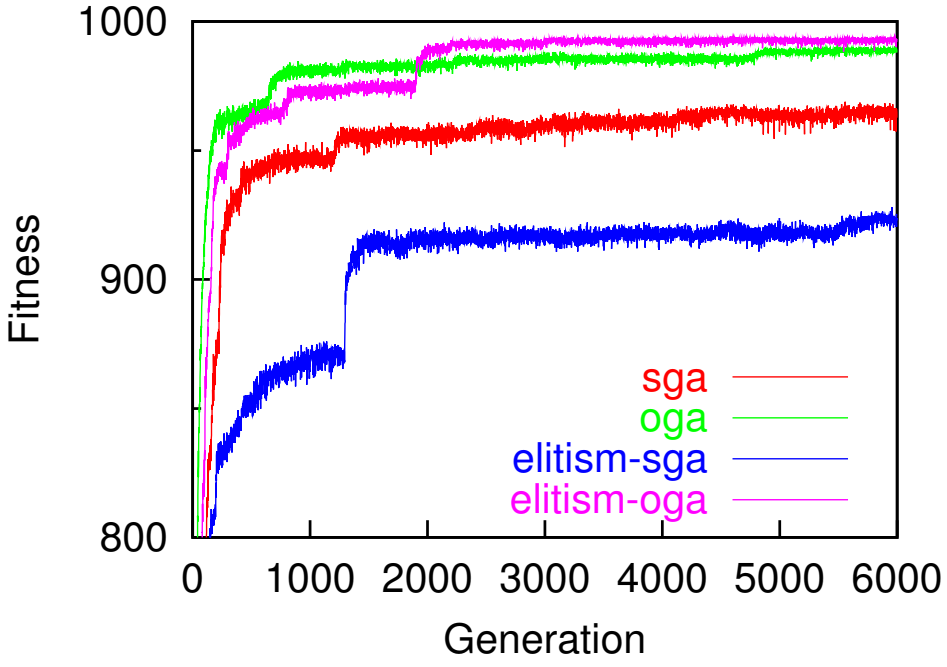
Fig. 4.10 shows the maximum fitness at each generation for the SGA and OGA, with and without elitism, for controllers with $N_h = 1$. Fig. 4.10(a) and 4.10(b) show the results for the four GA conditions for $s = 2$ and 6, respectively. For $s = 2$, fitness increased faster with the OGA than with the SGA in the early generations. In the final generation, there was no significant difference between the SGA and the OGA. For $s = 6$, the SGA was trapped on local optima, whereas the OGA continued to find better regions of the search space. In addition, the SGA performed better without elitism than with it. These results are consistent with the results obtained using terraced NK landscapes in Section 2.3. With respect to final generation fitnesses, there was no significant difference between the SGA with $s = 2$ and the OGA for $s = 6$. However, a closer examination reveals that during the process of evolution the OGA with $s = 6$ performed better than the SGA with $s = 2$ and elitism (Fig. 4.11).

Fig. 4.12 shows the maximum fitness at each generation for the SGA and OGA, with and without elitism, for $s = 2$ and 6 with $N_h = 10$. With $N_h = 10$, differences between the SGA and the OGA were much more pronounced than with $N_h = 1$. Even for $s = 2$, fitness increased faster for the OGA than for the SGA (Fig. 4.12(a)). This is consistent with the results obtained using the Balance Beam Function when the mutation rate is below the optimal mutation rate in Section 2.2. As with $N_h = 1$, for $s = 6$, the SGA with elitism was trapped on local optima (Fig. 4.12(b)), whereas the OGA continued to find better regions; also as before, the SGA performed better without elitism than with it. The OGA for $s = 6$ also outperformed the SGA for $s = 2$ (Fig. 4.13).

Under all conditions, the OGA performed better than the SGA on this task, either by achieving higher final fitnesses, or by achieving high fitnesses faster, or both. This shows that the OGA's variable mutation rate strategy was beneficial on this problem.



(a) $s = 2$



(b) $s = 6$

Figure 4.10: Maximum fitness at each generation for $N_h = 1$

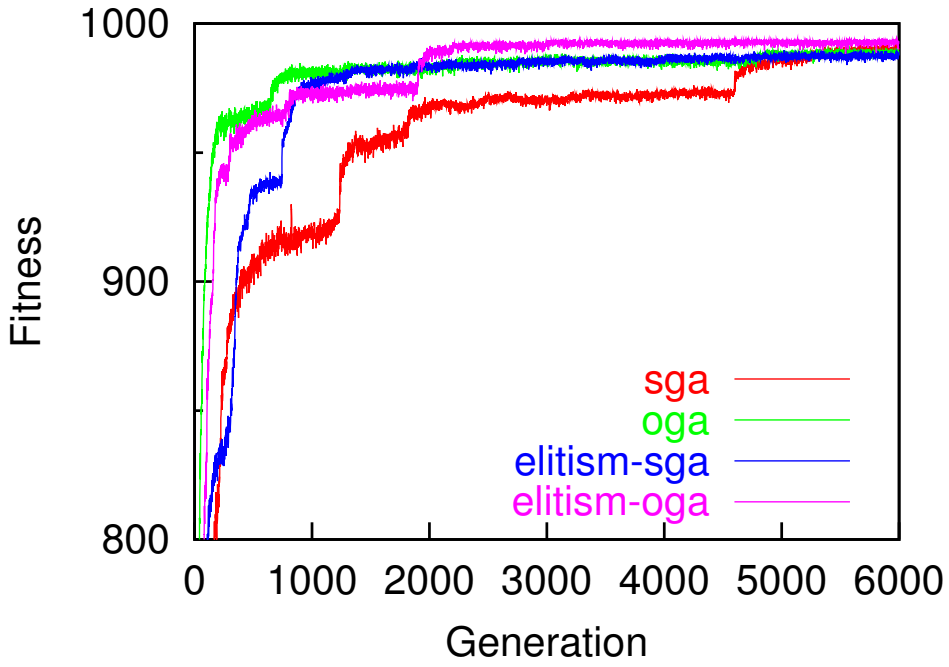
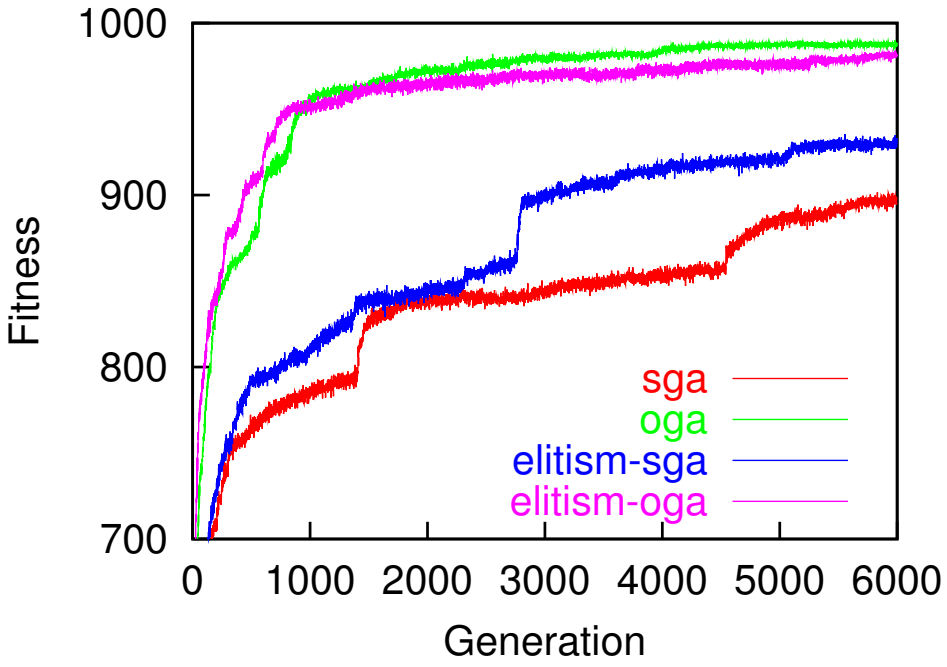
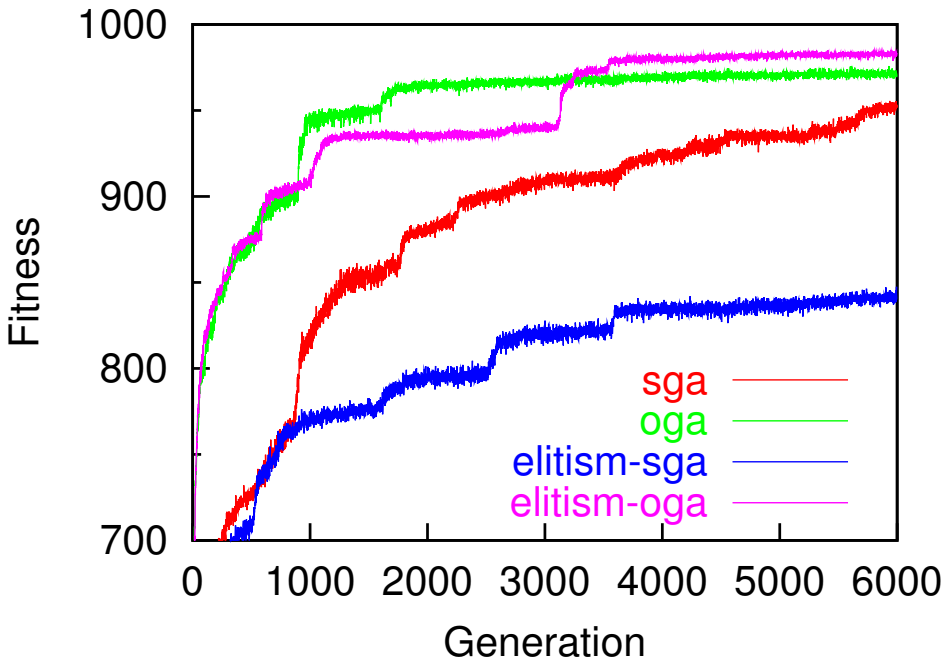


Figure 4.11: Comparison between the SGA for $s = 2$ and the OGA for $s = 6$ for $N_h = 1$



(a) $s = 2$



(b) $s = 6$

Figure 4.12: Maximum fitness at each generation for $N_h = 10$

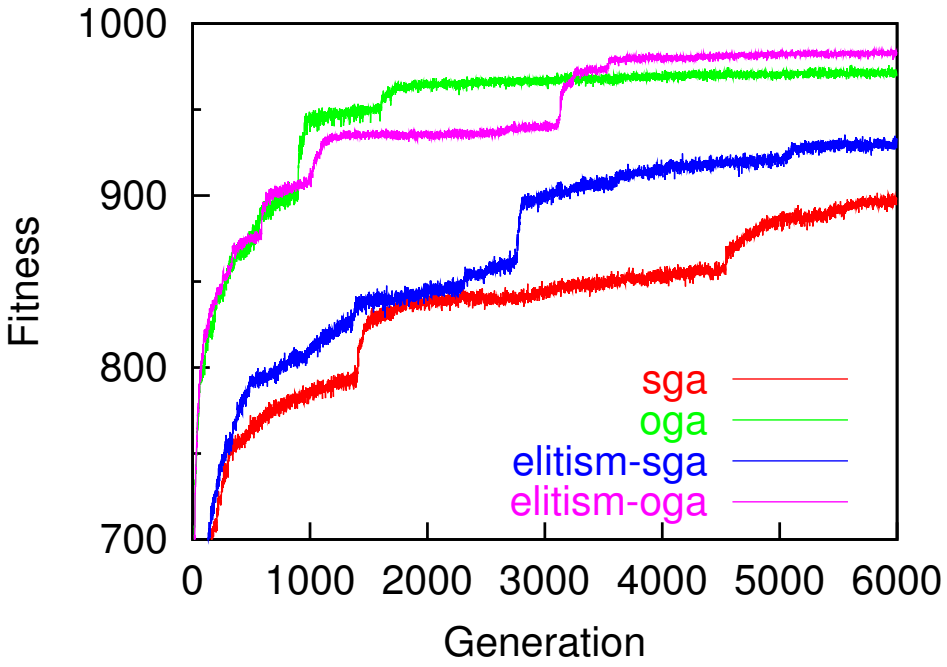


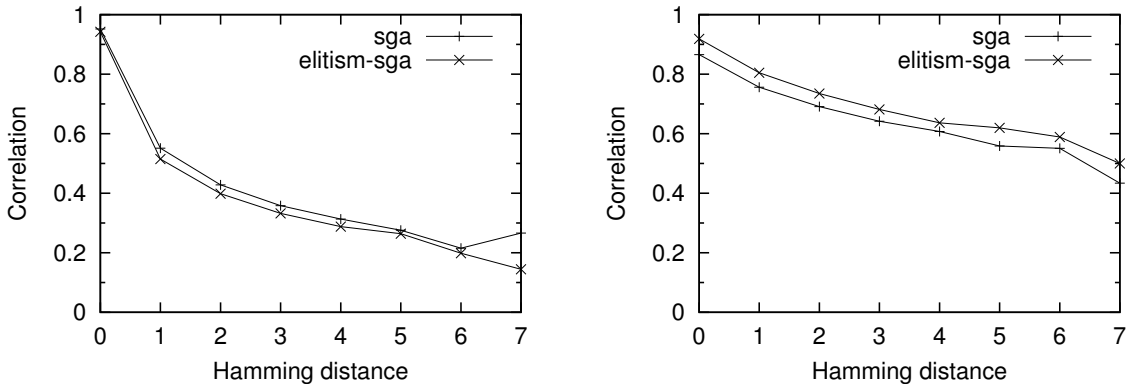
Figure 4.13: Comparison between the SGA for $s = 2$ and the OGA for $s = 6$ for $N_h = 10$

Discussion

The evolutionary dynamics observed in this experiment can be explained in the same way as in Chapter 2.

The evolutionary dynamics that were observed showed phases of neutral evolution for each run (Appendix E) and the average one (Fig. 4.10 and 4.12), implying that the fitness landscapes include neutral networks. However, large fluctuations that sometimes cause the best individuals to be lost were not observed under any of the four GA conditions. That is, there was no influence of the error threshold at the mutation rate $q = 1/L$. Therefore, it can be assumed that the effective mutation rate of $q = 1/L$ was below the error threshold under each condition.

The correlation of the landscapes was analyzed in order to investigate overall GA performance. It has been shown in Fig. 4.7 that the landscape with $N_h = 10$ was more correlated than the one with $N_h = 1$. In addition to this, Fig. 4.14(a) and 4.14(b) show the larger correlation coefficients as a function of the Hamming distance between parents and offspring (Manderick et al., 1991) for the SGA, with and without elitism, with $N_h = 10$ than those with $N_h = 1$. They suggest high fitness correlation in both landscapes, with the $N_h = 10$ landscape being more highly correlated that the $N_h = 1$ landscape.



(a) $N_h = 1$ (b) $N_h = 10$

Figure 4.14: Correlation coefficient as a function of the Hamming distance between parents and offspring for the SGA

As predicted, with $s = 6$, the SGA with and without elitism was trapped on local optima when $N_h = 1$, due to the low mutation rate and high selection pressure. With $s = 6$ and $N_h = 10$, the SGA with elitism was also trapped on local optima. However, the SGA without elitism for $s = 6$ and $N_h = 10$ was not obviously trapped. Based on the analysis of ruggedness shown in Fig. 4.14(b), it seems likely that fitnesses would continue improving if the runs were extended beyond their final generation. Further computer simulations were therefore conducted in order to observe the SGA runs over additional 4,000 generations. Fig. 4.15 shows the maximum fitness for 10,000 generations of the SGA with $N_h = 10$. The SGA without elitism continued to find better regions of the search space. This indicates that the SGA without elitism can escape from local optima with this level of ruggedness.

When compared on the same landscapes, the OGA continued to find much better regions of search space than the SGA. The continued improvement observed with the OGA was due to the online adaptation of mutation rates during the process of evolution. In addition to this, with the OGA, the effective mutation rate will have been below the error threshold even with low selection pressure (i.e. when $s = 2$). This is why the variable mutation rate strategy of the OGA was a better approach on this problem with both high and low selection pressure.

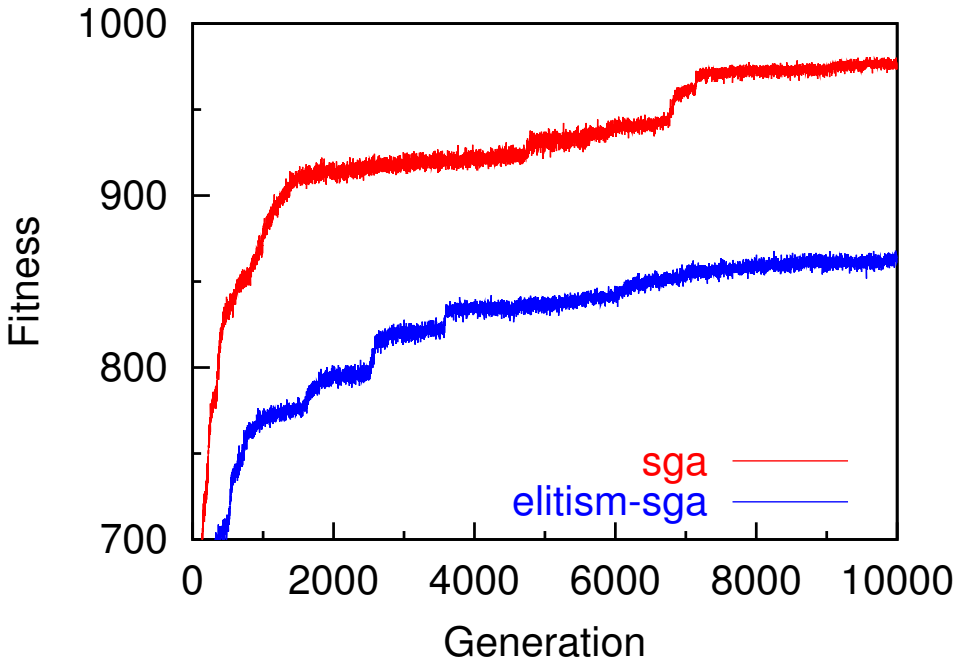
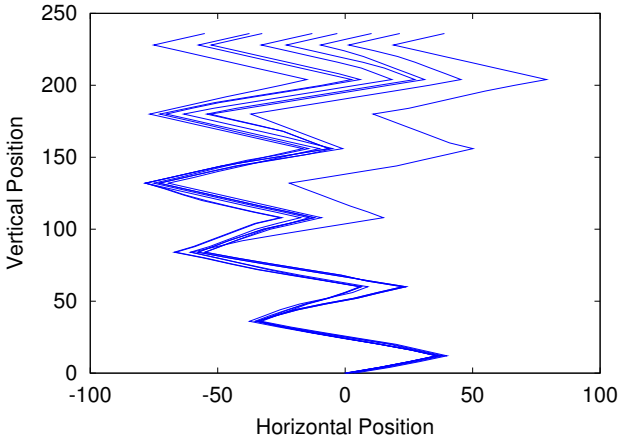


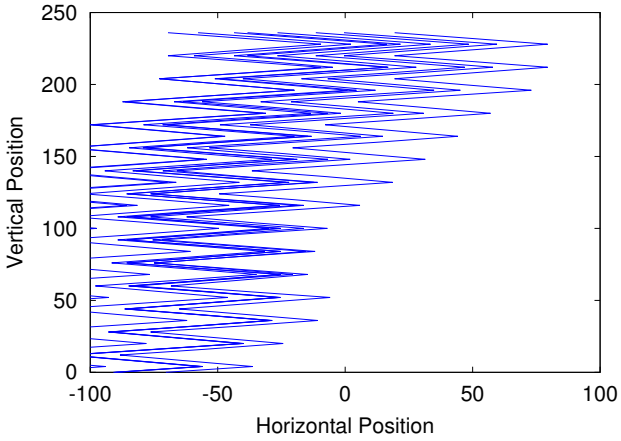
Figure 4.15: Maximum fitness over 10,000 generations for the SGA for $s = 6$ and $N_h = 10$

Agent's Behavior

Fig. 4.16 shows typical behaviors of the best evolved agent for $N_h = 10$ by the SGA with elitism for $s = 2$. Fig. 4.16(a) and 4.16(b) show the motion trajectories relative to the agent of objects falling vertically with the long period and the short period from several different initial horizontal offsets, respectively. It is found that the best agent can discriminate between the long and short period on all the trials.



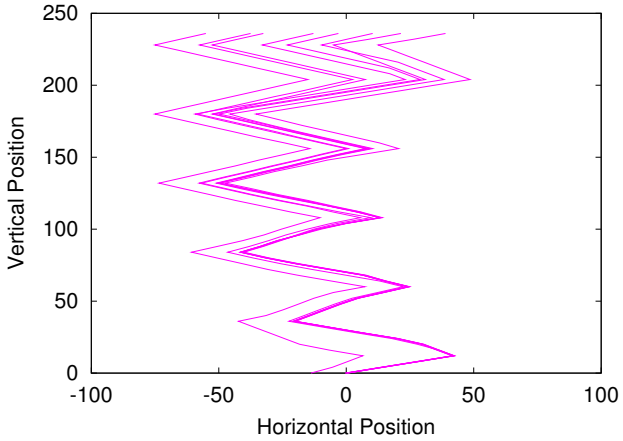
(a) For long period



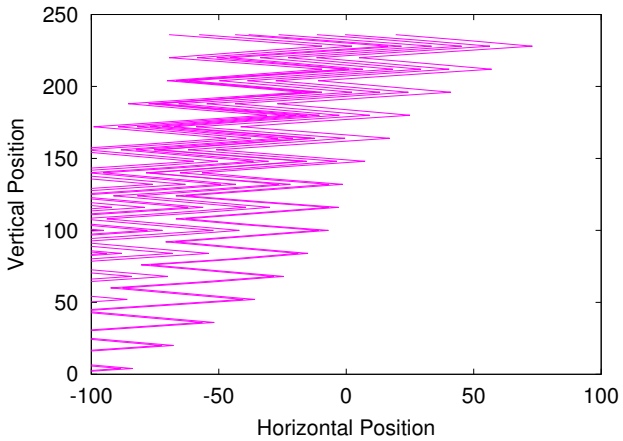
(b) For short period

Figure 4.16: Behaviors of the genetically determined controller by the SGA with elitism for $N_h = 10$

Fig. 4.17 shows typical behaviors of the best evolved agent for $N_h = 10$ by the OGA with elitism for $s = 6$. It is also found that the best agent can discriminate between both the long and short period on all the trials. The best evolved agent by the OGA can discriminate between both the periods faster on each trial than the one by the SGA. That is, the best evolved agent by the OGA can avoid the object at a further distance, then get a higher fitness value.



(a) For long period



(b) For short period

Figure 4.17: Behaviors of the genetically determined controller by the OGA with elitism for $N_h = 10$

4.3 Pursuit Problem

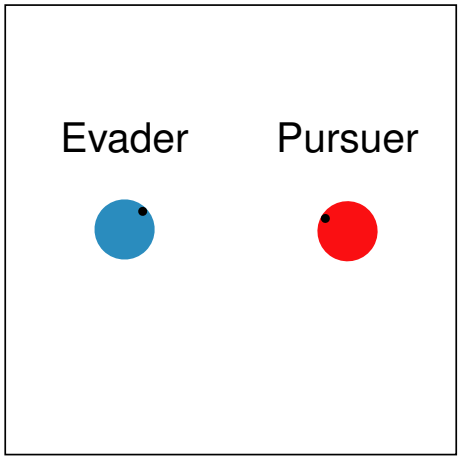
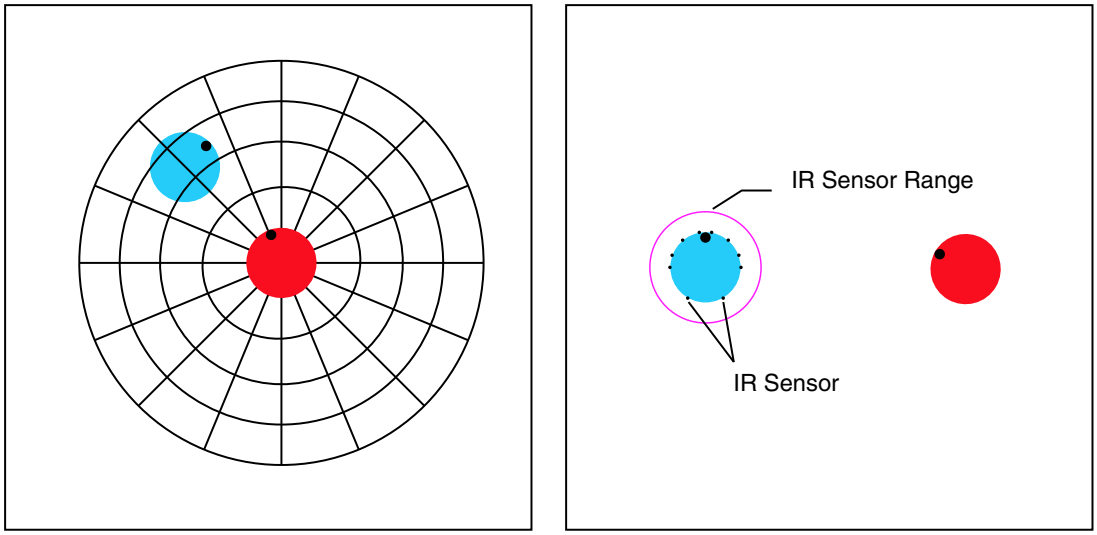


Figure 4.18: Experimental setup for a pursuit problem. The initial starting positions for the pursuer and the evader are given in the arena. The initial orientation is set at random.

In the previous section, it has been shown that the fitness landscape of the discrimination problem is relatively smooth. In this section, a pursuit problem is studied, which is expected to have more rugged landscapes than the discrimination problem (will be described in Section 4.3.1). The simulated environment is shown in Fig. 4.18, where a pursuer seeks to hit (capture) an evader. The pursuit problem is often investigated in embodied cognitive science (Pfeifer and Scheier, 1999; Nolfi and Floreano, 2000) as a metaphor for predator-prey. Generally, predators and preys are set belonging to different species which have different sensors and motors. Following this setting, two kinds of agent were employed, one (the pursuer) is equipped with an omni-directional vision (Fig. 4.19(a)) while the other (the evader) is equipped with 10 infrared proximity sensors (eight on the front side and two on the back in Fig. 4.19(b)). The pursuer's behavior was controlled by a PNN. On the other hand, the evader's one was controlled by "Braitenberg vehicle(Braitenberg, 1996)" in which the type 2a and 3b are combined for evasions (Fig. 4.20).



(a) Omni-directional sensor for a pursuer (b) IR sensors for an evader

Figure 4.19: Simulated models of sensors for each agent

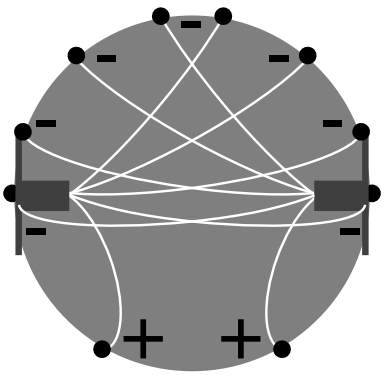


Figure 4.20: Architecture of a Braintenberg vehicle

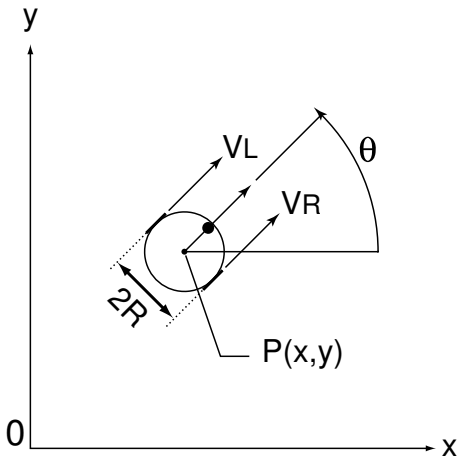


Figure 4.21: Model of a mobile robot

Employing a mathematical model of a mobile robot (Fig. 4.21), the displacement of the agent was computed as follows:

$$\begin{aligned} x_{k+1} &= x_k + \frac{V_R + V_L}{2}\cos\theta_k + \omega_{xk+1} \\ y_{k+1} &= y_k + \frac{V_R + V_L}{2}\sin\theta_k + \omega_{yk+1} \\ \theta_{k+1} &= \theta_k + \frac{V_R - V_L}{2R} + \omega_{\theta k+1}, \end{aligned} \tag{4.2}$$

where V_R and V_L are the velocities applied to the right and left wheel respectively, R is the radius of an agent, $2R$ is the interval between the wheels and $\omega_{[.]k}$ is the system error. It has been known that the system error in Equation 4.2 is modeled as the normal distribution $N(0, \sigma)$ with mean zero and standard deviation, σ (Komoriya et al., 1993). For this experiment, σ was set at the 10 percent of the displacement at the current step.

For this experiment, an omni-directional vision was assumed to be used for the sensory inputs of the pursuer (Fig. 4.19(a)). The simulated model of an omni-directional vision[1] were computed as follows. Assuming the binarized image taken from the omni-directional image, the direction of the center of gravity of the closest object around

[1]The algorithm for detecting the presence of an object by a real omni-directional camera is referred in Section 4.4.

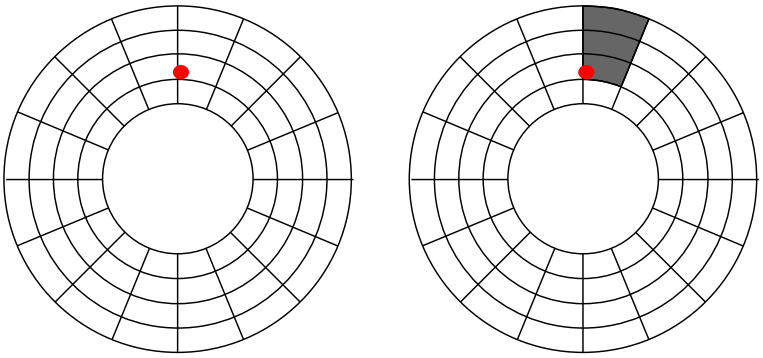


Figure 4.22: Simulated omni-directional image plane

the pursuer and the nearest point of the object to the pursuer are first calculated, where the center of the image is identified with the center of the pursuer (Fig. 4.22). Next, the image is divided into I individual cells in a circumferential direction and J individual cells in a radial direction. Then, the presence of the object is judged for each cell ($q_{ij} = \{0, 1\}, i = 1, \ldots, I; j = 1, \ldots, J$) by using the above two values, where q_{ij} signifies the presence or absence of the object. According to the characteristic of the omni-directional image, if the presence of an object whose height is almost the same as the one of the camera is detected in the j-th cell, the one is also detected in the $(j + 1)$-th cell (Right in Fig. 4.22). In this simulation, the pursuer is not able to detect walls.

The evader was provided with 10 infrared proximity sensors which have a maximum detection range in the environment (Fig. 4.19(b)). If a pursuer or a wall intersects a proximity sensor, the sensor outputs a value inversely proportional to the distance between the pursuer and the sensor.

At the beginning of each trial, the pursuer and evader were always positioned on a horizontal line at the random orientation in the middle of the environment at a distance corresponding to half the environment width (Fig. 4.18). One trial ended either when the pursuer hits the evader or when 400 steps are performed without the hit. Based on the fitness function used by Floreano and Nolfi (1997), the performance

measure to be maximized was as follows:

$$Fitness = \frac{1}{NumTrials} \sum_{i=1}^{NumTrials} (1 - \frac{Step}{MaxStep}) \tag{4.3}$$

where $NumTrials$ is the number of trials for an individual (pursuer) (10 trials for each individual) and $MaxStep$ is set at 400. The fitness function increases as the pursuer catches the evader more quickly.

4.3.1 Describing the Fitness Landscape

Simulation Conditions

The predator controller was a PNN with 64 sensory neurons corresponding to the number of cells $I \times J$ (Fig. 4.22), 2 fully interconnected motor neurons and 3 fully interconnected hidden neurons. The network's connection weights and the firing threshold for each neuron were genetically encoded and evolved. The total number of parameters was equal to 350. The other parameters of the neurons and synapses were the same as in Section 4.2.2.

Computer simulations were conducted using populations of size 50. Each individual was encoded as a binary string with 10 bits for each parameter. Therefore, the total length of the genotype was $L = 3500$. The SGA were adopted to evolve PNN parameters. The genetic operation for the SGA was standard bit mutation. The genetic distance was used for estimating the degree of neutrality in the fitness landscape. Based on the assumption of Equation 3.3 in Chapter 3, the per-bit mutation rate, q, was set at $1/L = 0.000286$. Tournament selection was adopted. Elitism was applied. The tournament size was set at 2. A generational model was used. Each run lasted 6,000 generations. 10 independent runs were conducted for each PNN controller.

The features of the TNK(p) landscapes for $N = 20, K \in \{0, 2, 6, 12, 19\}, F = 2, (P = 0.99)$, which were obtained in the complementary computer simulations with the same parameters of the SGA as those of the EANNs, were used as baselines for estimating indirectly the features of the EANNs in the same way as in Section 4.2.1.

Simulation Results

Fig. 4.23 shows the genetic distance at each generation. The approximately linear increases were observed in all runs. This indicates the presence of neutrality in the fitness landscape of the EANN. Thus, α was calculated by using the method of least squares on the results of all the runs in the following parts.

Neutrality of the EANN was estimated indirectly by using the $\dot{E}_b$-α curves in the TNK(p) landscape (F, P are constant) as baselines (Fig. 4.24). By using the curve for $P = 0.99$ as a baseline, it was observed that neutrality of this landscape was much larger than those of the landscapes in the discrimination problem in Section 4.2.1 as well as larger than those of the TNK(p) for $P = 0.99$. Note that the TNK(p) was defined in order to increase neutrality in the TNK (Section 4.2.1). This means that the landscape of this problem includes relatively large neutrality. In addition, the fitness landscape was more rugged than those of the discrimination problem as predicted in the beginning of this section.

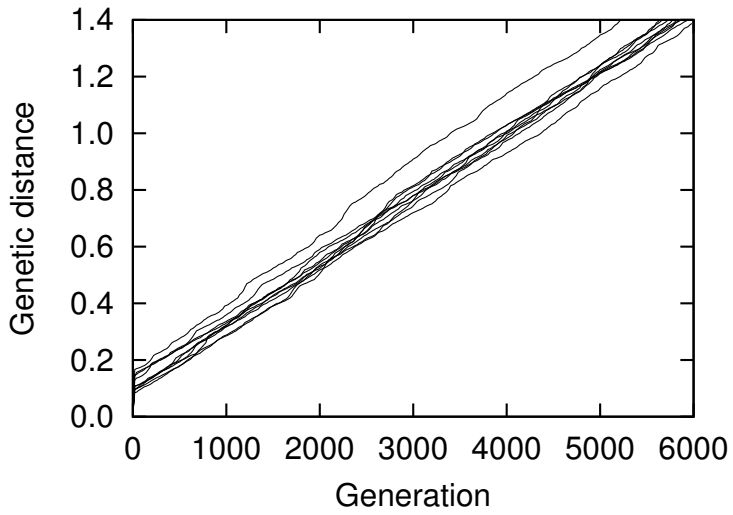


Figure 4.23: Genetic distance at each generation for the SGA at $q = 1/L$ in 10 runs

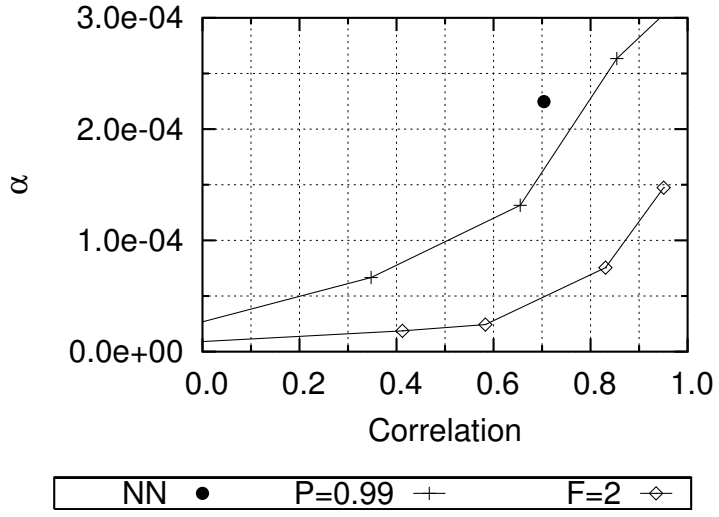


Figure 4.24: α as a function of the correlation ($\dot{E}_b$)

4.3.2 Evolutionary Dynamics

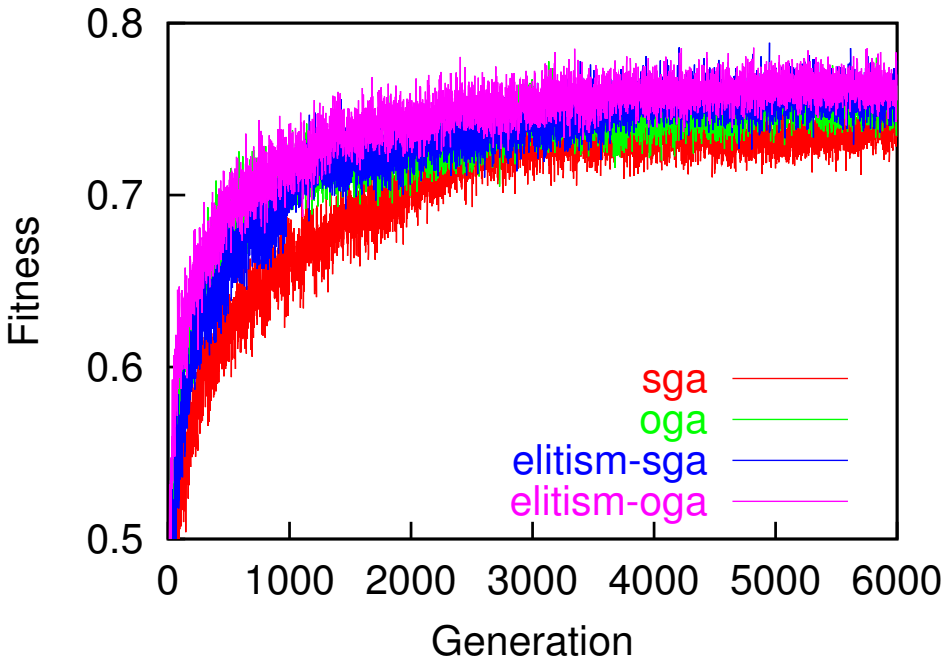
Simulation Conditions

The parameters of a PNN were the same as in the previous section 4.3.1. Computer simulations were conducted using populations of size 50. Each individual was encoded as a binary string with 10 bits for each parameter. Therefore, the total length of the genotype was $L = 3500$. The SGA and the OGA were employed to evolve the PNN parameters. The OGA uses the same genetic operators as in Section 2.3: *connection*, *division*, *duplication*, *deletion* and *inversion*. The probabilities for genetic operations were set at 0.3 for *connection* and *division*, 0.6 for *duplication* and 0.3 for *deletion* and *inversion* (in the case of $s = 6$ with elitism), and 0.3 for *connection* and *division*, 0.2 for *duplication* and 0.05 for *deletion* and *inversion* (in the other cases), based on the previous results in Section 2.3 due to the rugged landscape described in Section 4.3.1. The length of the value list in a locus was 6. The genetic operation for the SGA was standard bit mutation. For both GAs, the per-bit mutation rate, q, was set at $1/L = 0.000286$. Tournament selection was adopted. Elitism was optionally applied. The tournament size s was set at $\{2, 6\}$. A generational model was used. Each run lasted 6,000 generations. 10 independent runs were conducted for each of the eight conditions. All results were averaged over 10 runs.

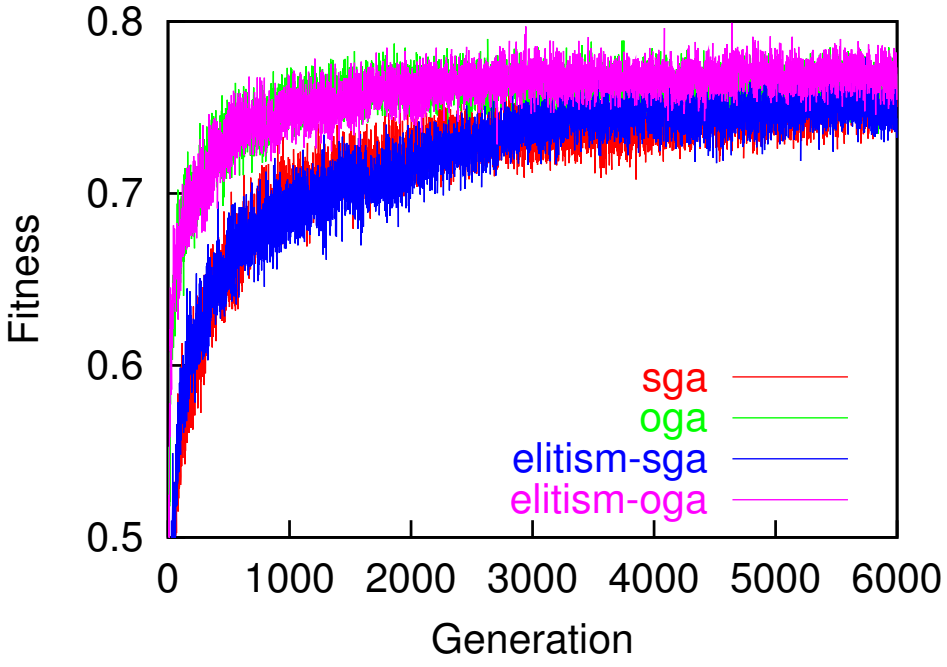
Simulation Results

Fig. 4.25 shows the maximum fitness at each generation for the SGA and OGA, with and without elitism. Note that the transition of the average maximum fitness did not clearly show phases of neutral evolution, compared with the results obtained in Section 4.2.2. However, phases of neutral evolution were found for each run (Appendix F). In Section 4.3.1, it was confirmed that the landscape of this problem includes relatively large neutrality. These indicate that the average maximum fitness at each generation does not necessarily show phases of neutral evolution even in fitness landscapes with neutrality.

Fig. 4.25(a) and 4.25(b) show the results for the four GA conditions for $s = 2$ and 6, respectively. For $s = 2$, fitness increased faster for the OGA than for the SGA



(a) $s = 2$



(b) $s = 6$

Figure 4.25: Maximum fitness at each generation

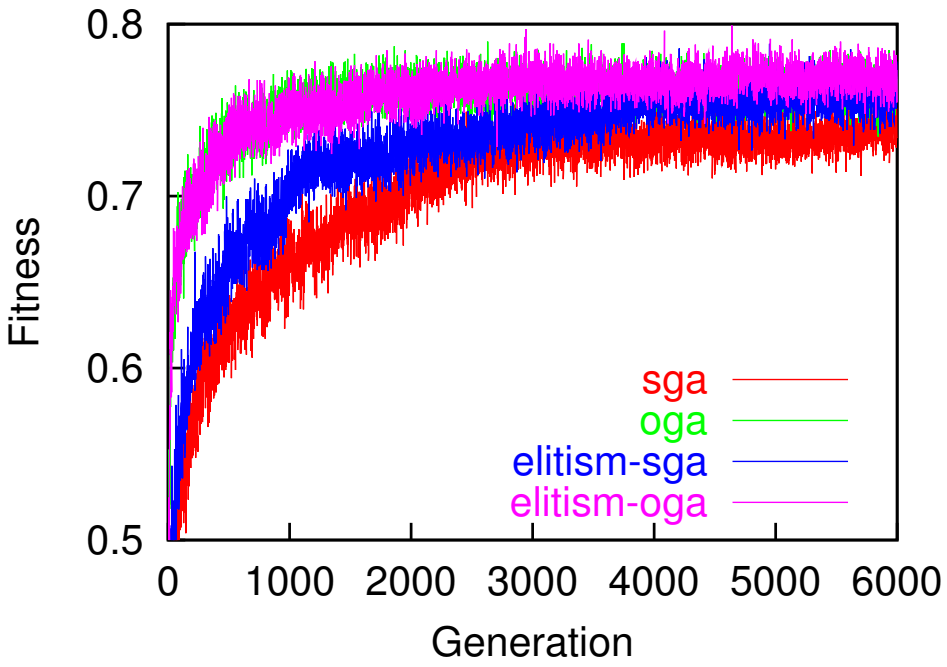


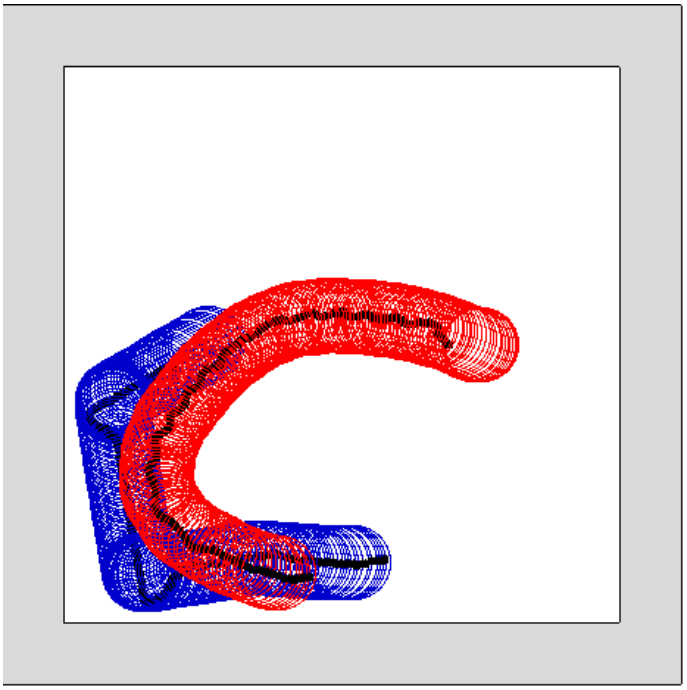
Figure 4.26: Comparison between the SGA for $s = 2$ and the OGA for $s = 6$

in the early generations. In the final generation, there was no significant difference between the SGA with elitism and the OGA with elitism. For $s = 6$, the SGA was trapped on local optima as predicted by the ruggedness in Section 4.3.1. In contrast, the OGA for $s = 6$ continued to find better regions of the search space. Note that the probabilities for genetic operations in the OGA with elitism for $s = 6$ were set at 0.3 for *connection* and *division*, 0.6 for *duplication* and 0.3 for *deletion* and *inversion*. In the preliminary runs, it has been confirmed that the OGA with elitism for $s = 6$ setting the probabilities for genetic operations at 0.3 for *connection* and *division*, 0.2 for *duplication* and 0.05 for *deletion* and *inversion* was outperformed by the SGA. These are consistent with the results obtained using terraced NK landscapes in Section 2.3. The OGA for $s = 6$ also outperformed the SGA for $s = 2$ in Fig. 4.26. These results show that the OGA's variable mutation rate strategy with the probabilities for genetic operations tuned for rugged landscapes with high selection pressure is also beneficial to this problem.

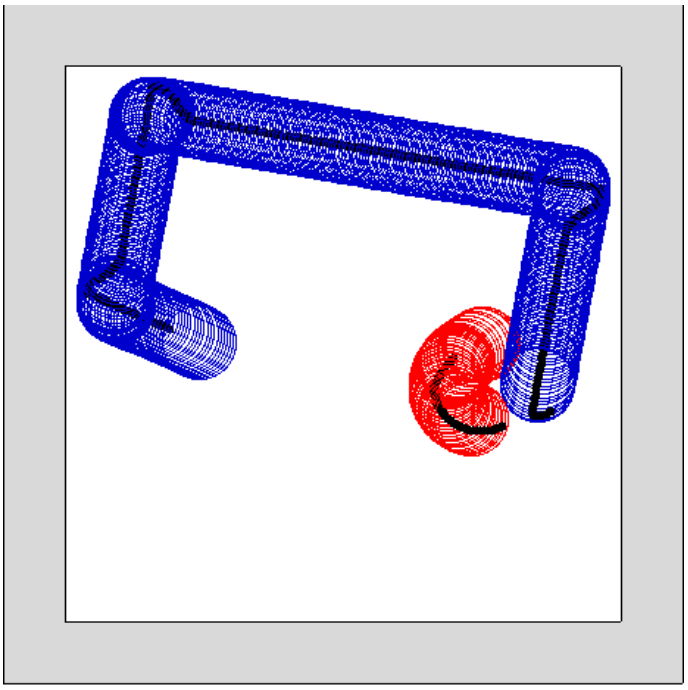
Agent's Behavior

Fig. 4.27 shows the typical behaviors of the best evolved agent by the SGA with elitism for $s = 2$. They show approximately two kinds of behavior: One is positive where the pursuer approaches to the evader as soon as the trial begins, follows the evader and then catches it. The other is negative where the pursuer stays around the initial position waiting for the evader approaching to the pursuer for a while and then catches it as soon as the evader enters the range in which the pursuer can catch the evader at the current speed.

Fig. 4.28 shows the typical behaviors of the best evolved agent by the OGA with elitism for $s = 6$. They show approximately a single behavior, positive one. This can be explained as follows. The fitness function employed in this experiment increases as a pursuer catches an evader more quickly. Thus, the negative behavior is not preferable even in the case that a pursuer can catch an evader because the negative one is time-consuming. This is why the positive behavior would become well suited in the process of evolution. Note that the OGA for $s = 6$ outperformed the SGA for $s = 2$ in Fig. 4.26. As a result, the positive behaviors would be selected more than the negative behaviors.

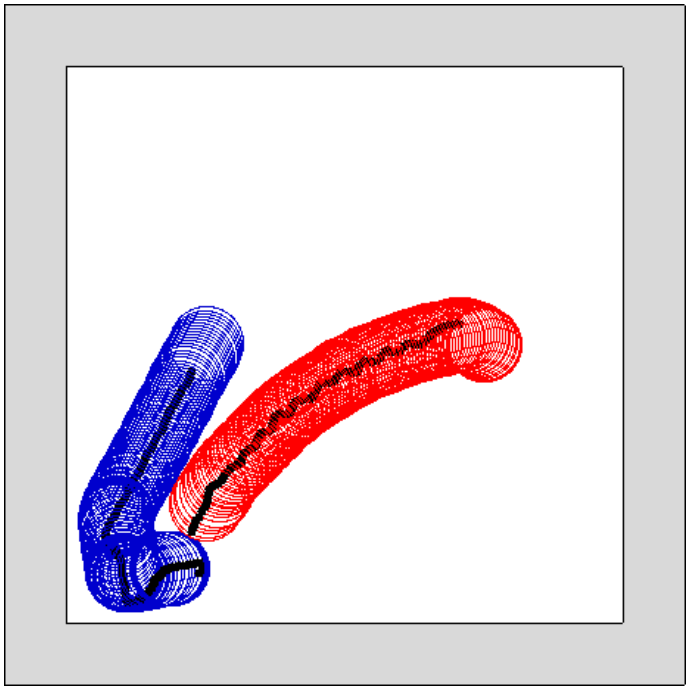


(a) Positive behavior

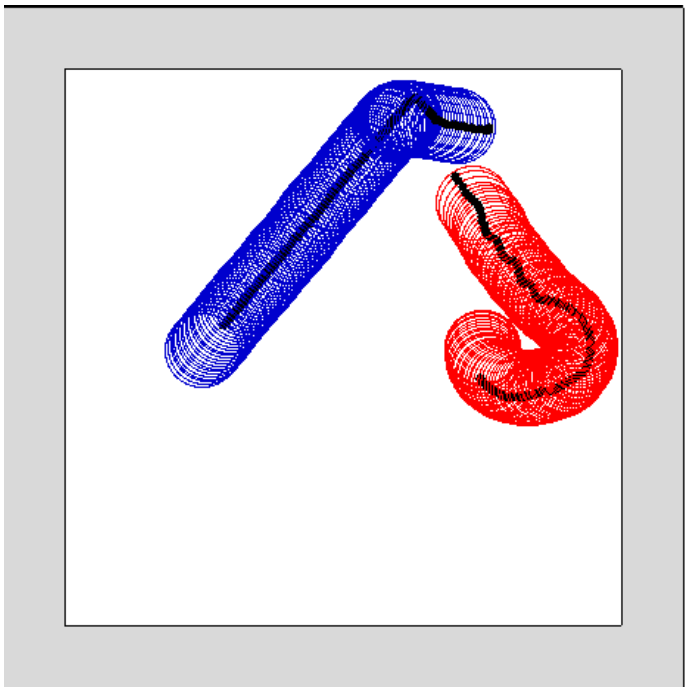


(b) Negative behavior

Figure 4.27: Behaviors of the genetically determined controller by the SGA with elitism: Red is pursuer, blue is evader



(a) Positive behavior 1



(b) Positive behavior 2

Figure 4.28: Behaviors of the genetically determined controller by the OGA with elitism: Red is pursuer, blue is evader

4.4 Goal Reach Problem

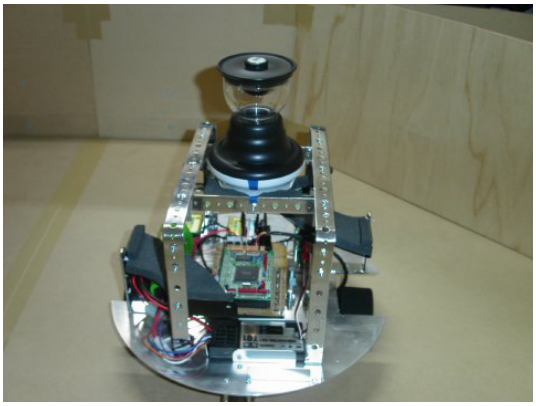
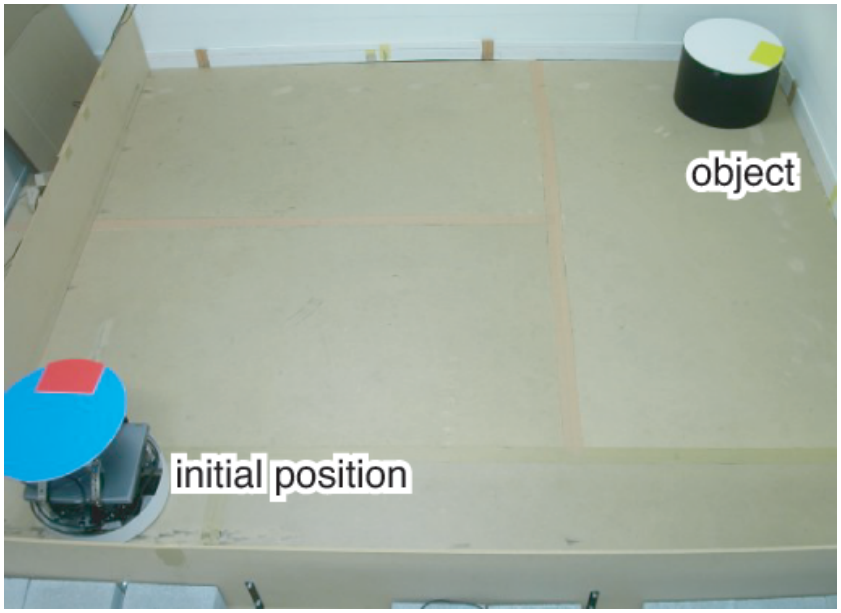


Figure 4.29: Mobile robot with omni-directional camera

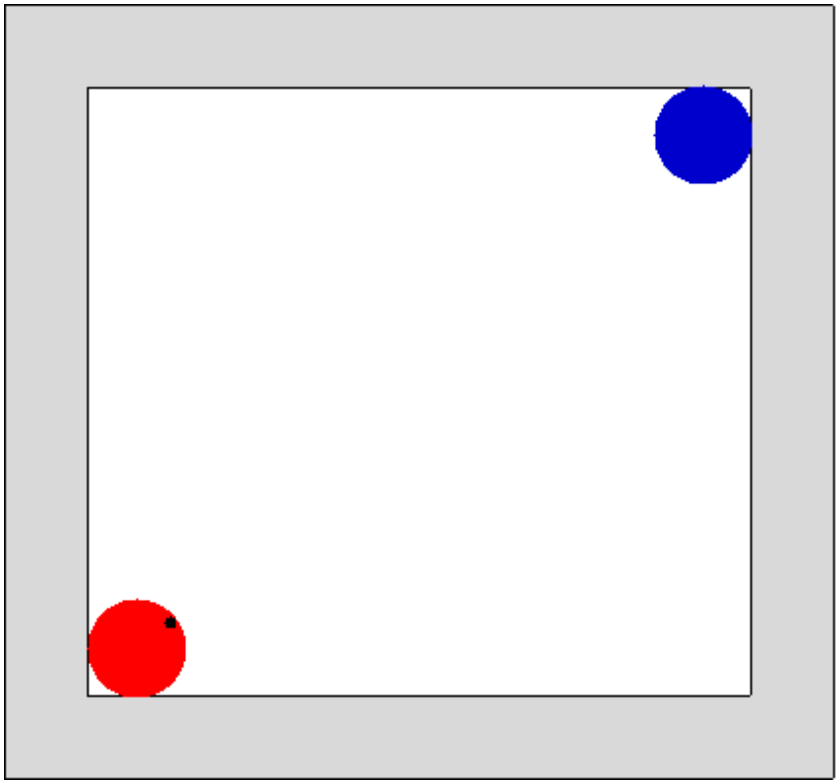
In this section, a goal reach problem is studied. The real robot used in this experiment (Fig. 4.29) was equipped with an omni-directional camera (see the details in Appendix G). The real and simulated environments are shown in Fig. 4.30, where a robot approaches to an object (goal). In the following part, first, the algorithm for detecting the presence of an object around the robot is described. Second, the methodology to model the real environment is explained.

The binarized image is taken from the omni-directional image. In an omni-directional image, the direction of the center of gravity of the closest object around the robot and the nearest point of the object to the robot are first calculated by using the pixel informations, where the center of the image is identified with the center of the robot (Fig. 4.31(a)). Next, the image is divided into I individual cells in a circumferential direction and J individual cells in a radial direction (Fig. 4.31(b)). Then, the presence of the object is detected for each cell ($q_{ij} = \{0, 1\}, i = 1, \ldots, I; j = 1, \ldots, J$) by using the above two values, where q_{ij} signifies the presence or absence of the object. According to the characteristic of the omni-directional image, if the presence of the object whose height is almost the same as the one of the camera is detected in the j-th cell, the one is also detected in the $(j + 1)$-th cell. In the experiment, the robot is not able to see walls.

The robot's controller was a PNN with 64 sensory neurons, 2 fully interconnected

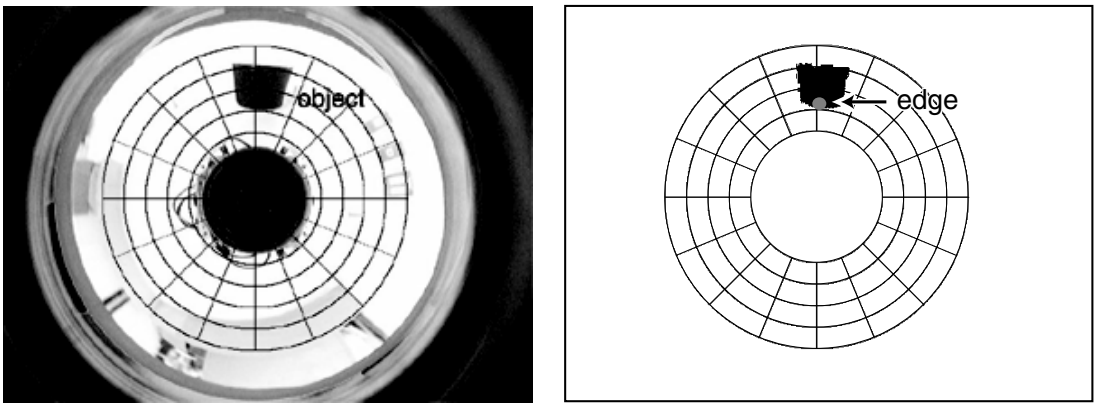


(a) Real environment



(b) Simulated environment

Figure 4.30: Experimental setup for a goal reach problem

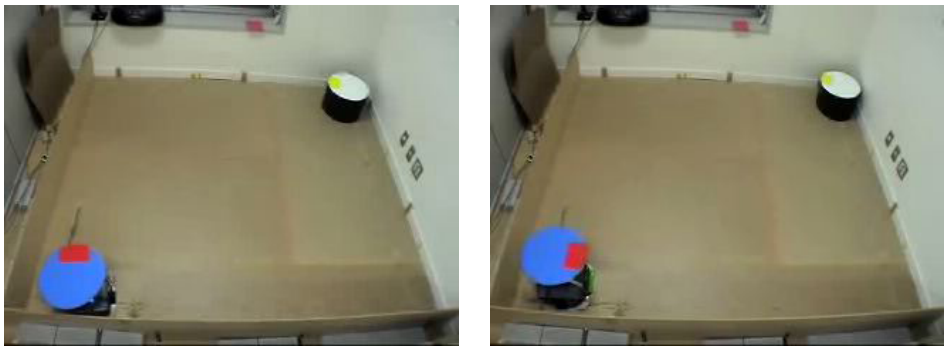


(a) Camera image (b) Binarized image

Figure 4.31: Omni-directional image plane

motor neurons and one fully interconnected hidden neurons. The controllers are evolved in simulation, before being transferred to the real robot. It has been shown that for a mobile robot with two wheels in flat environments there can be small discrepancies between its behaviors in a simulated environment and those in the real environment by sampling motor outputs and the displacements of the robot. Thus, this methodology were employed in this experiment as follows; The motor outputs and the displacements of the robot are sampled in advance without the case of collisions with walls (If the simulated robot attempts to hit against a wall, its position and orientation are not updated.). These values are then stored as the sampled data for probabilistic models, the normal distribution given by Equation 4.2 in Section 4.3, and accessed through a look-up table depending on the motor signal whenever the displacement of the robot is calculated. Note that the simulator is based on real motor outputs sampled from the robot, not on a mathematical model employed in Section 4.3. The setting of simulated visual inputs are the same as in Section 4.3.

At the beginning of each trial, a robot is always positioned on the catercorner of an object located at a corner (Fig. 4.30), at the upward orientation or the rightward orientation (Fig. 4.32). One trial ends either when the robot reach the object or when 40 steps are performed without the goal. Based on the fitness function used by



(a) Upward (b) Rightward

Figure 4.32: Initial orientation for each trial

Floreano and Nolfi (1997), the performance measure to be maximized was as follows:

$$Fitness = \frac{1}{NumTrials} \sum_{i=1}^{NumTrials} (1 - \frac{Step}{MaxStep}) \tag{4.4}$$

where $NumTrials$ is the number of trials for a robot (2 trials for each robot) and $MaxStep$ is set at 40. The fitness function increases as the robot reaches the object more quickly.

Experimental Conditions

The parameters of a PNN were the same as in the previous section. Computer simulations were conducted using populations of size 50. Each individual was encoded as a binary string with 10 bits for each parameter. Therefore, the total length of the genotype was $L = 2040$. The SGA and the OGA were employed to evolve the PNN parameters. The OGA uses the same genetic operators as in Section 2.3: *connection*, *division*, *duplication*, *deletion* and *inversion*. The probabilities for genetic operations were set at 0.3 for *connection* and *division*, 0.6 for *duplication* and 0.3 for *deletion* and *inversion*, based on the previous results in Section 2.3 and the obtained features of this fitness landscape (will be described in the next subsection). The length of the value list in a locus was 6. The genetic operation for the SGA was standard bit

mutation. For both GAs, the per-bit mutation rate, q, was set at $1/L = 0.00049$. Tournament selection was adopted. *Elitism* was applied. The tournament size was set at 2 for the SGA and 6 for the OGA. A generational model was used. The run lasted 1,000 generations.

The features of the TNK(p) landscapes for $N = 20, K \in \{0, 2, 6, 12, 19\}, F = 2, (P = 0.99)$, which were obtained in the complementary computer simulations with the same parameters of the SGA as those of the EANNs, were used as baselines for estimating indirectly the features of the EANNs in the same way as in Section 4.2.1.

Experimental Results

Fig. 4.33 shows the genetic distance at each generation. The approximately linear increase was observed in the run. This indicates the presence of neutrality in the fitness landscape of the EANN.

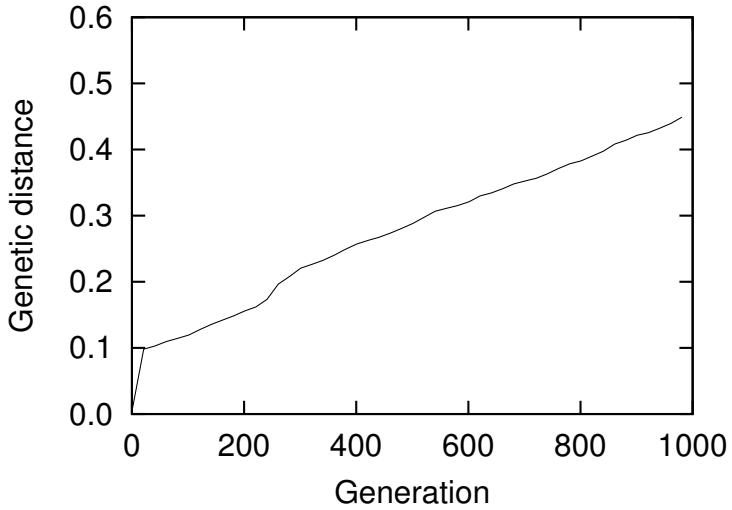


Figure 4.33: Genetic distance at each generation for the SGA at $q = 1/L$ in 10 runs

α was calculated by using the method of least squares on the result. Table 4.1 shows the correlation ($\dot{E}_b$) and the α for the SGA. The landscape was more rugged than those of the discrimination problem and the pursuit problem.

Table 4.1: Correlation and α for the SGA

Correlation ($\dot{E}_b$)	α
0.353505	0.000371056

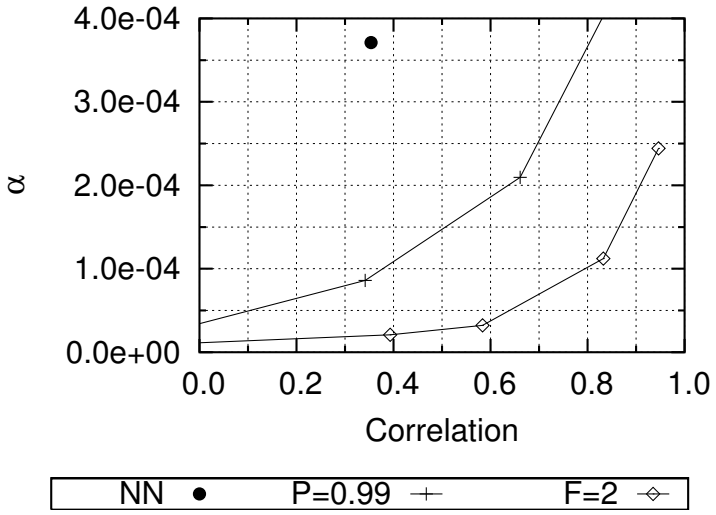


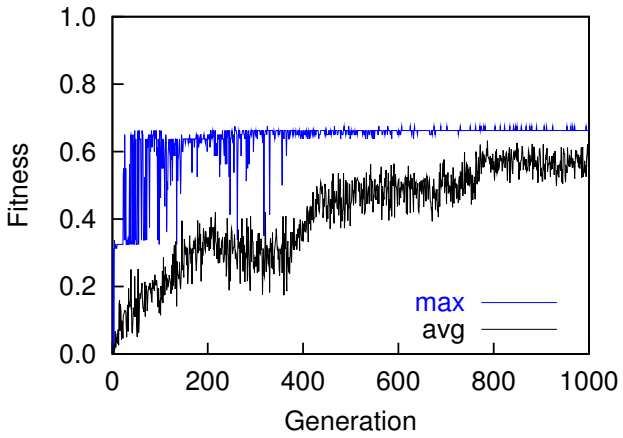
Figure 4.34: α as a function of the correlation ($\dot{E}_b$)

The neutrality of the EANN were estimated indirectly by using the correlation-the number of substitutions curves in the TNK(p) landscape (F, P are constant) as baselines (Fig. 4.34). By using the curve for $P = 0.99$ as a baseline, it was observed that neutrality of this landscape is so large, compared with those of the discrimination problem and the pursuit problem.

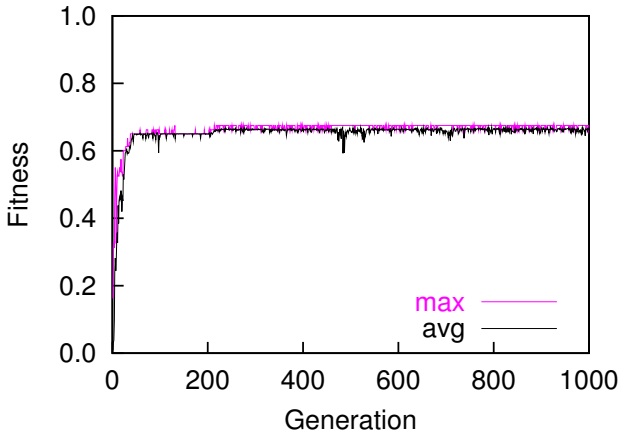
Fig. 4.35 shows the maximum and average fitness at each generation for the SGA and the OGA. The SGA for $s = 2$ was not trapped on local optima although the landscape was so rugged. In the final generation, there was no significant difference between the SGA and the OGA. However, both the maximum and average fitness increased faster for the OGA than for the SGA.

These would be due to large neutrality in the fitness landscape. In (Barnett, 1997; Newman and Engelhardt, 1998; Smith et al., 2002a) mentioned in Chapter 1, it has been reported that ruggedness is independent of neutrality in tunably neutral NK

landscapes. However, the performance of the SGA in this real-world problem cannot be explained enough with related to this assumption. Therefore, the GA employing the variable mutation rate strategy, such as the OGA, which shows good performances in both smooth and rugged landscapes is preferable to real-world problems where fitness landscapes include neutrality.



(a) SGA for $s = 2$

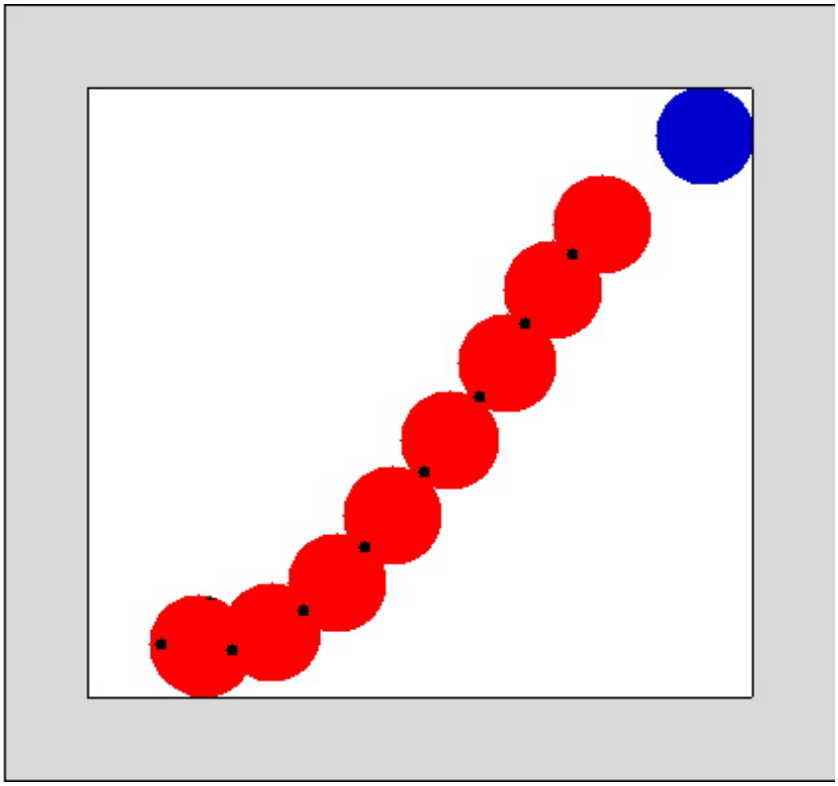


(b) OGA for $s = 6$

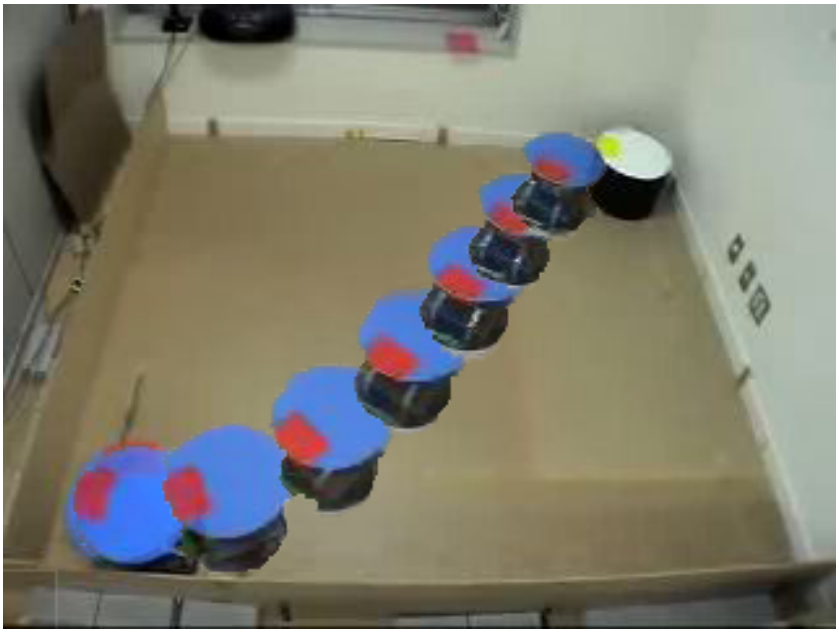
Figure 4.35: Maximum and average fitness at each generation for the SGA and the OGA

Agent's behavior

The controllers evolved in simulation were transferred to the real robot. Fig. 4.36 to 4.39 show the typical behaviors from each initial orientation of the best evolved agent by the SGA and the OGA, respectively. The controllers evolved by the SGA displayed a single strategy for each initial orientation (Fig. 4.36 and 4.37); The robot stayed at initial position turning left to direct toward the object, then he approached to it. On the other hand, the controllers evolved by the OGA displayed more interesting behaviors, that is, two kinds of strategy for each initial orientation (Fig. 4.38 and 4.39); At initial upward orientation, the robot displayed the same strategy as the robot with the controller evolved by the SGA. At initial rightward one, the robot stayed at initial position turning right to direct toward the object, then he approached to it. This strategy is more efficient to reach the object than the strategy in which the robot always turns left for both initial orientations. These difference would be due to the difference of the fitness value obtained by the individual of the SGA and the OGA in the last generation, because the fitness function employed in this experiment induces the individuals to reach the object quickly.

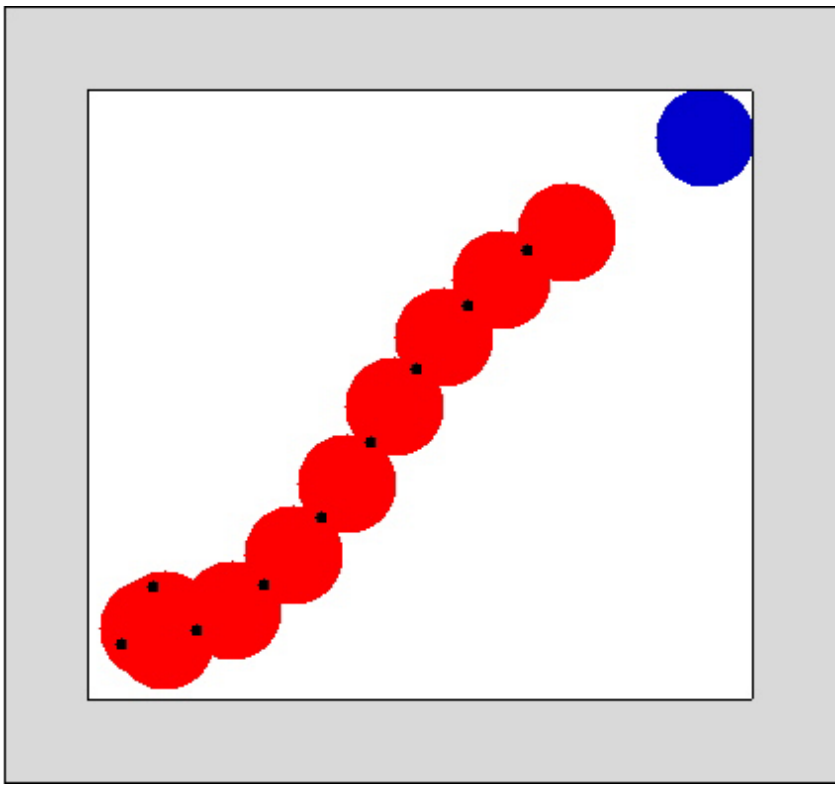


(a) Behavior in the simulated environment

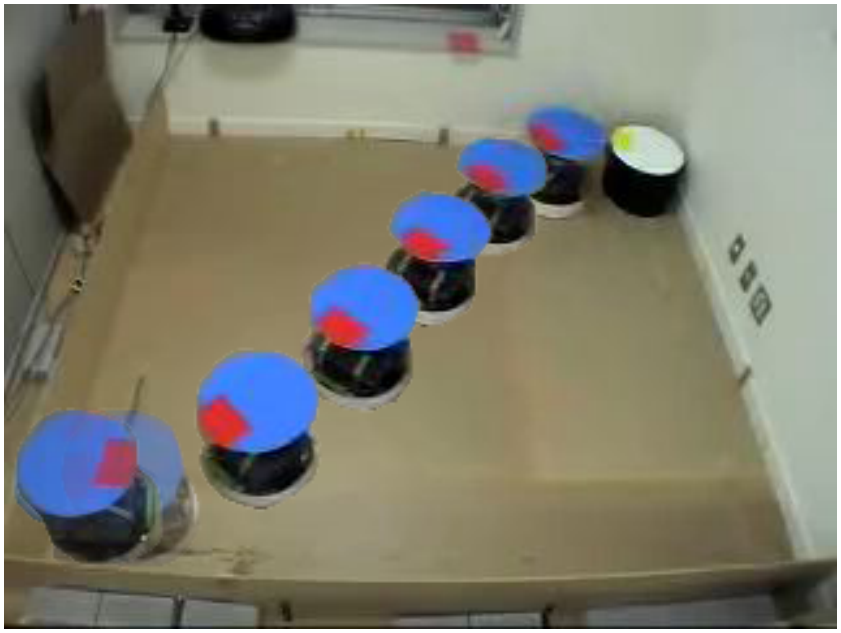


(b) Behavior in the real environment

Figure 4.36: Behavior from the initial upward orientation of the best evolved agent by the SGA

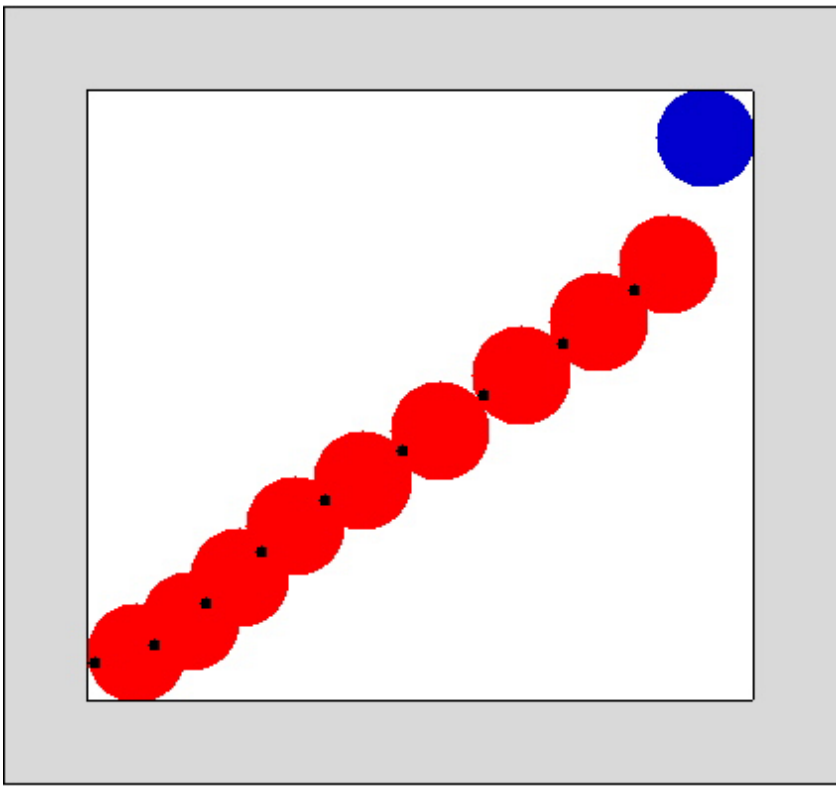


(a) Behavior in the simulated environment

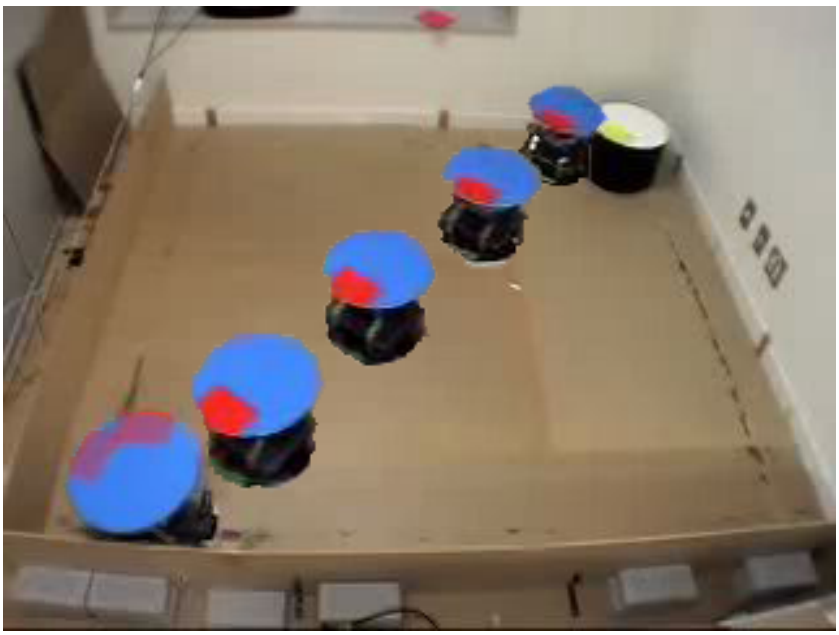


(b) Behavior in the real environment

Figure 4.37: Behavior from the initial rightward orientation of the best evolved agent by the SGA

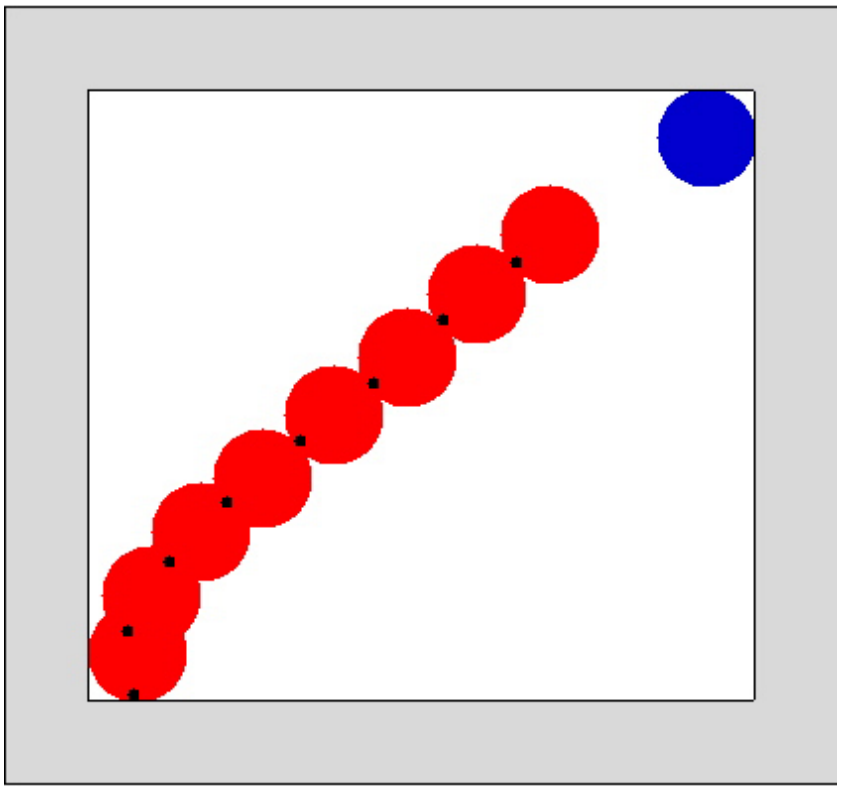


(a) Behavior in the simulated environment

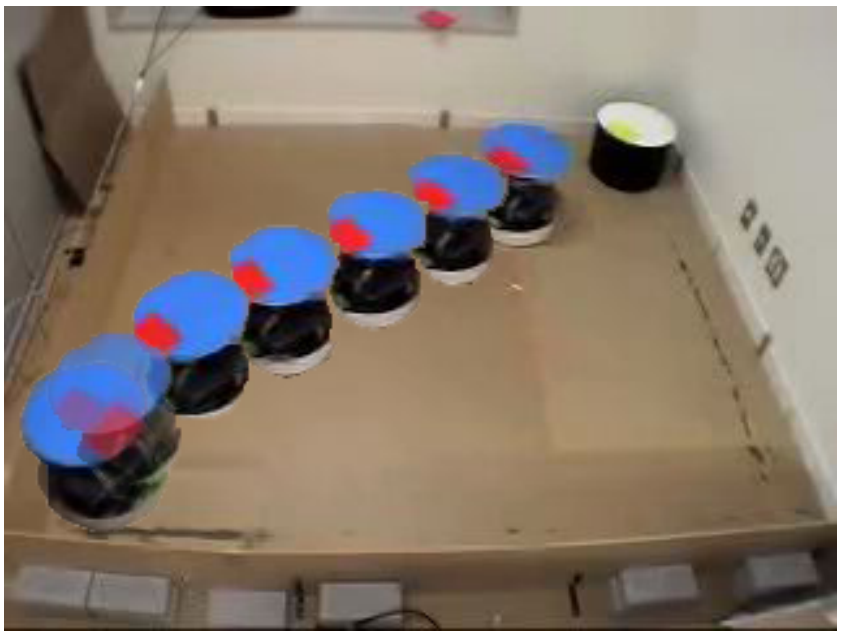


(b) Behavior in the real environment

Figure 4.38: Behavior from the initial upward orientation of the best evolved agent by the OGA



(a) Behavior in the simulated environment



(b) Behavior in the real environment

Figure 4.39: Behavior from the initial rightward orientation of the best evolved agent by the OGA

4.5 Summary

In this chapter, the Nei's standard genetic distance were applied for describing the fitness landscapes of evolution of artificial neural networks for robot control in the simulated and the real problem to validate the use of it for estimating the degree of neutrality. The results can be summarized as follows:

- In all the problems, the Nei's standard genetic distance increased approximately linearly over generations in all runs. This demonstrates the presence of neutrality in the fitness landscape of those EANNs.
- There are some cases that it is not possible to estimate the increase of neutrality between the landscapes even if combined with the measure of ruggedness.
- The complementary use of the features of a test function as baselines makes possible for the proposed approach to estimate the degree of neutrality in the fitness landscape in the real-world problems.
- The number of hidden neurons in evolutionary artificial neural networks for robot control has a great influence on the features of the fitness landscape.
- The landscapes of the pursuit problem and the goal reach problem are more rugged and more neutral than the one of the discrimination problem.

Based on those features of the fitness landscapes, the standard GA and the operon-GA were applied to the problems to investigate their performances. The results can be summarized as follows:

- The discrimination problem does show phases of neutral evolution. This is consistent with the description by the Nei's standard genetic distance. However, the pursuit problem does not show such clear phases of neutral evolution in the average run although the result estimated by the Nei's standard genetic distance tells the presence of neutrality in the fitness landscape.
- The standard GA with low selection pressure and the operon-GA were able to continually find better regions of the search space.

- The standard GA can easily get trapped on local optima under conditions of high selection pressure and a low mutation rate.
- The benefits of the variable mutation rate strategy used by the operon-GA increase as the ruggedness of the landscapes increases or with a larger genetic search space.
- The best evolved controllers of the neural networks by the operon-GA was better suited in the simulated and the real environment than those by the standard GA.

These results are consistent with results using simple neutral networks and terraced NK landscapes in Chapter 2. Therefore, this demonstrates that the proposed approaches are applicable to real-world problems.

Chapter 5

Conclusions

The objective of this thesis is to establish the theoretical guidelines of artificial evolution on fitness landscapes with neutral networks, describing fitness landscapes where neutral evolution occurs, and analyzing the behavior of evolving populations on the fitness landscapes under various conditions.

Chapter 1 introduced the theoretical background of neutral networks and pointed out the intrinsic problems arising from them. Then, the approaches in this thesis to these problems were shown. Finally, the outline of this thesis was described.

Chapter 2 investigated the evolutionary dynamics of GAs in simple test function and complex test functions, where their fitness landscapes includes neutral networks. Two types of GA were used. One was the standard GA, which employs a constant mutation rate strategy and the other was the operon-GA, which employs a variable mutation rate strategy. A simple test function, called the Balance Beam Function was defined for examining the effect of selection pressure and the variable mutation rate strategy on the evolutionary dynamics in it. The results demonstrate that speed has an optimal mutation rate and an error threshold since plotting speed against mutation rate resulted in a concave curve. Increasing selection pressure increased the speed of a population's movement on a neutral network. The variable mutation rate strategy of the operon-GA improved the efficiency of the search. In order to investigate whether the results in the simple test function generalize to more complex functions, the terraced NK landscape incorporating both neutrality and ruggedness

was introduced. In this function, the standard GA prefers low selection pressure while the operon-GA prefers high selection pressure. It is found that the variable mutation rate strategy is also beneficial with this more complex test function, and that these benefits increase as the fitness landscape becomes more rugged.

Chapter 3 introduced the second main direction of this thesis, describing fitness landscapes with neutral networks. The Nei's standard genetic distance, which originates from population genetics, was proposed after minor modifications and then the characteristics of it were investigated in tunably neutral landscapes. The results suggest that the genetic distance can provide a guideline for estimating the degree of neutrality in fitness landscapes combined with the measure of ruggedness. It was also confirmed that the results were consistent with the neutral theory and the nearly neutral theory in population genetics.

Chapter 4 explored how well the results obtained in the previous chapters apply to real-world problems. Section 4.2 investigated a simulated real-world problem, a discrimination problem. The Nei's genetic distance was used to estimate and compare the degree of neutrality in the fitness landscapes among the neural network controllers varying the number of hidden neurons. By using the features of the fitness landscape of a test function as baselines, the proposed method can estimate and compare indirectly the features of the fitness landscape of the EANNs. It was confirmed that the number of hidden neurons have a great influence on the features of the fitness landscape of neural networks. According to the obtained features of the fitness landscape, the standard GA and the operon-GA were applied to this problem. This problem does show phases of neutral evolution for each run, which is consistent with the description of the fitness landscape. The standard GA with low selection pressure and the operon-GA were able to continually find better regions of the search space. But the standard GA can easily get trapped on local optima under conditions of high selection pressure and a low mutation rate. The benefits of the variable mutation rate strategy used by the operon-GA were more pronounced with a larger genotypic search space. These results are also consistent with the results obtained using simple neutral networks and terraced NK landscapes. Section 4.3 investigated another simulated real-world problem, a pursuit problem. Neutrality of the problem was also

estimated by the genetic distance with the features of a test function as baselines. It was found that neutrality and ruggedness of this landscape were larger than those of the landscapes in the discrimination problem. Both GAs tuned based on the features of the fitness landscape were applied to this rugged problem. The standard GA with high selection pressure was trapped on local optima as predicted. In contrast, the operon-GA with high selection pressure continued to find better regions of the search space and outperformed the standard GA with low selection pressure. The best evolved controllers of the neural networks by the operon-GA showed more efficient behavior for pursuiting the evader than the ones by the standard GA. These results show that the operon-GA's variable mutation rate strategy is also beneficial to this problem. Section 4.4 analyzed a single evolutionary run for each GA in a real-world problem, a goal reach problem. The controllers were initially evolved in simulation, before being transferred to the real robot. The Nei's genetic distance was applied to estimate the degree of neutrality in the fitness landscape. Based on the features of the fitness landscape, both GAs were applied. In the final generation, there was no significant difference between the standard GA and the operon-GA. However, both the maximum and average fitness increased faster for the operon-GA than for the standard GA. The best evolved controllers of the neural networks in the simulation were evaluated in the real environment, then successfully accomplished their task.

From these results, it can be concluded that this thesis has established the theoretical guidelines of artificial evolution on fitness landscapes with neutral networks and they can be successfully applied to real-world problems.

Appendix A

The Standard GA

The framework of genetic algorithms (GAs) was proposed as one of artificial adaptive systems. However, it is frequently employed as an optimization method. A general problem to be solved by GAs can be formulated as follows:

$$\max \quad F(x) \tag{A.1}$$

$$\textit{subject to} \quad x \in \mathcal{G} \tag{A.2}$$

$$\mathcal{G} \in \mathcal{H} \tag{A.3}$$

where, F is the objective function, $\mathcal{H}$ is the defined space and $\mathcal{G}$ is the valid defined space in $\mathcal{H}$. Equation A.3 shows the constraints of the problem. For this problem, this framework first map x which is called *phenotype* into a sequence of symbols which is called *genotype*. In the standard GAs, the genotype is represented as a binary string. The positions of the string are called *loci* of the genotype. The variable at a locus is called *gene*, its value *allele*.

Similarly to biological species that evolve in environments by passing and modifying their genetic codes through generations, this framework considers a population of individuals, i.e., a group of search points. At the alternation of generations, they can exchange their parts of genetic codes stochastically based on various predefined rules, which are generally called *genetic operations*.

The canonical steps of GAs can be summarized as follows (Fig. A.1):

STEP1 Create an initial population randomly.

STEP2 Compute phenotypes of all the individuals in the population.

STEP3 Evaluate all the individuals by the objective function.

STEP4 Compute fitness values based on the distribution of the objective values in the population.

STEP5 Select individuals for the next generation.

STEP6 Apply genetic operations such as crossover and mutation, forming a new population.

STEP7 Return to STEP2, until the termination condition is settled.

The objective function is a scalar function and the fitness function determines the reproduction ratio by the method, such as scaling or ranking. The selection process is often performed by the roulette wheel selection.

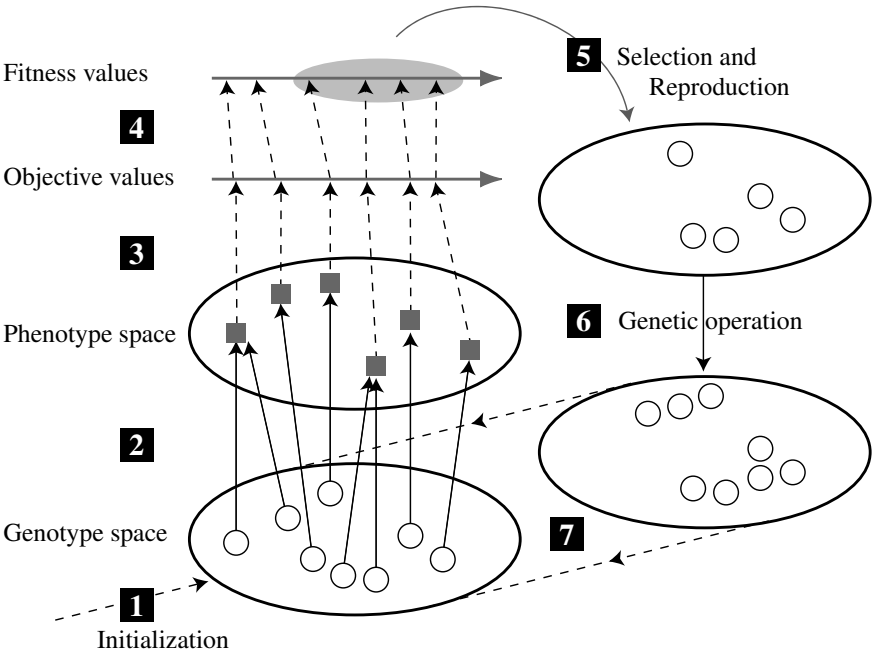


Figure A.1: Canonical steps of genetic algorithms

Appendix B

The Operon-GA

The operon-GA(Ohkura and Ueda, 1999) has the same computational steps as those of the standard GA. The only differences from the standard GA are (1) a genotype has redundant genetic information with respect to its phenotype, and (2) it uses five new genetic operators for the redundant genotype.

Genotype and Phenotype: The genotype (*string*) can be written as follows:

$$string = \{operon_i \mid i = 1, \ldots, l\} \tag{B.1}$$

$$operon_i = \{locus_j \mid j \in O_i\} \tag{B.2}$$

$$locus_j = \{label_j, x_1, x_2, \ldots, x_n\} \tag{B.3}$$

where l is the number of operons and O_i is the set of indices of loci included in the i-th operon. Assuming that S is the set of the indices of all the loci determined by a problem, $\cup_i O_i = S$, $O_i \cap O_j = \phi$, $i \neq j$. A locus has a label ($label_j$) and a value list ($x_1, x_2, \ldots, x_n$), where the first element (x_1) is active and the others ($x_2, \ldots, x_n$) are inactive. The phenotype is defined as the vector of active values in the order of labels.

Genetic Operations: The two genetic operations, g_{con} and g_{div}, are defined as follows:

$$g_{con} \ : \ operon_i + operon_j \longrightarrow operon_i' \tag{B.4}$$

$$g_{div} \quad : \quad \begin{array}{l} (O_i \cup O_j = O'_i) \\ operon_i \longrightarrow operon'_i + operon'_j \\ (O_i = O'_i \cup O'_j, O'_i \cap O'_j = \phi) \end{array} \tag{B.5}$$

The other three genetic operations introduce some genetic change through the following procedure: (1) select an operon stochastically, and then (2) introduce the same type of genetic change to all the loci in the operon. Assuming that $locus_j = \{label_j, x_1, \ldots, x_n\}$ is changed to $locus'_j = \{label_j, x'_1, \ldots, x'_n\}$, the three genetic operations called g_{dup}, g_{del} and g_{inv} are defined as follows :

$$g_{dup} \quad : \quad \{x_1 \rightarrow x'_1 = x'_2, x_2 \rightarrow x'_3, \ldots, x_{n-1} \rightarrow x'_n\} \tag{B.6}$$

$$g_{del} \quad : \quad \{x_2 \rightarrow x'_1, x_3 \rightarrow x'_2, \ldots, random\{0,1\} \rightarrow x'_n\} \tag{B.7}$$

$$g_{inv} \quad : \quad \{\ x_k \rightarrow x'_1, x_1 \rightarrow x'_k\}, \exists k \neq 1 \tag{B.8}$$

Conventional genetic operations, mutations and crossovers, can also be defined for the genotype.

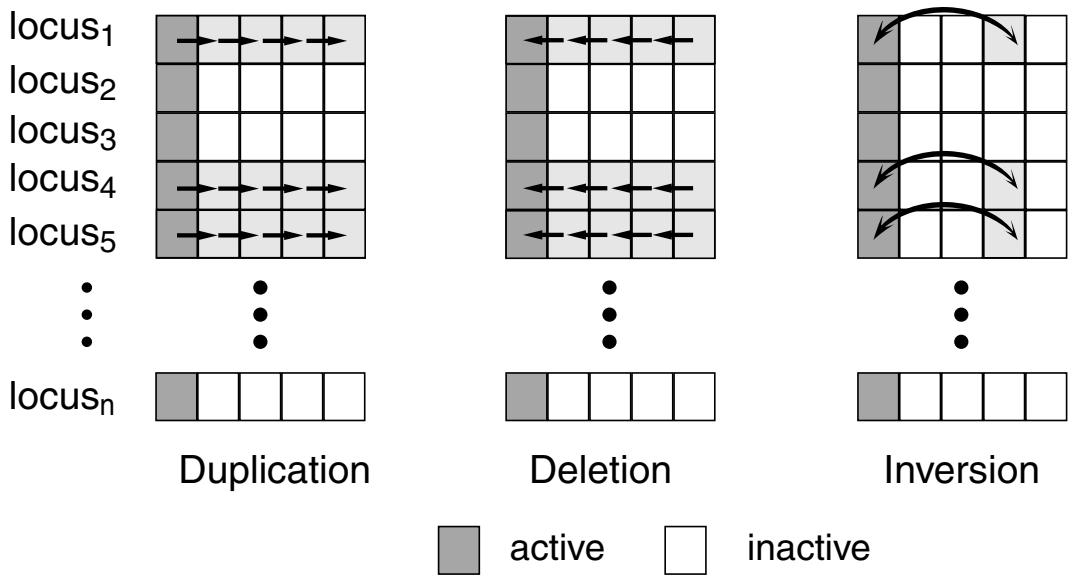


Figure B.1: Genetic operations in an operon layer

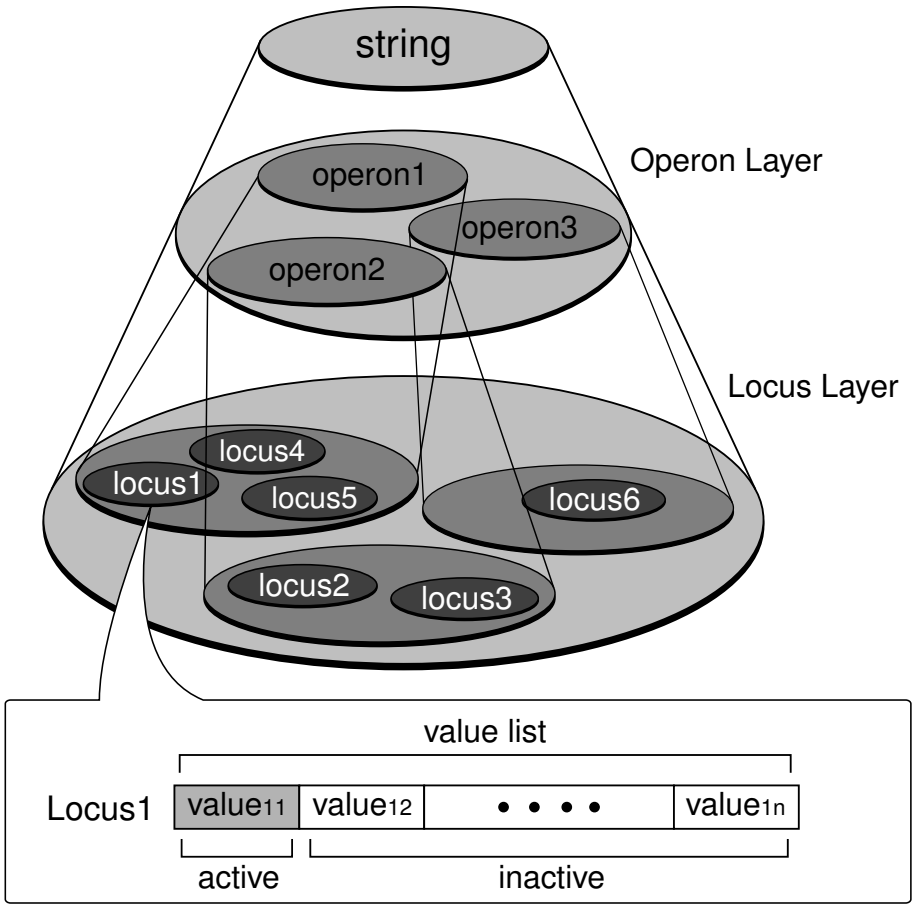
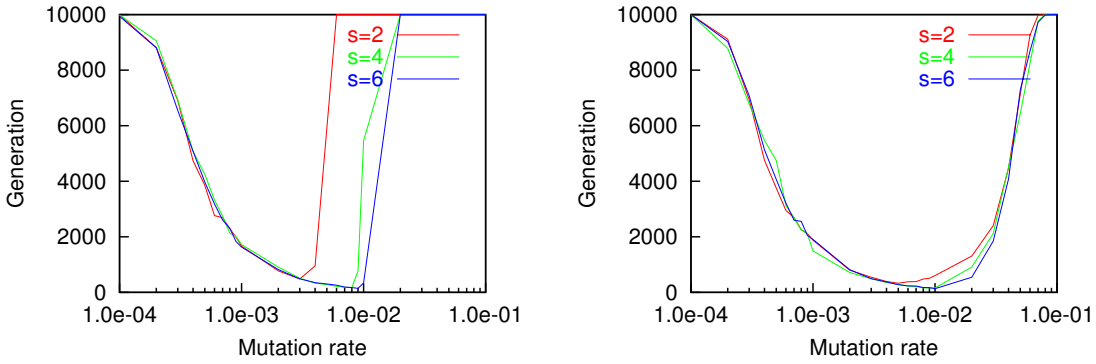


Figure B.2: Representation of genotype in the operon-GA

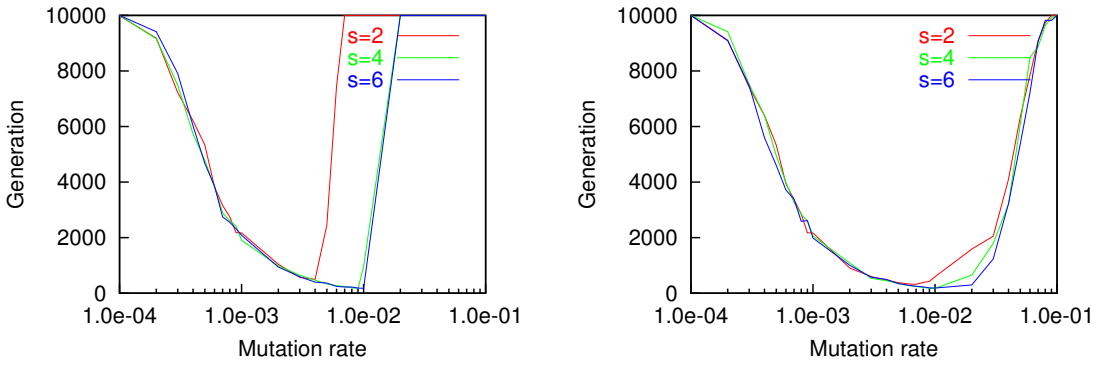
Appendix C

Evolutionary Dynamics in the Balance Beam Function



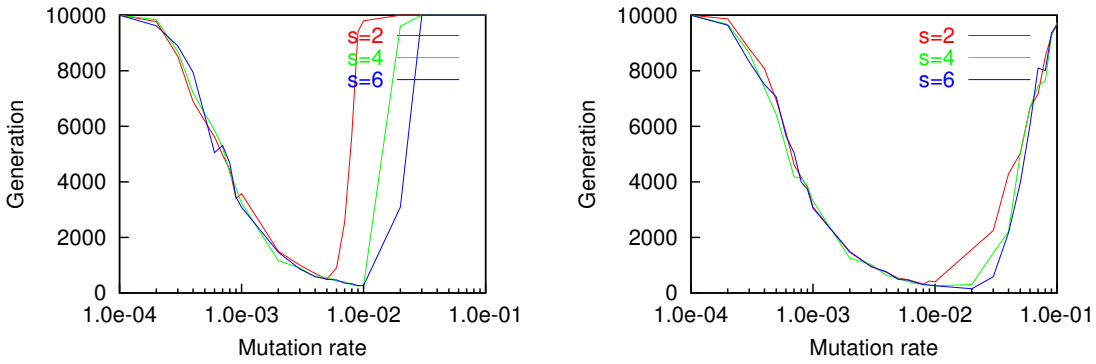
(a) without elitism, $w = 5$

(b) elitism, $w = 5$



(c) without elitism, $w = 10$

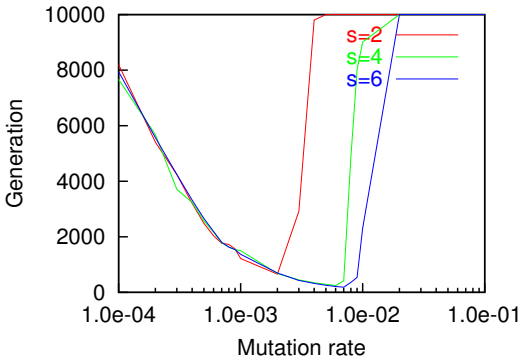
(d) elitism, $w = 10$



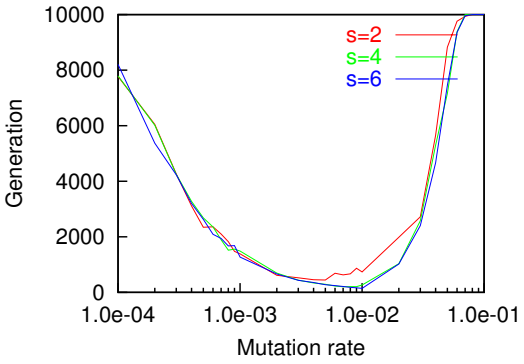
(e) without elitism, $w = 20$

(f) elitism, $w = 20$

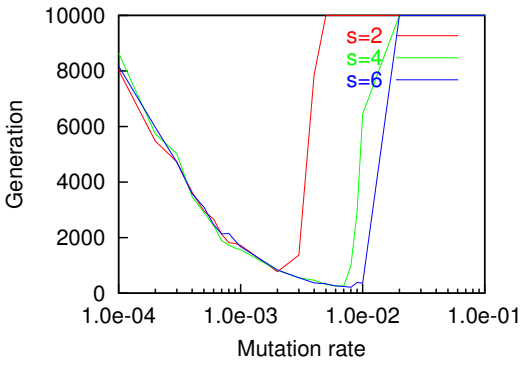
Figure C.1: Average generations to reach the furthest distance in 50 runs by the SGA for each w



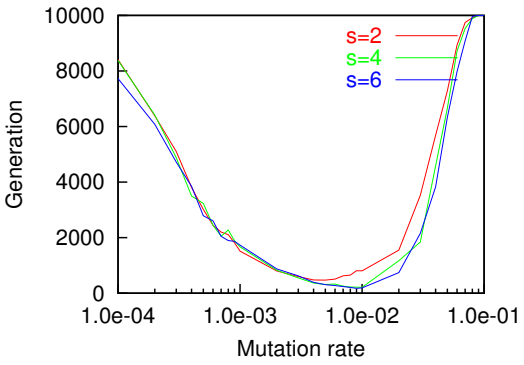
(a) without elitism, $w = 5$



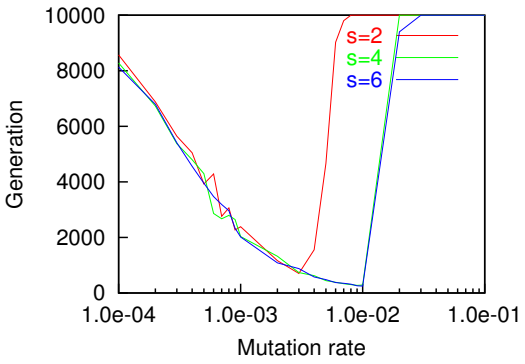
(b) elitism, $w = 5$



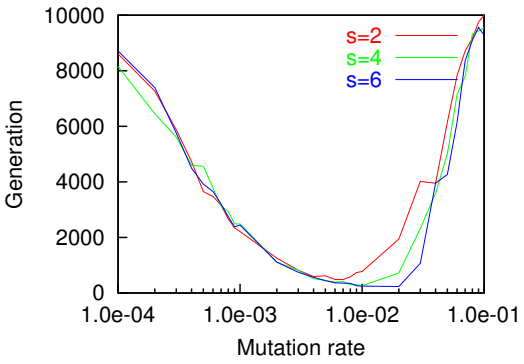
(c) without elitism, $w = 10$



(d) elitism, $w = 10$

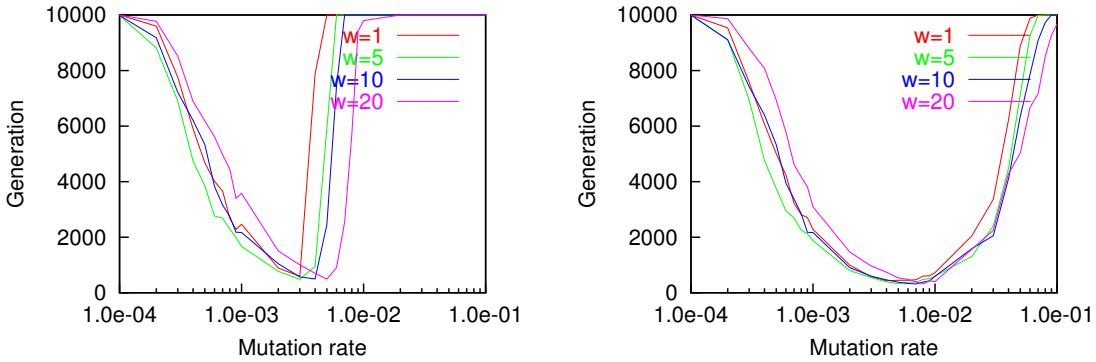


(e) without elitism, $w = 20$



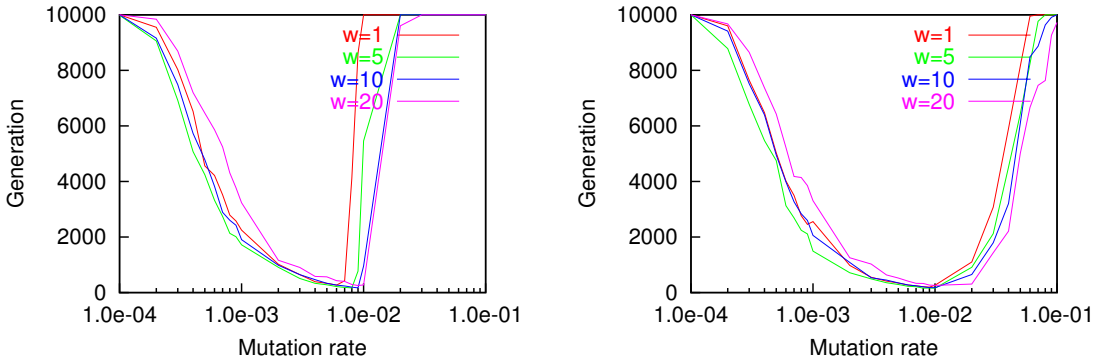
(f) elitism, $w = 20$

Figure C.2: Average generations to reach the furthest distance in 50 runs by the OGA for each w



(a) without elitism, $s = 2$

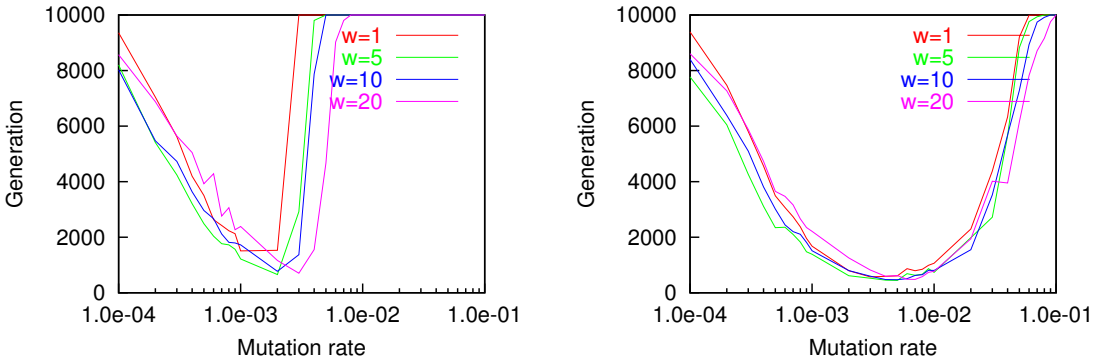
(b) elitism, $s = 2$



(c) without elitism, $s = 4$

(d) elitism, $s = 4$

Figure C.3: Average generations to reach the furthest distance in 50 runs by the SGA for each s



(a) without elitism, $s = 2$

(b) elitism, $s = 2$

(c) without elitism, $s = 4$

(d) elitism, $s = 4$

Figure C.4: Average generations to reach the furthest distance in 50 runs by the OGA for each s

Appendix D

Neural Controller –Spike Response Model–

The agent's behavior is controlled by the *spike response model* network (Maass and Bishop, 1998), which is a form of *Pulsed Neural Network* (PNN). The state of a spiking neuron is described by the voltage difference across its membrane, which is called, "membrane potential" u. Incoming spikes can increase or decrease the membrane potential. A neuron emits a spike when the total amount of excitation due to incoming excitatory and inhibitory spikes exceeds its firing threshold, θ. After firing, the membrane potential of the neuron is set to a low negative voltage, it then gradually returns to its resting potential; during this refractory period, a neuron cannot emit a new spike (Fig. D.1). This recharging period is called the *refractory period*. The membrane potential of a neuron i at time t is given by:

$$u_i(t) = \sum_{t_i^{(f)} \in F_i} \eta_i(t - t_i^{(f)}) + \sum_{j \in \Gamma_i} \sum_{t_j^{(f)} \in F_j} \omega_{ij} \varepsilon_{ij}(t - t_j^{(f)}) \tag{D.1}$$

where $t_i^{(f)}$ is the firing time of neuron i, F_i is the set of firing times in a neuron i. The neuron i may receive the input from presynaptic neurons $j \in \Gamma_i$. The weight ω_{ij} is the strength of the connection from the j-th neuron.

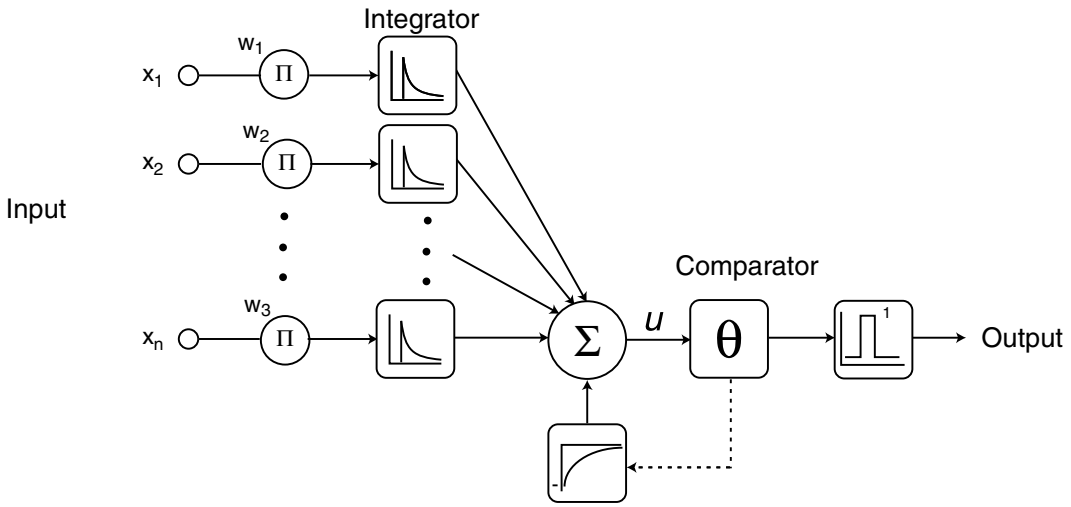


Figure D.1: Neuron model.

The function η_i, accounting for neuronal refractoriness, is given by:

$$\eta_i(r) = -\exp(-\frac{r}{\tau_m})H(r) \tag{D.2}$$

Here $r = t - t_i^{(f)}$ is the difference between the time t and the time of firing $t^{(f)}$ of neuron i, τ_m is a membrane time constant and $H(r)$ is the Heaviside step function which vanishes for $r < 0$ and gives a value of 1 for $r > 0$.

The function ε_{ij} describes the response to postsynaptic spikes:

$$\varepsilon_{ij}(r) = [\exp(-\frac{r - \Delta^{ax}}{\tau_m})(1 - \exp(-\frac{r - \Delta^{ax}}{\tau_s}))]H(r - \Delta^{ax}), \tag{D.3}$$

where τ_s is a synaptic time constant, Δ^{ax} is the axonal transmission delay. The amplitude of the response is scaled via the strength of the connection ω_{ij} given in Equation D.1. In this thesis, the parameters of the neurons and synapses are set as follows: $\tau_m = 4$, $\tau_s = 10$, $\Delta^{ax} = 2$ for all neurons and all synapses in the network following the recommendations given in (Floreano and Mattiussi, 2001).

Appendix E

Evolutionary Dynamics in a Discrimination Problem

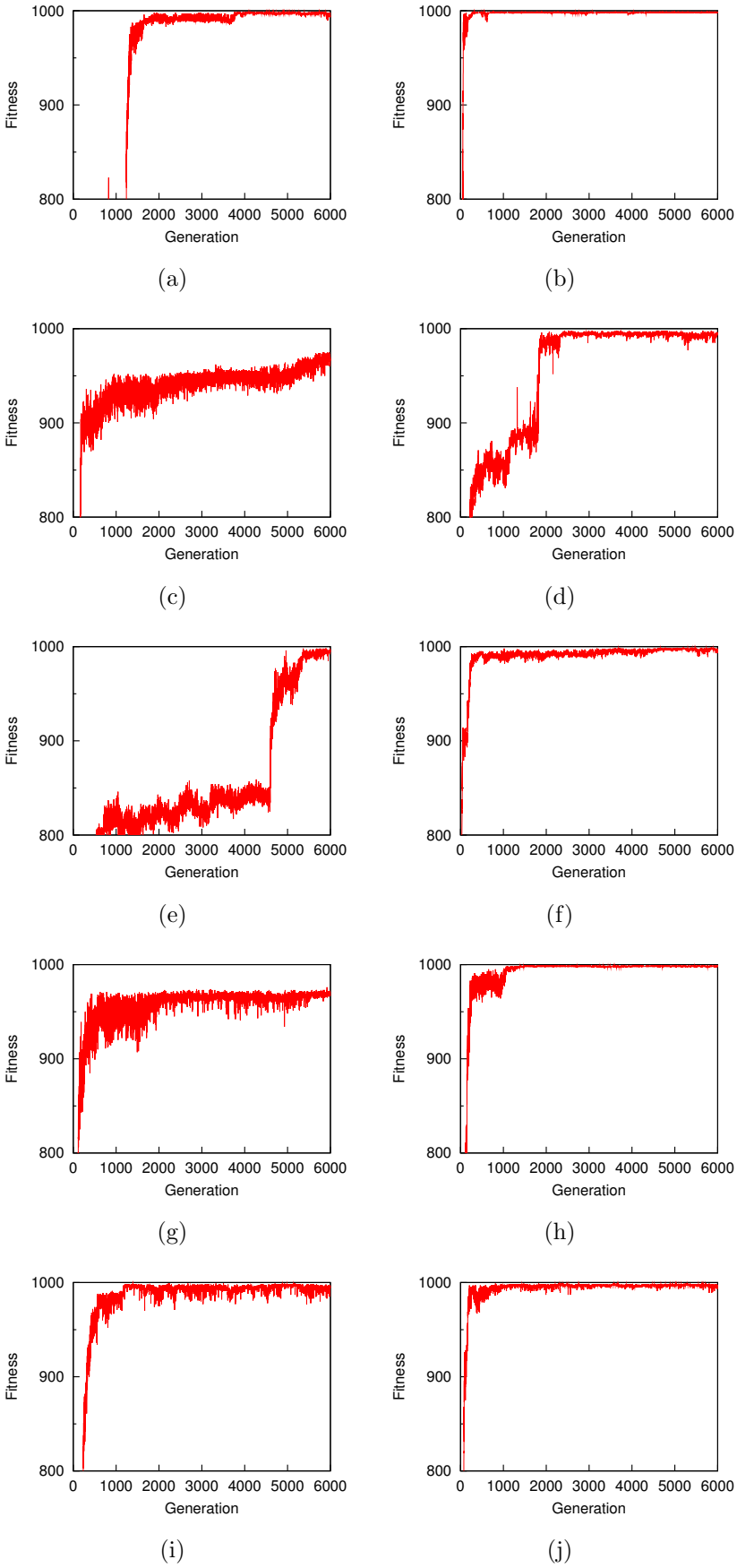


Figure E.1: Maximum fitness at each generation in each run by the SGA without elitism for $s = 2$ and $Nh = 1$

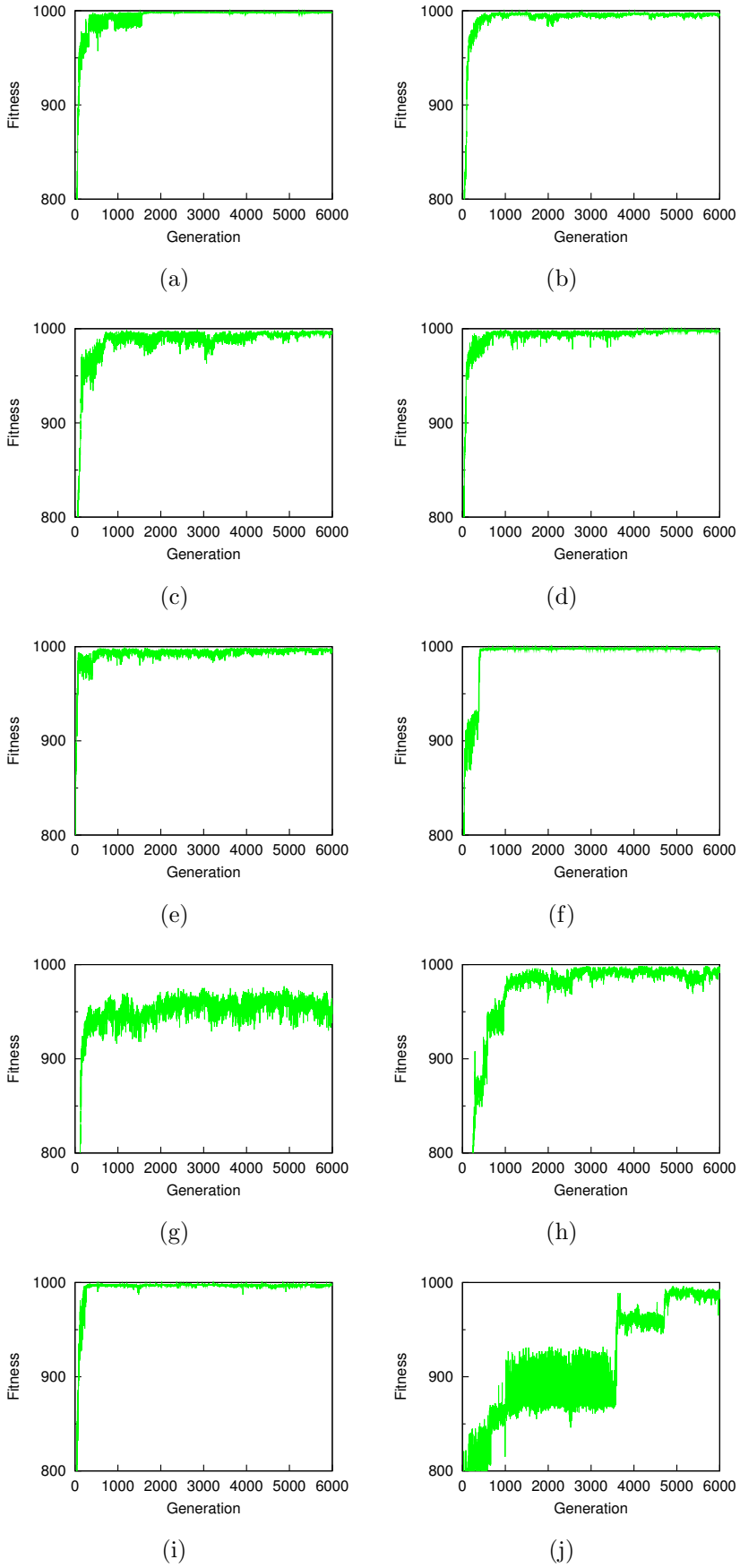


Figure E.2: Maximum fitness at each generation in each run by the OGA without elitism for $s = 2$ and $Nh = 1$

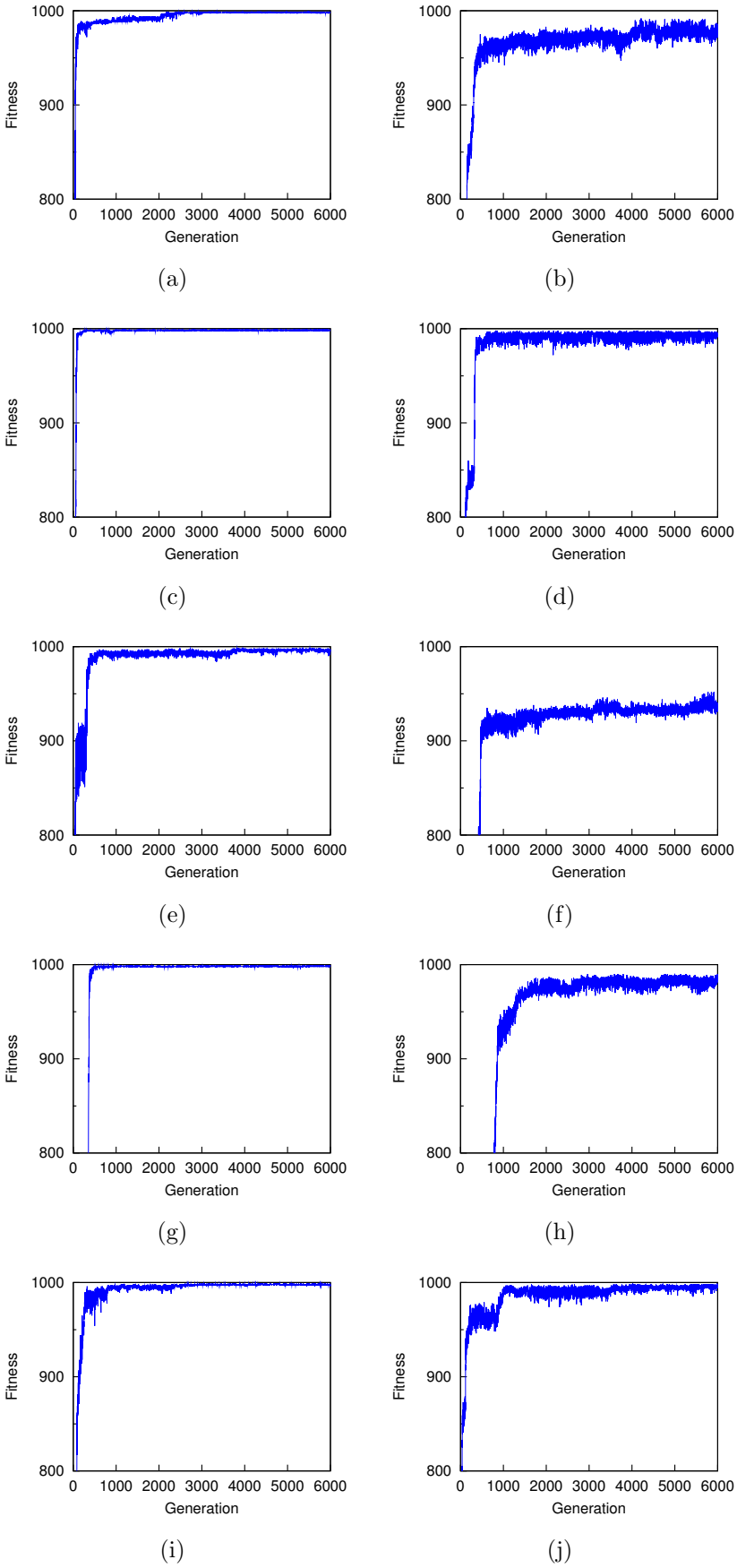


Figure E.3: Maximum fitness at each generation in each run by the SGA with elitism for $s = 2$ and $Nh = 1$

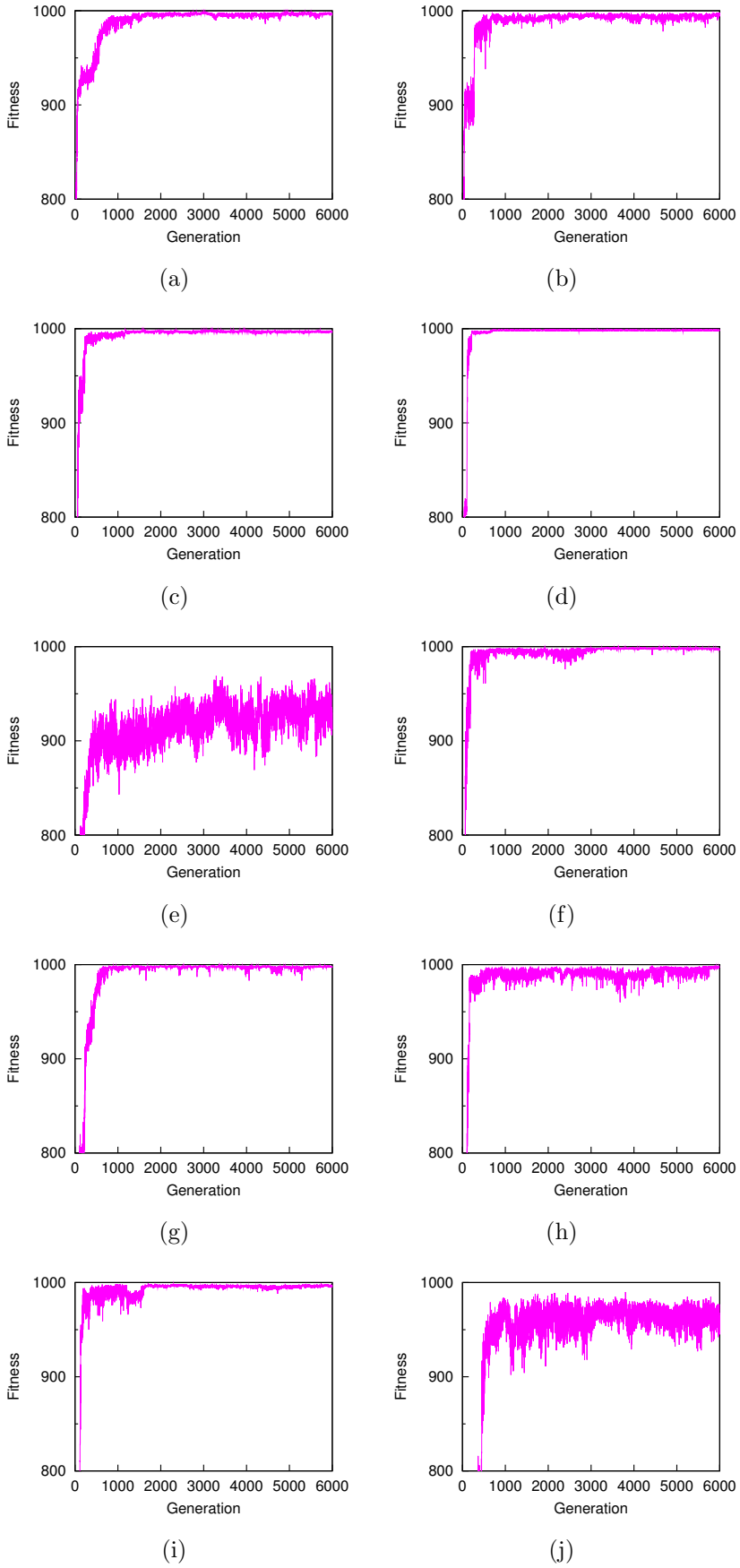


Figure E.4: Maximum fitness at each generation in each run by the OGA with elitism for $s = 2$ and $Nh = 1$

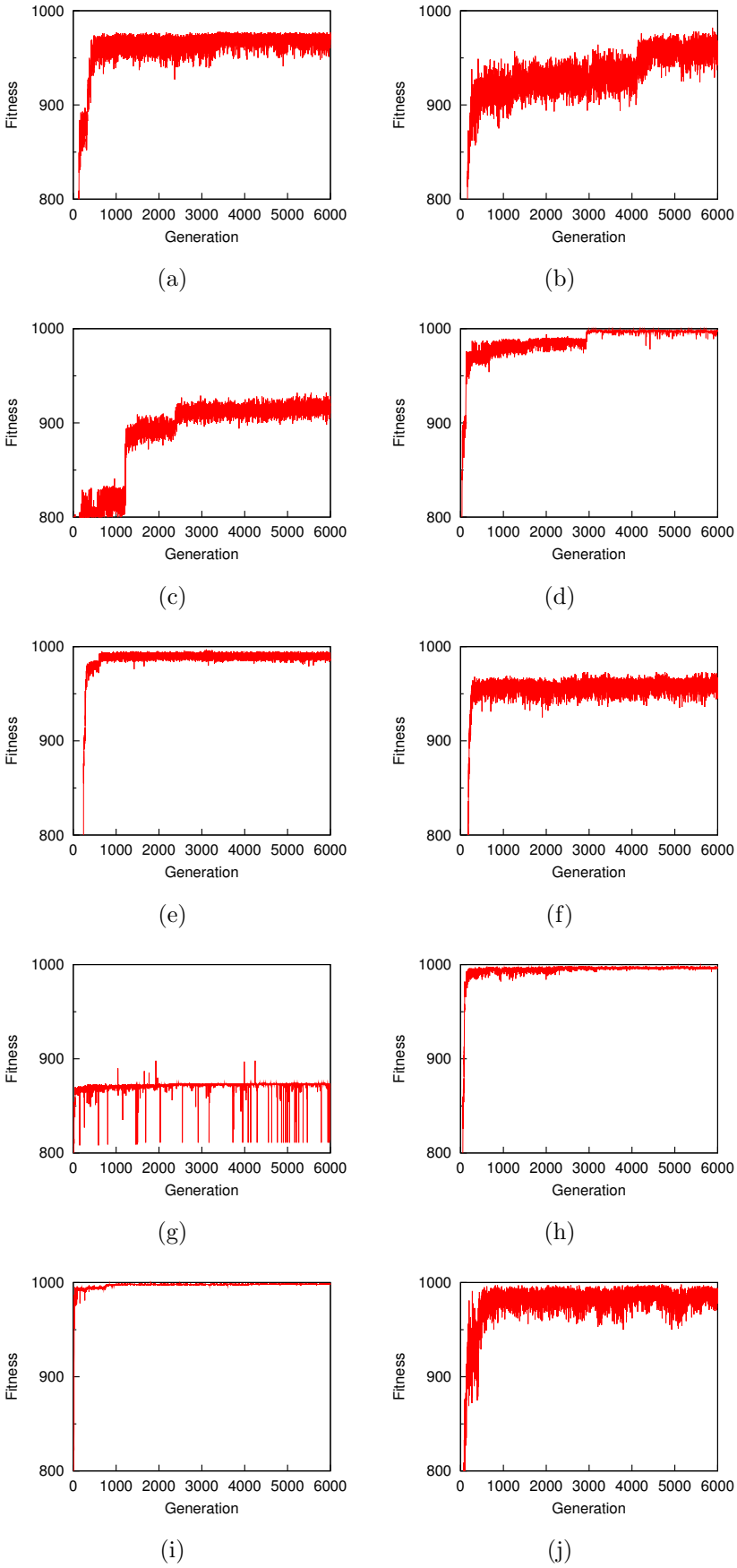


Figure E.5: Maximum fitness at each generation in each run by the SGA without elitism for $s = 6$ and $Nh = 1$

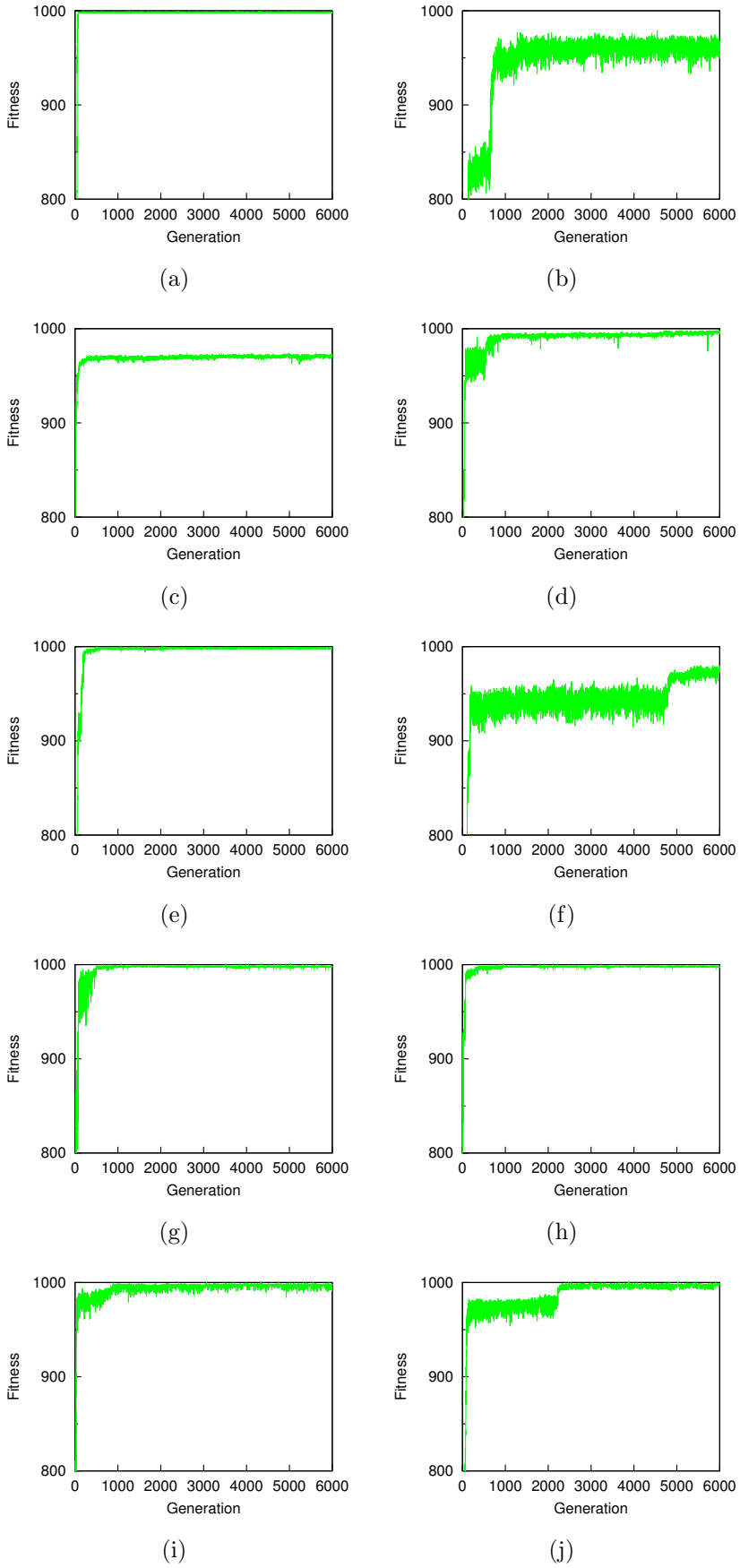


Figure E.6: Maximum fitness at each generation in each run by the OGA without elitism for $s = 6$ and $Nh = 1$

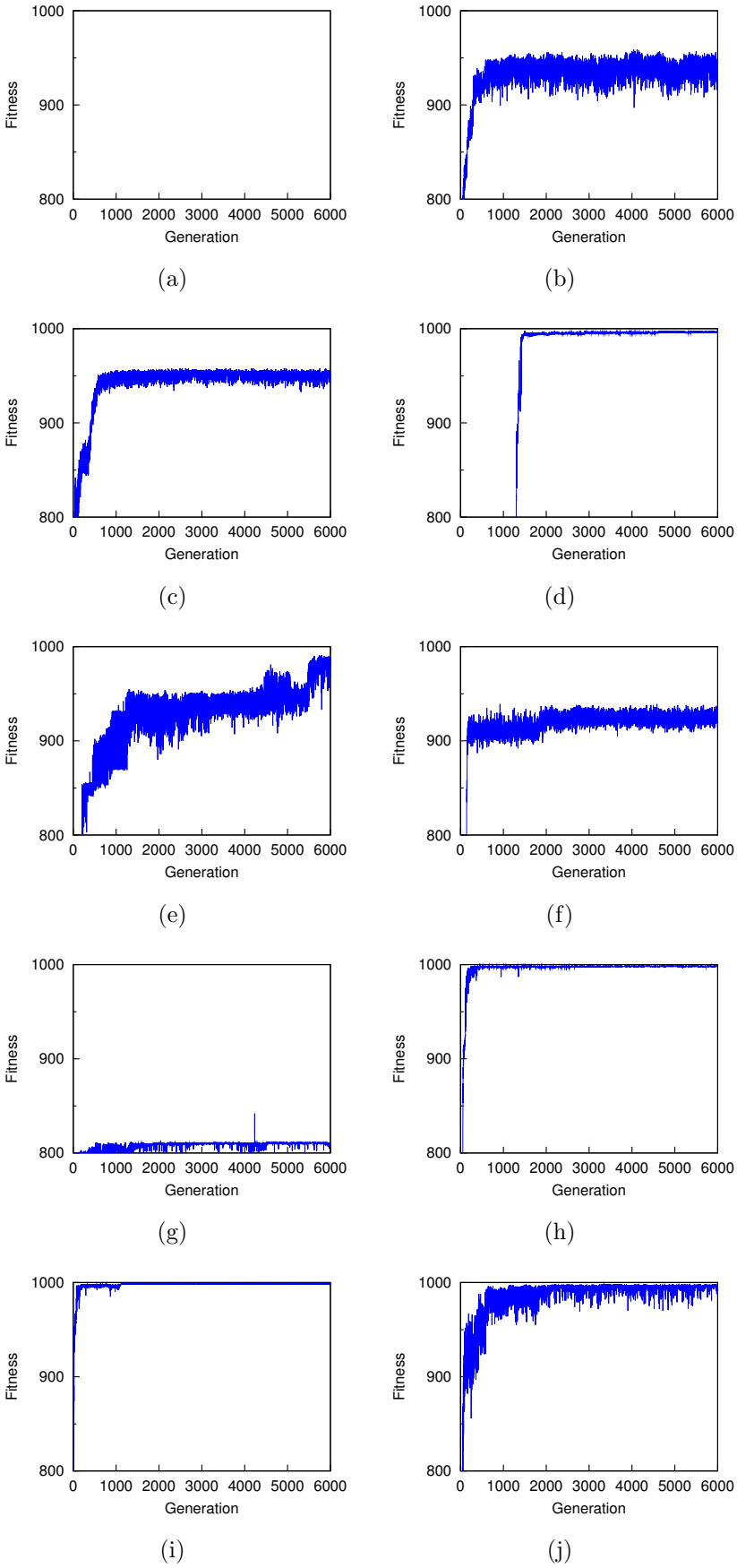


Figure E.7: Maximum fitness at each generation in each run by the SGA with elitism for $s = 6$ and $Nh = 1$

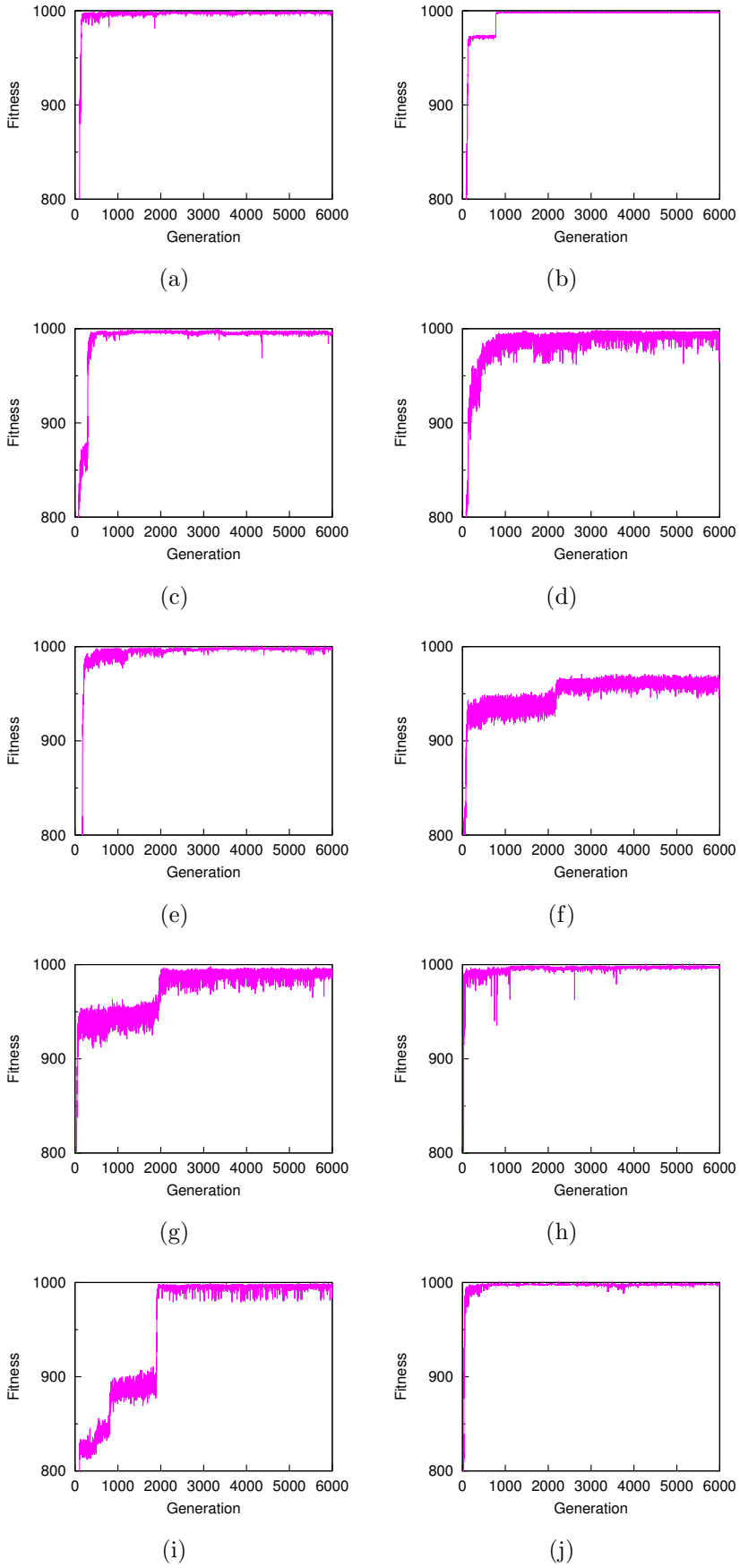


Figure E.8: Maximum fitness at each generation in each run by the OGA with elitism for $s = 6$ and $Nh = 1$

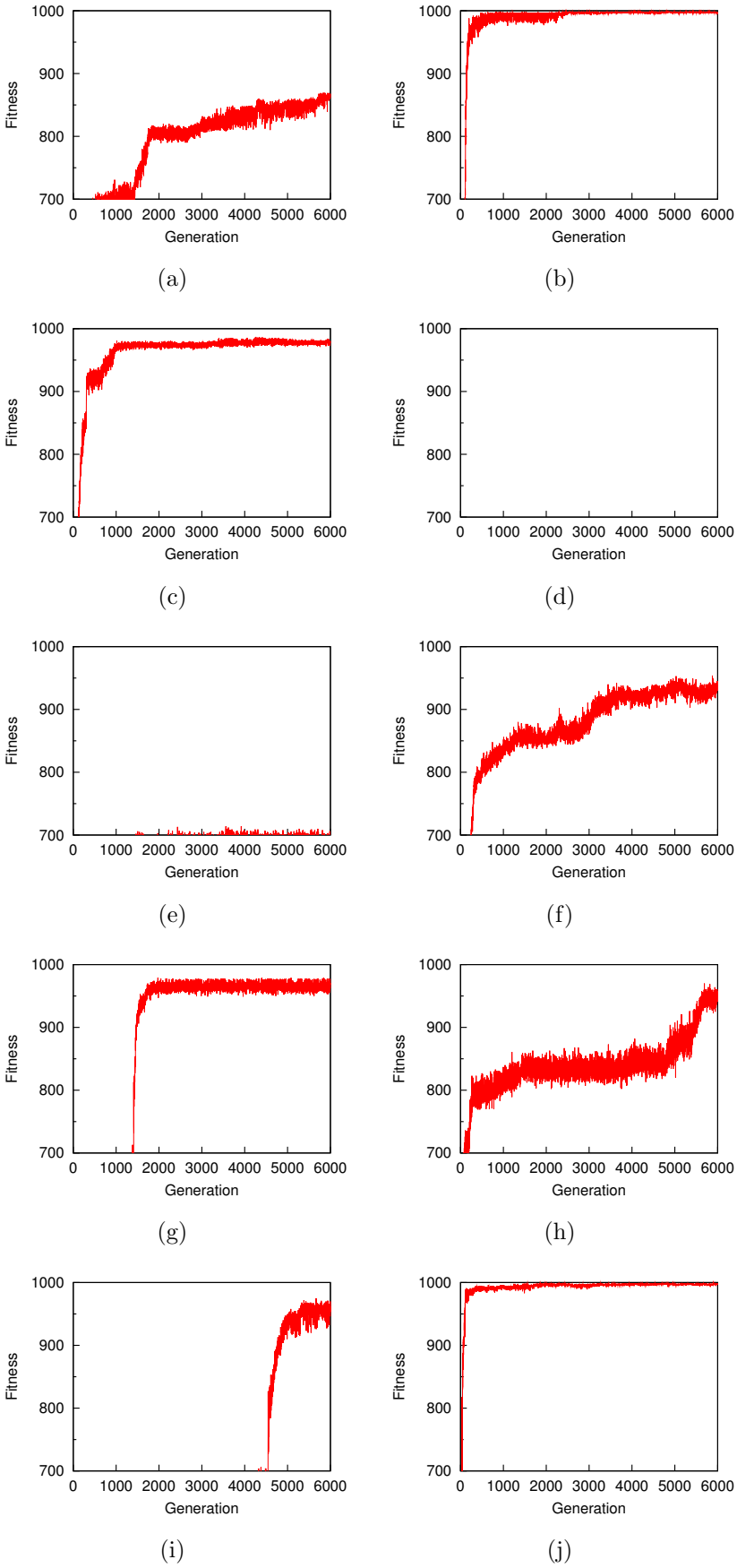


Figure E.9: Maximum fitness at each generation in each run by the SGA without elitism for $s = 2$ and $Nh = 10$

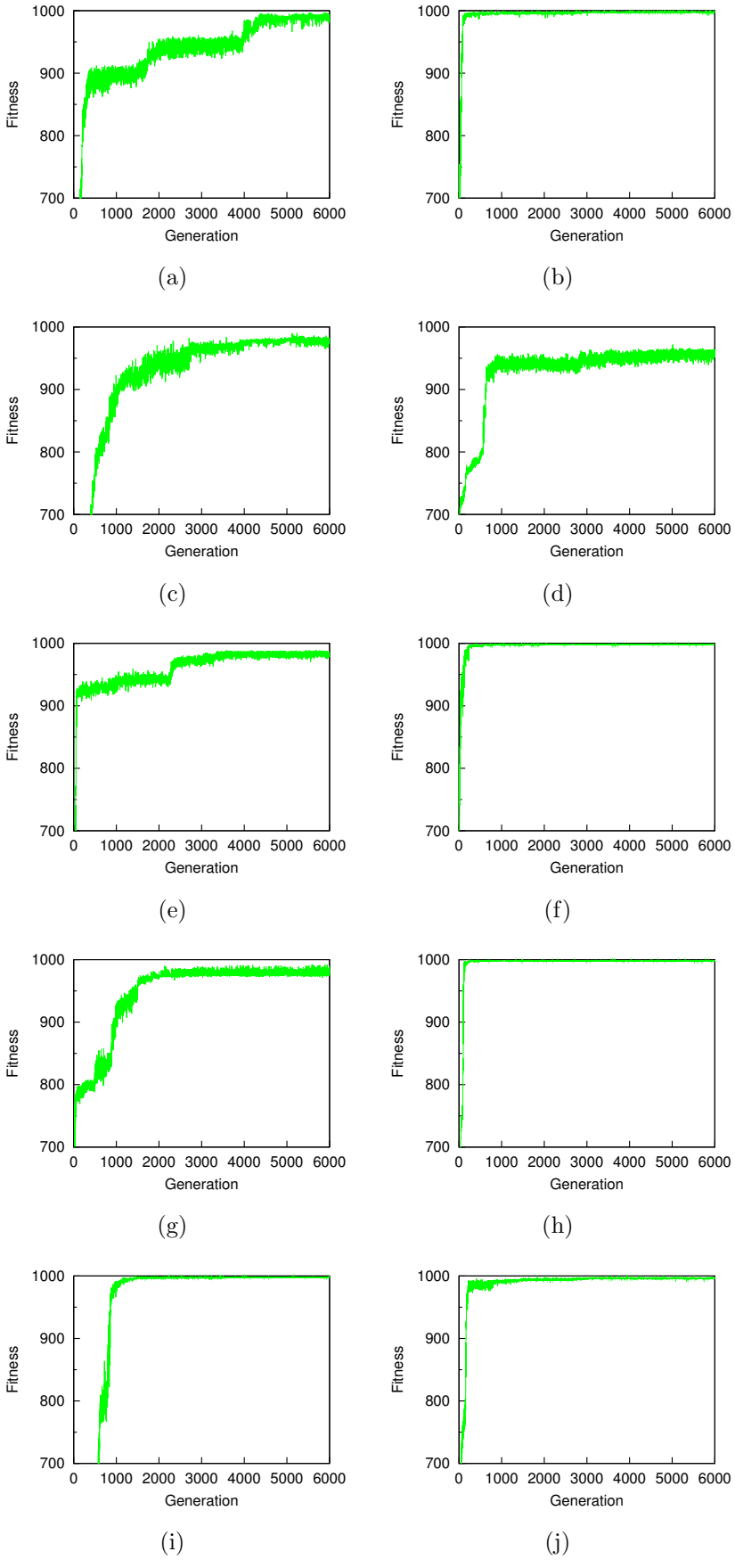


Figure E.10: Maximum fitness at each generation in each run by the OGA without elitism for $s = 2$ and $Nh = 10$

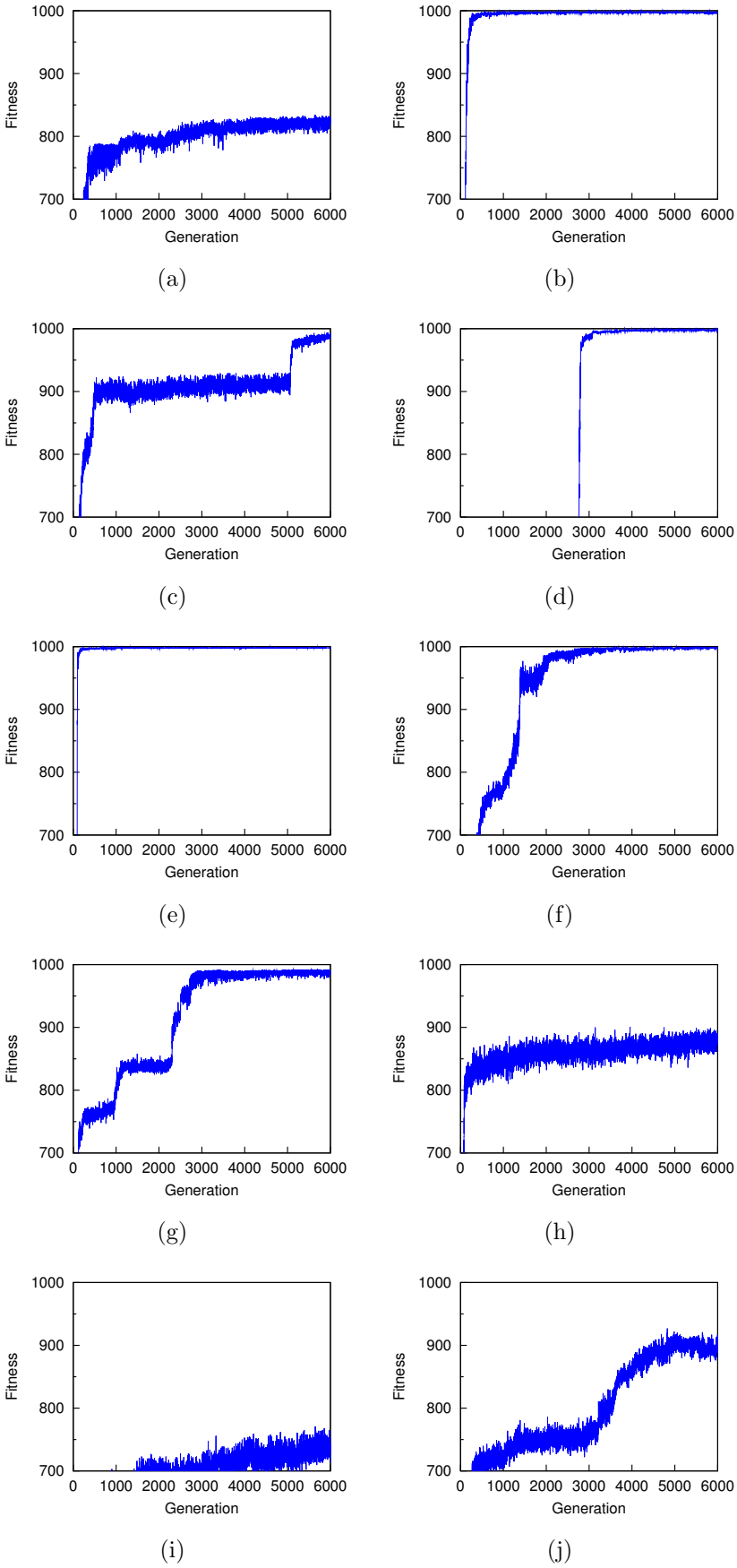


Figure E.11: Maximum fitness at each generation in each run by the SGA with elitism for $s = 2$ and $Nh = 10$

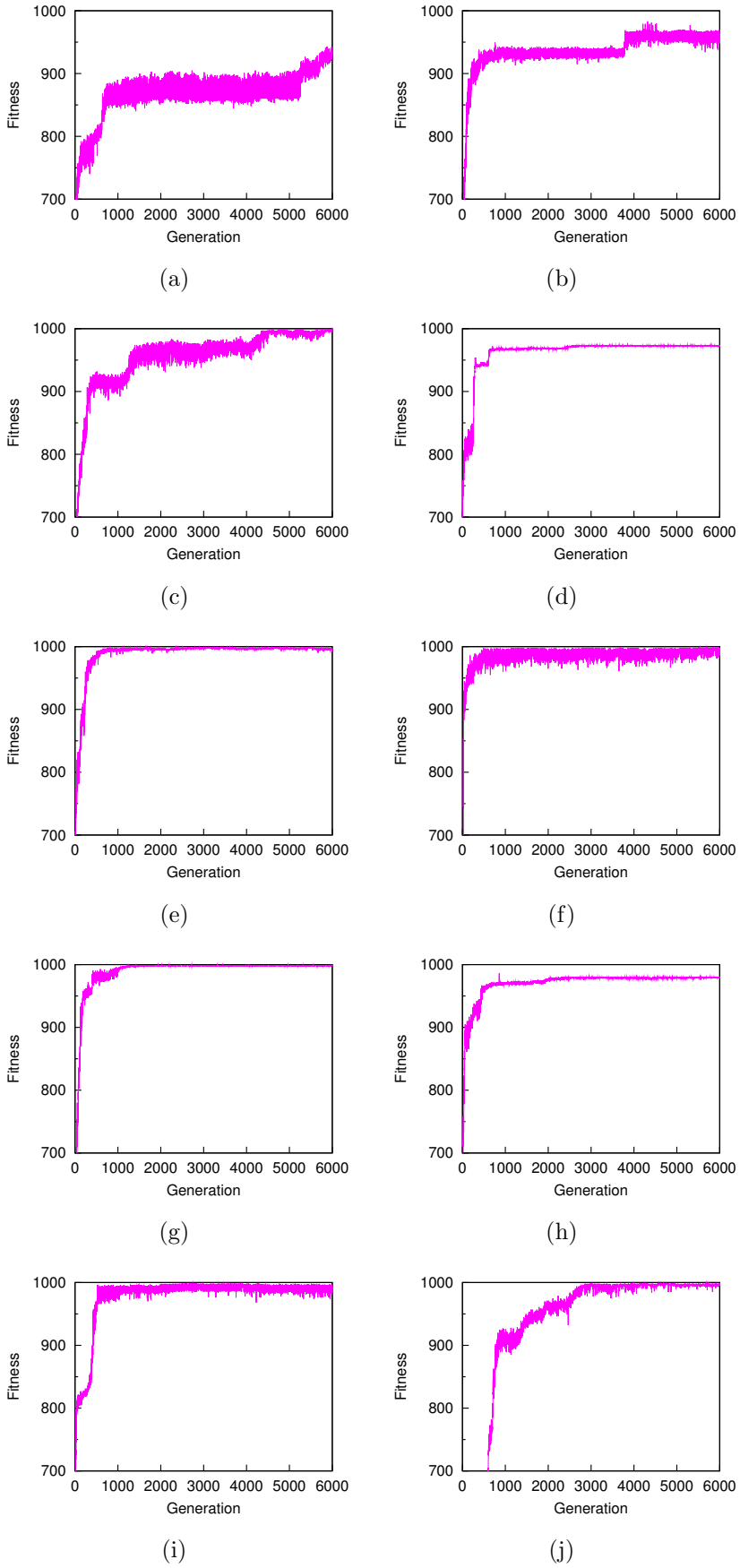


Figure E.12: Maximum fitness at each generation in each run by the OGA with elitism for $s = 2$ and $Nh = 10$

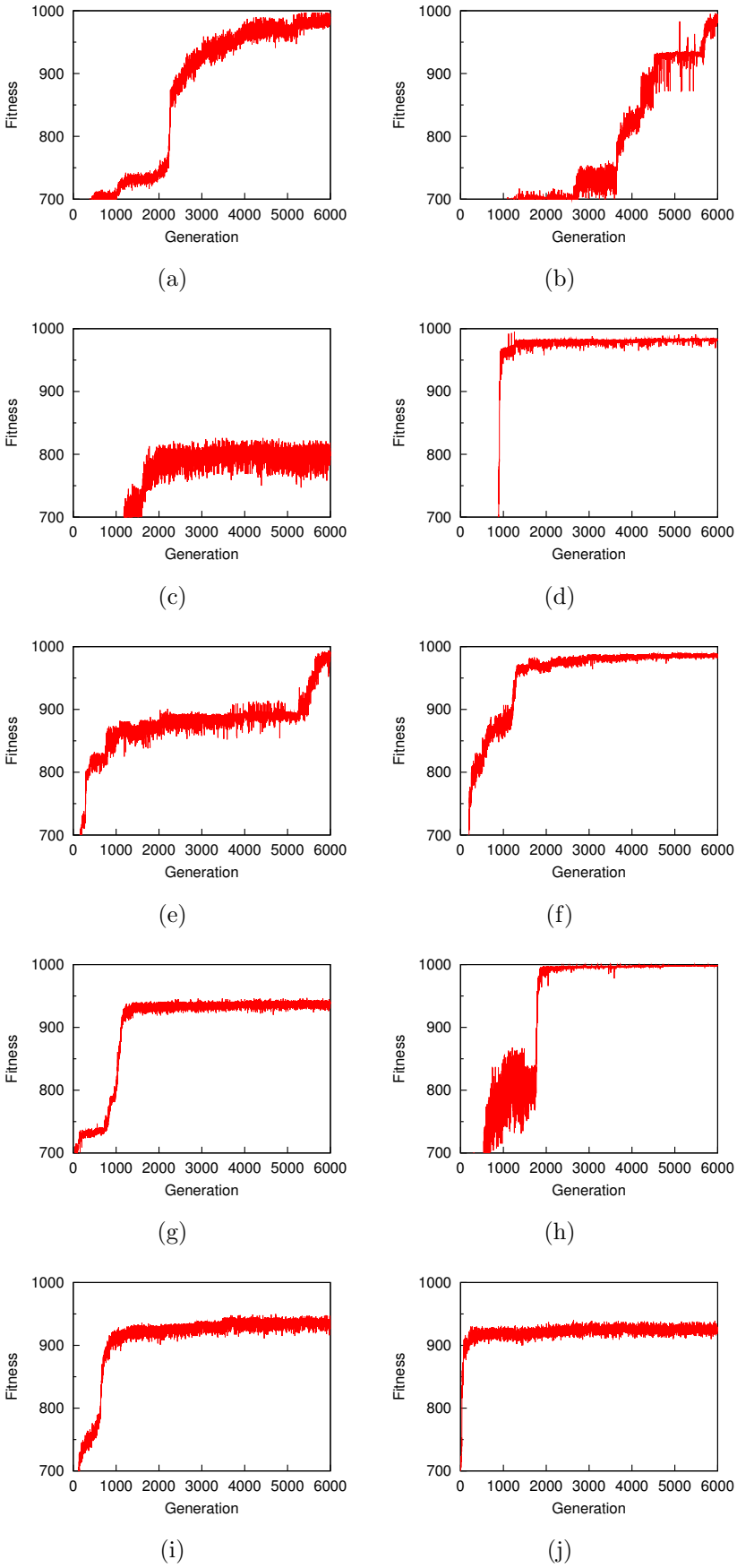


Figure E.13: Maximum fitness at each generation in each run by the SGA without elitism for $s = 6$ and $Nh = 10$

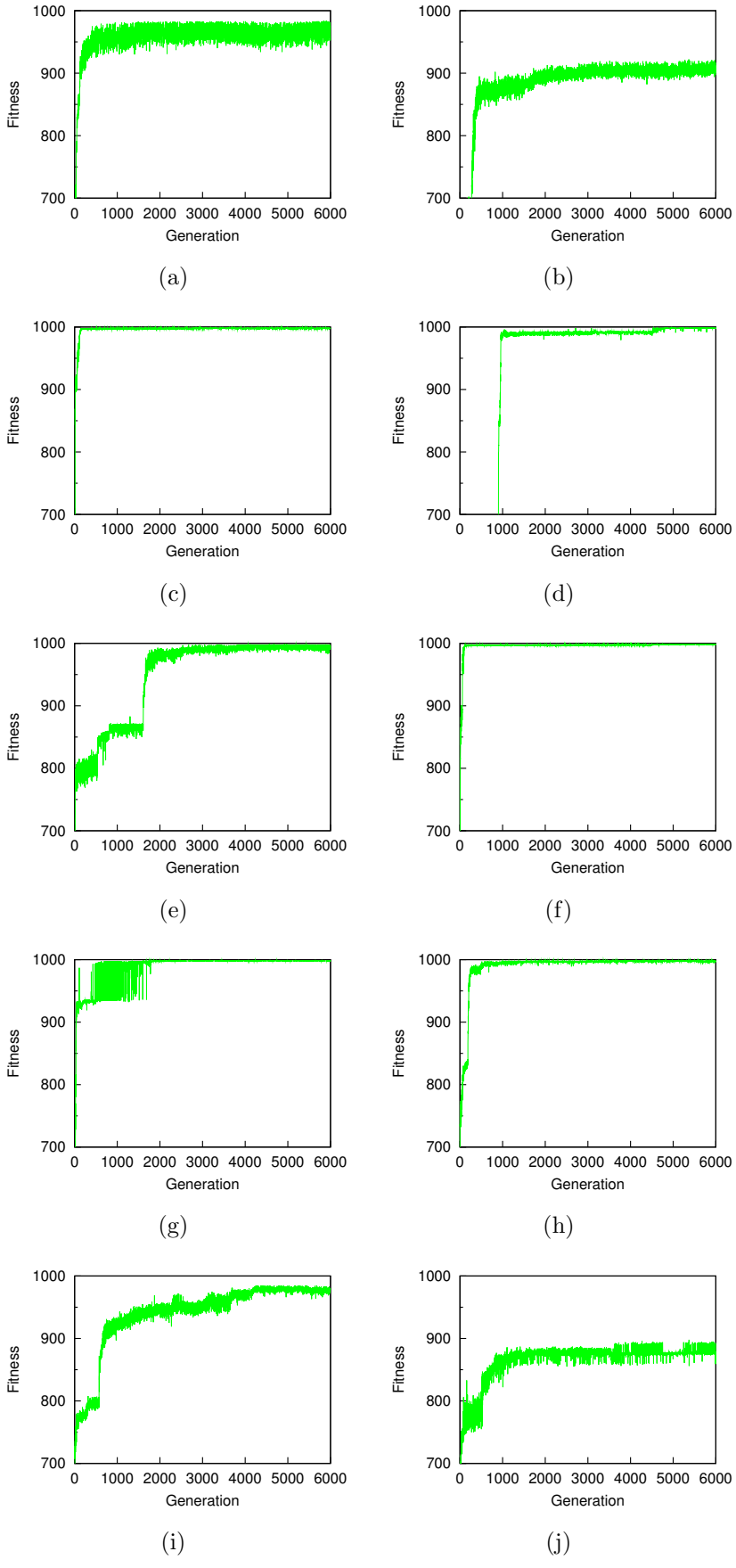


Figure E.14: Maximum fitness at each generation in each run by the OGA without elitism for $s = 6$ and $Nh = 10$

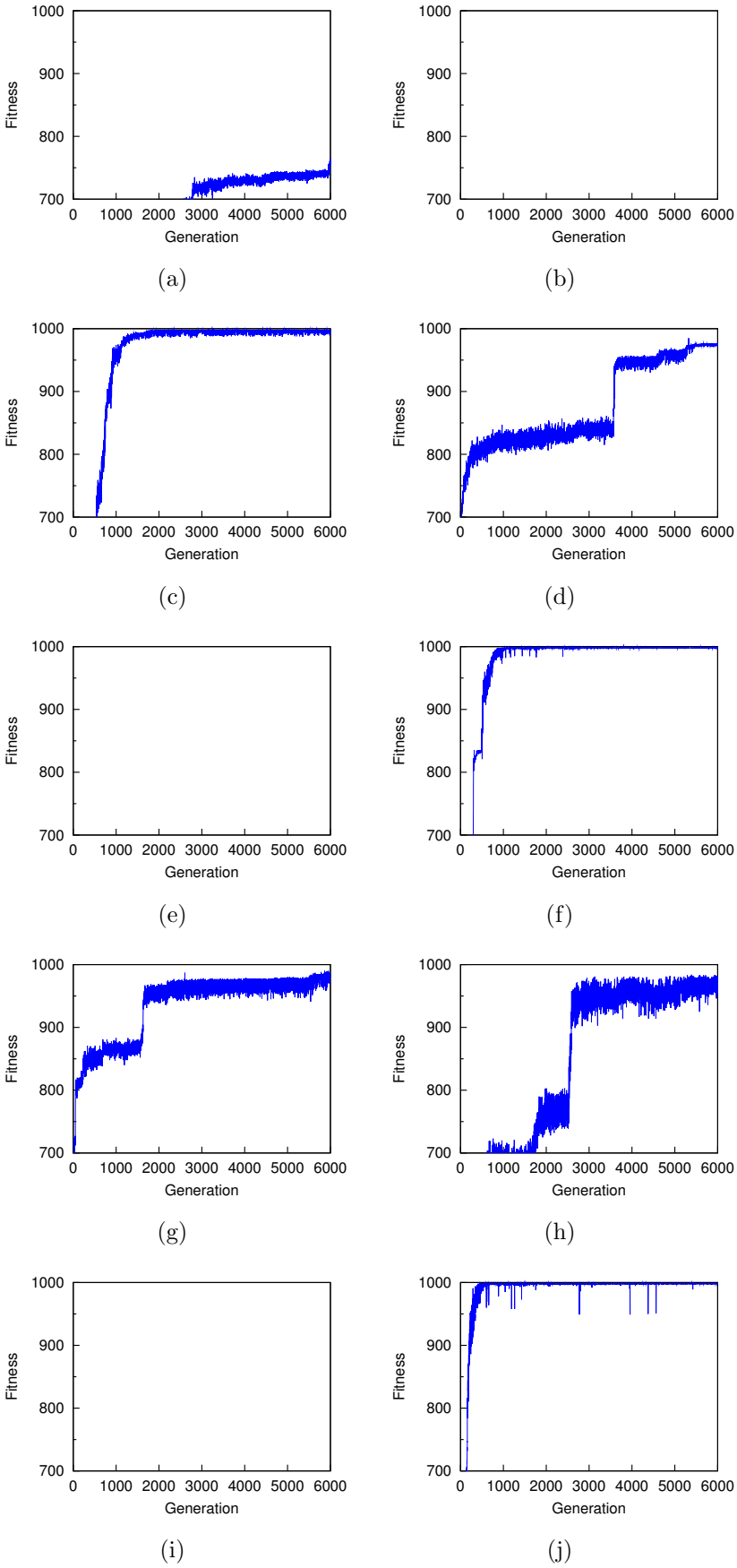


Figure E.15: Maximum fitness at each generation in each run by the SGA with elitism for $s = 6$ and $Nh = 10$

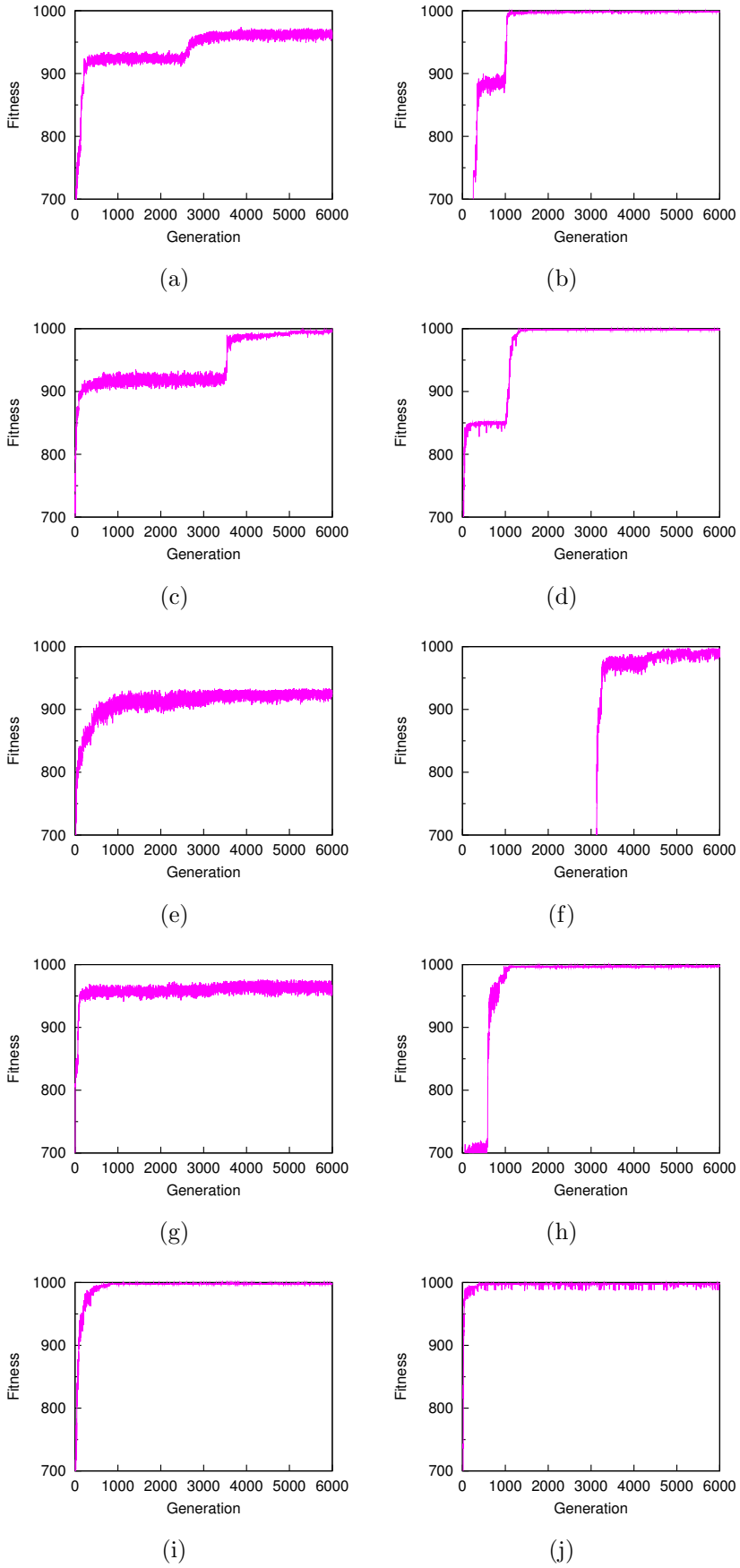


Figure E.16: Maximum fitness at each generation in each run by the OGA with elitism for $s = 6$ and $Nh = 10$

Appendix F

Evolutionary Dynamics in a Pursuit Problem

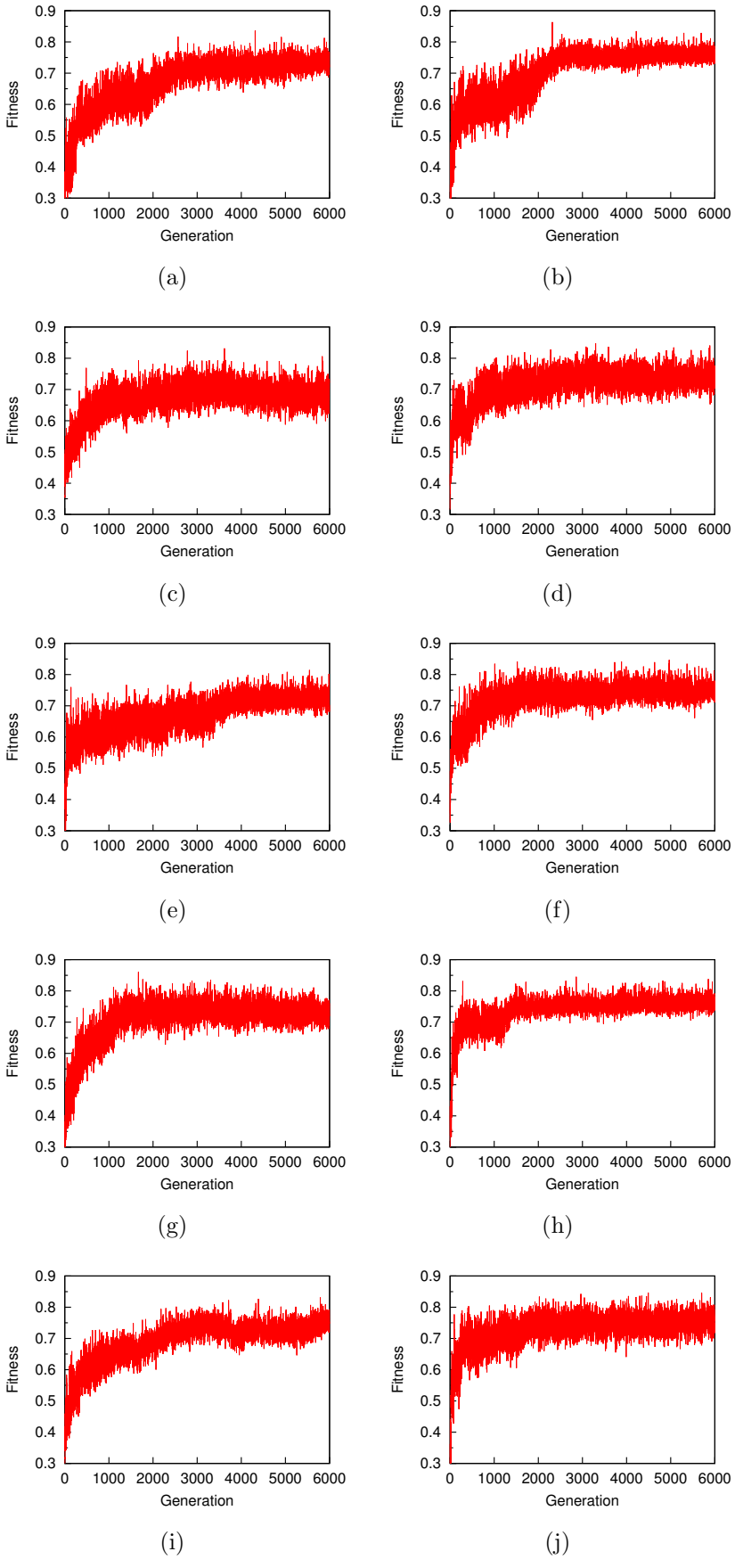


Figure F.1: Maximum fitness at each generation in each run by the SGA without elitism for s=2

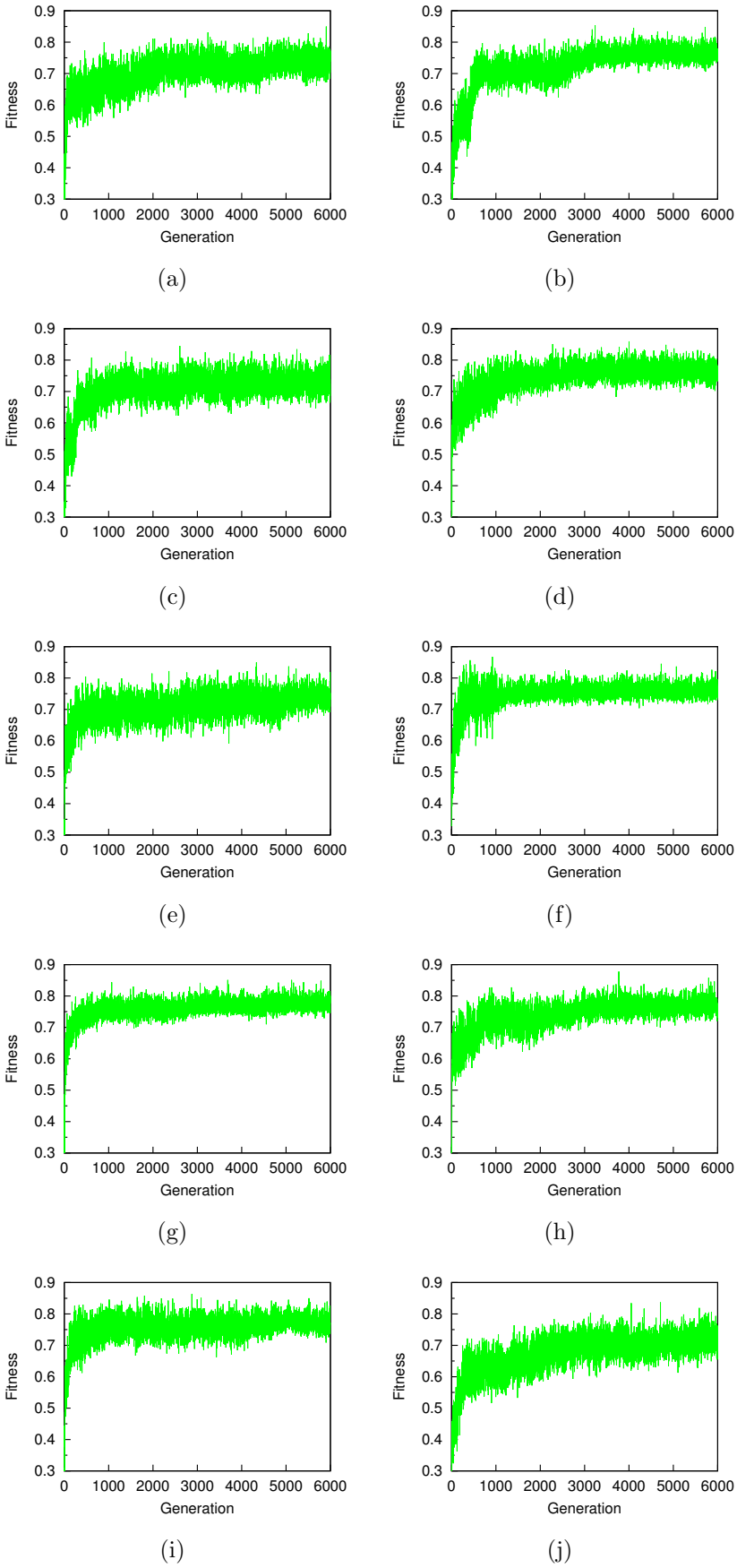


Figure F.2: Maximum fitness at each generation in each run by the OGA without elitism for s=2

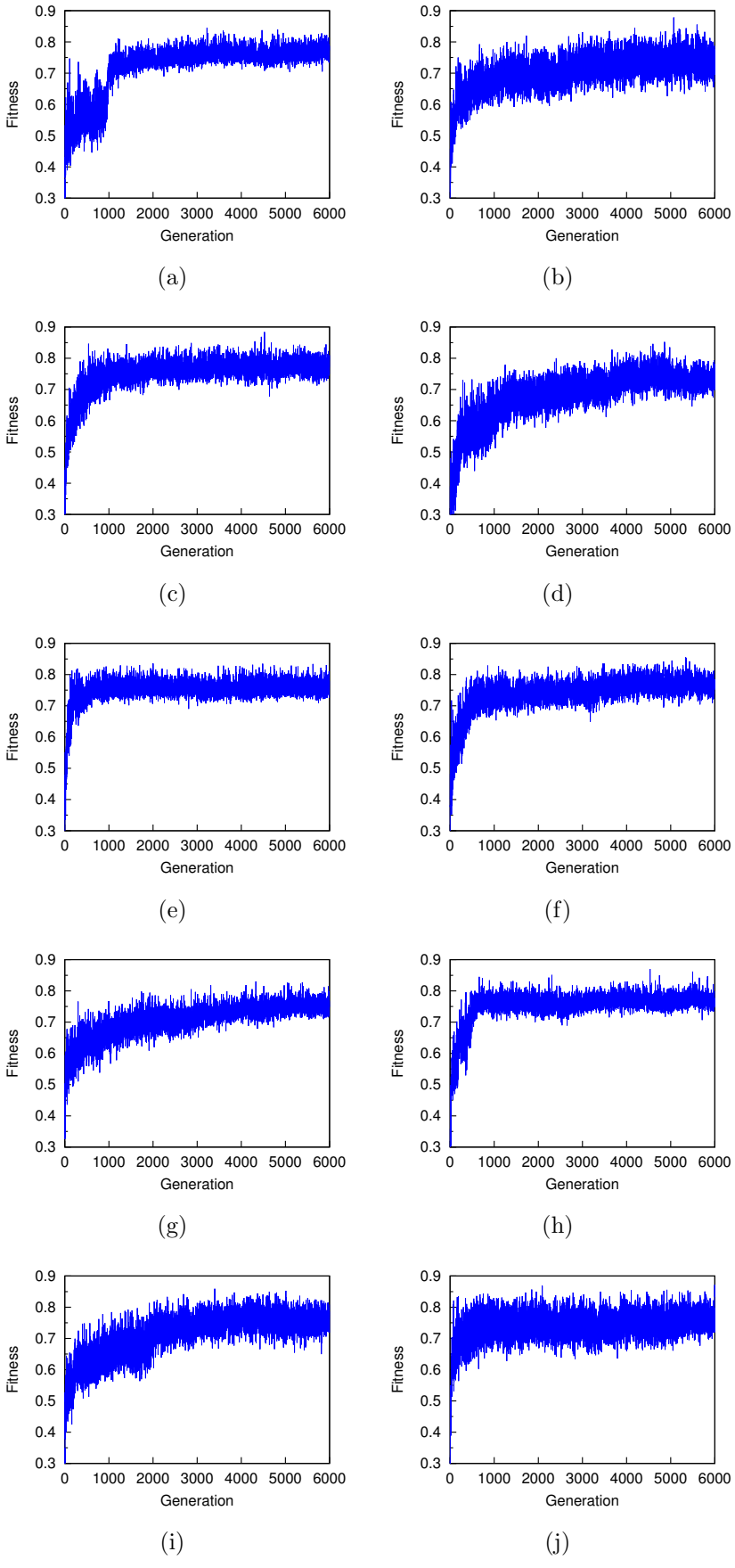


Figure F.3: Maximum fitness at each generation in each run by the SGA with elitism for s=2

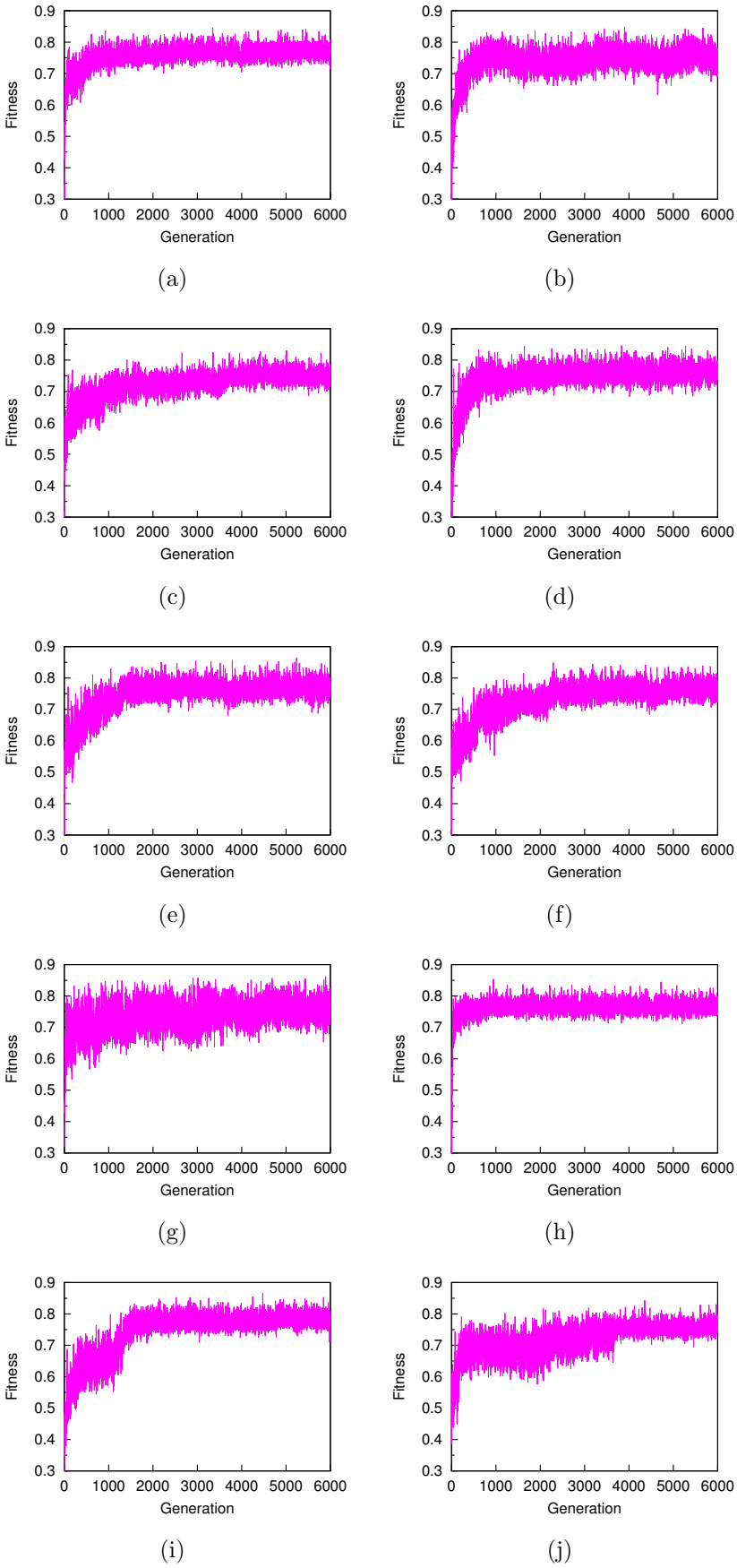


Figure F.4: Maximum fitness at each generation in each run by the OGA with elitism for s=2

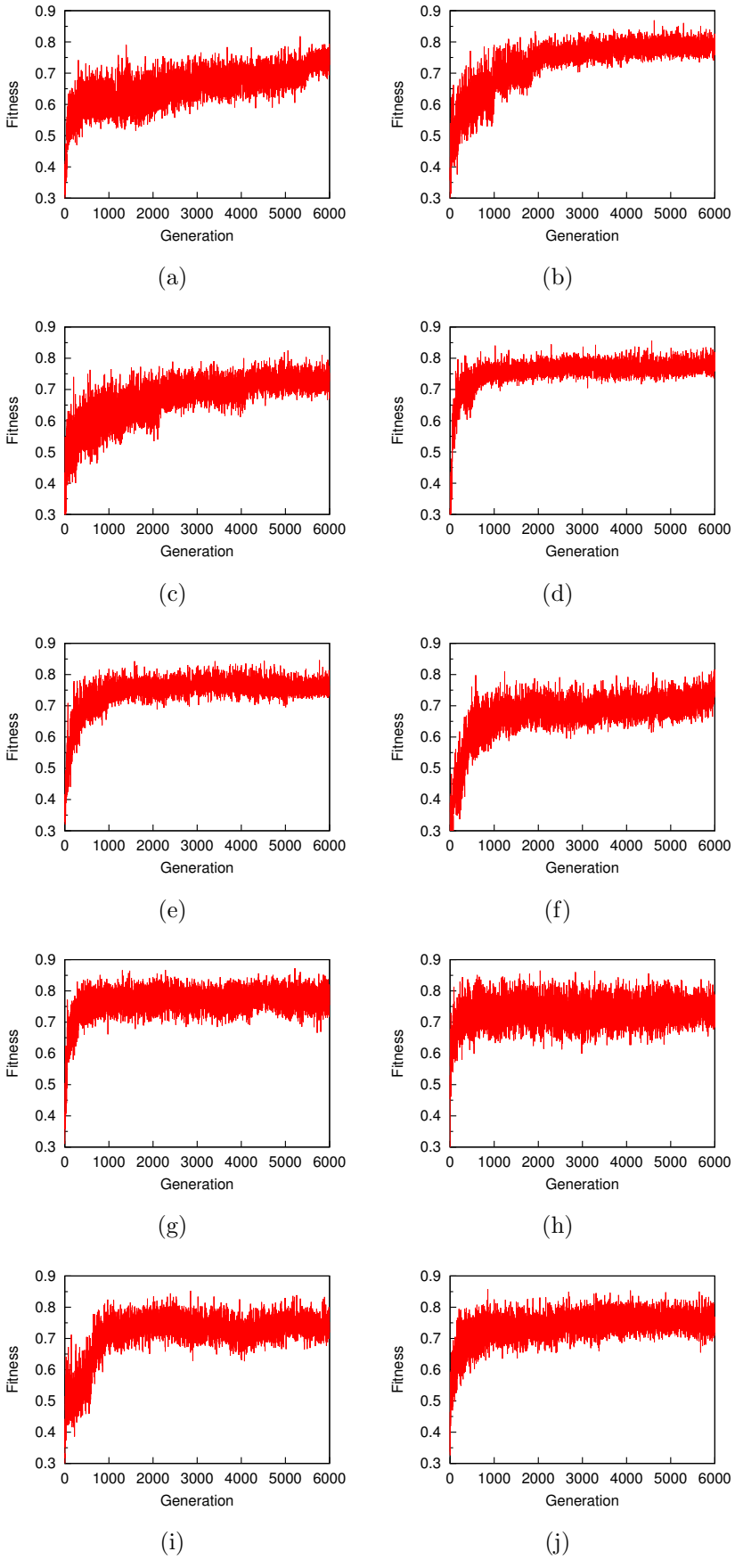


Figure F.5: Maximum fitness at each generation in each run by the SGA without elitism for s=6

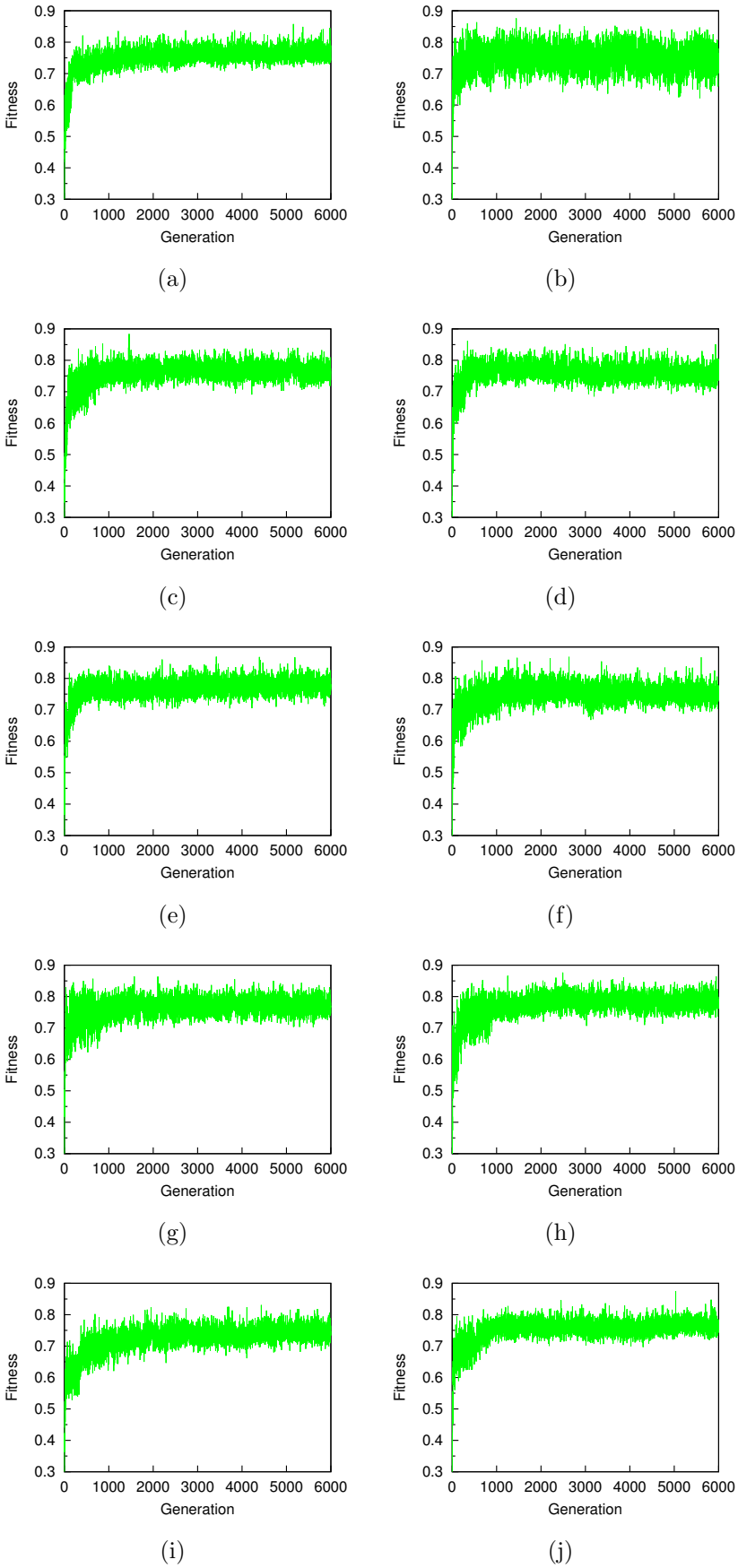


Figure F.6: Maximum fitness at each generation in each run by the OGA without elitism for s=6

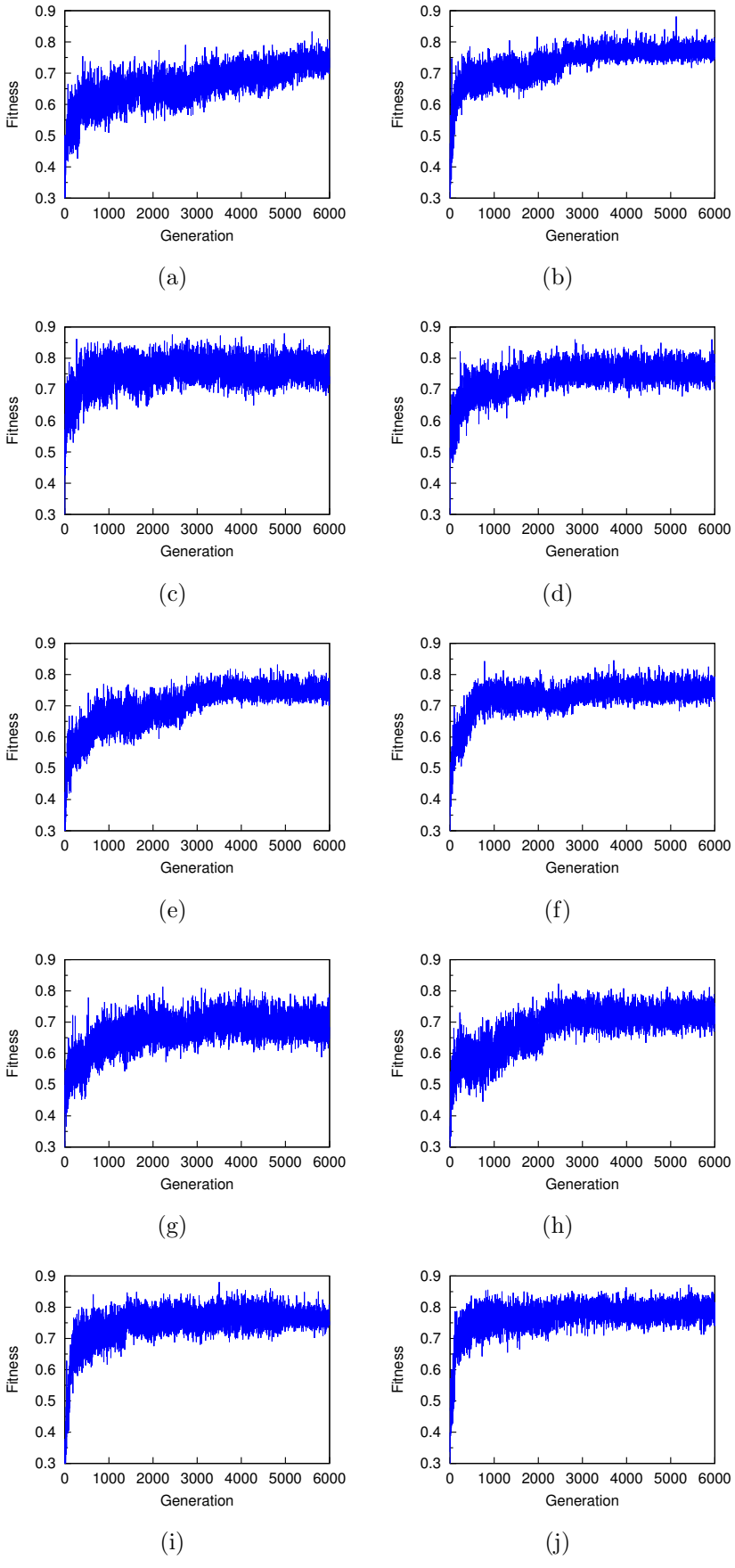


Figure F.7: Maximum fitness at each generation in each run by the SGA with elitism for s=6

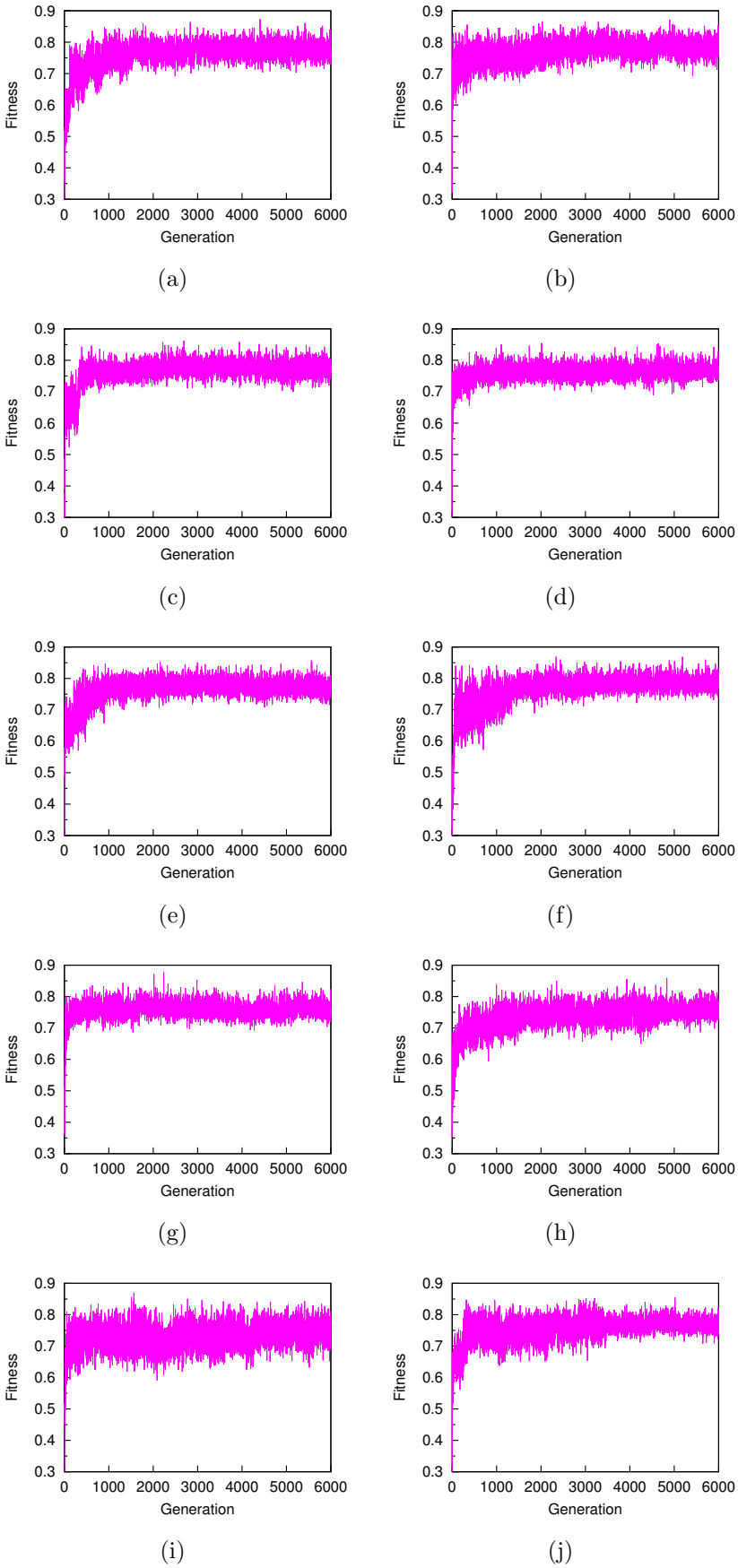


Figure F.8: Maximum fitness at each generation in each run by the OGA with elitism for s=6

Appendix G

Specification of a Real Robot

A two-wheeled robot was used in the experiment. The robot is 0.35 m in diameter. Two motor-driven wheels, made of foam-rubber, are located on either side of the robot, providing locomotion through differential drive, giving the robot a top speed of around 0.34 m/s. Two passive caster wheels, placed front-center and rear-center, ensure stability. A robot's source of sensory input comes from an omnidirectional camera. The robot has a CPU board for low-level(motor/sensor) control and a laptop computer(CPU: Pentium III 800 MHz, Memory: 376 MB) loaded on the robot for high-level control(decision making).

The details of components are as follows.

Motor Every wheel is moved by a TAMIYA DC motor 3633K75 coupled with the wheel directly. The theoretical maximum torque is 3.0×10^3 kg/m and the number of revolutions is 156 rpm.

Servo controller The MiniSSC II from Scott Edwards Electronics is a servo motor controller with a simple serial interface(2400 or 9600 baud). Each MiniSSC can control up to 8 servos at once. MiniSSC has switchable range/resolution(90°/0.36° or 180°/0.72°). MiniSSC converts motor control signals from a CPU board into pulse waves and sends them to the motor controller.

Motor controller TAMIYA Digital Twin Motor Differential Control Unit T-01 receives the pulse waves from MiniSSC and controls the number of revolutions of

motors. T-01 conducts the operation in two channels.

Omnidirectional camera SUEKAGE omnidirectional camera SOIOS 55-cam for ROBOT has a hyperbolic mirror and a catadioptric panoramic camera(the number of pixels: 0.38M, the maximum frame rate: 30 fps, resolution: 640 $\times$ 480 pixels). The mirror is 55.0 $\times$ 10^{-3} m in diameter and the valid angle of view is 12°(elevation)/50°(depression). The camera has IEEE1394 connection to a laptop computer.

CPU board YellowSoft YS50-1 has a single-chip RISC(Reduced Instruction Set Computer) with SH-2 core(26 MIPS, 20 MHz, 5.0 V). SH-2 has a built-in 32-bit multiplier and a built-in flash memory (128K) with single power supply/RAM (6K), and has enhanced peripheral functions(Timer:ATU(FRT: 10ch, DC: 8ch), Compare match timer: 2ch, A/D converter: 10 bits $\times$ 16ch, Serial: 3ch, DMAC: 4ch).

Battery The robot is equipped with a 7.2V rechargable Nickel-hydrogen battery for motors, and a 8.4V rechargable Nickel-Cadmium battery for a CPU board and a servo controller. These batteries allow the robot to move without external power supplies by wired connections.

Appendix H

List of Publications

Journal Papers

1. 片田 喜章, 大倉 和博, 上田 完次
   ″Neutral Networksを含む適応度景観における遺伝的アルゴリズムの進化ダイナミクス″,
   システム制御情報学会論文誌, Vol.17, No.5, pp. 187-195, 2004.

2. Y. Katada, K. Ohkura and K. Ueda,
   "Tuning Genetic Algorithms for Problems Including Neutral Networks",
   Information Sciences(Elsevier) (submitted)

3. 片田 喜章, 大倉 和博, 上田 完次
   ″根井の標準遺伝距離を用いた適応度景観におけるneutralityの推定″,
   システム制御情報学会論文誌, （投稿予定）.

International Conferences

1. Y. Katada, Y. Yasuda, K. Ohkura and K. Ueda, "Evolutionary Dynamics on Neutral Networks", Proceedings of International Workshop on Emergent Synthesis (IWES), pp.35-40, (2002).
2. Y. Katada, K. Ohkura and K. Ueda, "The effect of population size on evolutionary speed on the problems including neutral networks", Proceedings of The 6th International Conference on Complex Systems (CS02), pp.353-360, (2002).
3. Y. Katada,K. Ohkura and K. Ueda, "Artificial Evolution of Pulsed Neural Networks on the Motion Pattern Classification System", Proceeding 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), pp.318-323, (2003).
4. Y. Katada, K. Ohkura and K. Ueda, "Tuning Genetic Algorithms for Problems Including Neutral Networks -The Simplest Case: The Balance Beam Function-", The Sixth International Conference on Computational Intelligence and Natural Computing (CINC), Proceedings of The 7th Joint Conference on Information Sciences, pp.1657-1660, (2003).
5. Y. Katada, K. Ohkura and K. Ueda, "Tuning Genetic Algorithms for Problems Including Neutral Networks -A More Complex Case: The Terraced NK Problem-", The Sixth International Conference on Computational Intelligence and Natural Computing (CINC), Proceedings of The 7th Joint Conference on Information Sciences, pp.1661-1664, (2003).
6. Y. Katada, K. Ohkura and K. Ueda, "Measuring Neutrality of Fitness Landscapes Based on the Nei's Standard Genetic Distance", Proceedings of 2003 Asia Pacific Symposium on Intelligent and Evolutionary Systems: Technology and Applications, pp.107-114, (2003).
7. Y. Katada, K. Ohkura and K. Ueda, "An Approach to Evolutionary Robotics Problems from the Genetic Algorithm with the Variable Mutation Rate Strategy", Proceedings of The 5th International Workshop on Emergent Synthesis (IWES), pp.117-124, (2004).
8. Y. Katada, K. Ohkura and K. Ueda, "The Nei's Standard Genetic Distance in Artificial Evolution", Proceedings of the 2004 IEEE Congress on Evolutionary Computation (CEC2004), pp.1233-1239, (2004).
9. Y. Katada, K. Ohkura and K. Ueda, "An Approach to Evolutionary Robotics Using the Genetic Algorithm with Variable Mutation Rate Strategy", Parallel Problem Solving from Nature (PPSN VIII), (2004) (will be published).

National Conferences

1. 片田喜章, 大倉和博, 上田完次
"ニュートラルネットワーク環境下におけるGAの進化速度に関する一考察",
第46回システム制御情報学会研究発表講演会講演論文集, pp.585-586, (2002).

2. 片田喜章, 大倉和博, 上田完次
"適応度景観におけるneutralityの測定に関する一考察",
2003年度精密工学会春季大会学術講演論文集, pp.29, (2003).

3. 片田喜章, 大倉和博, 上田完次
"進化型ニューラルネットワークにおけるNeutralityとRuggedness",
第47回システム制御情報学会研究発表講演会講演論文集, pp.687-688, (2003).

4. 片田喜章, 大倉和博, 上田完次
"ノイズを含むneutral networks環境下における遺伝的アルゴリズムの進化ダイナミクス",
第46回自動制御連合講演会論文集, (TA2-02-3) pp.229-232, (2003).

5. 片田喜章, 大倉和博, 上田完次
"可変突然変異率戦略を用いたGAによる進化ロボティクスへのアプローチ",
第48回システム制御情報学会研究発表講演会講演論文集, pp.587-588, (2004).

6. 足立昌彦, 片田喜章, 大倉和博, 田浦俊春
"Neutral networks環境下におけるsteady state GAの進化ダイナミクス",
第48回システム制御情報学会研究発表講演会講演論文集, pp.585-586, (2004).

7. 高崎真也, 片田喜章, 大倉和博, 田浦俊春
"オンラインモデル更新を用いた自律移動ロボットの進化に関する基礎研究",
ロボティクス・メカトロニクス講演会'04,講演論文集 CD-ROM(ROBOMEC'04),
2P2-L1-23, (2004).

Bibliography

Babajide, A., Hofacker, I. L., Sippl, M. J., Stadler, P. F., 1997. Neutral networks in protein space, a computational study based on knowledge-based potentials of mean force. Folding & Design 2, 261–269.
URL citeseer.ist.psu.edu/babajide97neutral.html

Barnett, L., 1997. Tangled webs: Evolutionary dynamics on fitness landscapes with neutrality. Master's thesis, School of Cognitive and Computing Sciences, Sussex University, Brighton, UK.
URL citeseer.nj.nec.com/barnett97tangled.html

Barnett, L., 1998. Ruggedness and neutrality – the NKp family of fitness landscapes. In: Adami, C., Belew, R. K., Kitano, H., Taylor, C. (Eds.), Artificial Life VI: Proc. of the Sixth Int. Conf. on Artificial Life. The MIT Press, Cambridge, MA, pp. 18–27.
URL citeseer.nj.nec.com/144047.html

Barnett, L., May 2001. Netcrawling - optimal evolutionary search with neutral networks. In: Proceedings of the 2001 IEEE Congress on Evolutionary Computation: CEC2001. IEEE Press, Seoul, Korea, pp. 30–37.

Beer, R., 1996. Toward the evolution of dynamical neural networks for minimally cognitive behavior. In: Maes, P., Mataric, M., Meyer, J., Pollack, J., Wilson, S. (Eds.), Proceedings of From Animals to Animats 4. MIT press, Cambridge, pp. 421–429.

Braitenberg, V., 1996. Vehicles: Experiments in Synthetic Psychology. MIT Press.

Bullock, S., 2002. Will selection for mutational robustness significantly retard evolutionary innovation on neutral networks? In: Standish, R., Bedau, M., Abbass, H. (Eds.), Proceedings of the Eighth International Conference on Artificial Life. MIT Press, Sydney, Australia, pp. 192–201.

Ebner, M., Langguth, P., Albert, J., Shackleton, M., Shipman, R., 2001. On neutral networks and evolvability. In: Proceedings of the 2001 IEEE Congress on Evolutionary Computation: CEC2001. IEEE Press, Piscataway, New Jersey, pp. 1–8.

Eigen, M., McCaskill, J., Schuster, P., 1989. The molecular quasi-species. Advances in Chemical Physics 75, 149–263.

Floreano, D., Mattiussi, C., 2001. Evolution of spiking neural controllers. In: Gomi, T. (Ed.), Proceeding of Evolutionary Robotics -From Intelligent Robots to Artificial Life (ER'01). Springer-Verlag, pp. 38–61.

Floreano, D., Nolfi, S., 13-16 1997. God save the red queen! competition in co-evolutionary robotics. In: Koza, J. R., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M., Iba, H., Riolo, R. L. (Eds.), Genetic Programming 1997: Proceedings of the Second Annual Conference. Morgan Kaufmann, Stanford University, CA, USA, pp. 398–406.
URL `citeseer.ist.psu.edu/964.html`

Forst, C. V., Reidys, C., Weber, J., 1995. Evolutionary dynamics and optimization: Neutral networks as model-landscapes for RNA secondary-structure folding-landscapes. In: Proceedings of the Third European Conference on Artificial Life. pp. 128–147.
URL `citeseer.nj.nec.com/forst95evolutionary.html`

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Gould, S. J., Eldredge, N., 1977. Punctuated equilibria: the temp and mode of evolution reconsidered. Paleobiology 3, 115–251.

Gruner, W., Giegerich, R., Strothmann, D., Reidys, C., Weber, J., Hofacker, I., Stadler, P., Schuster, P., 1996. Analysis of RNA sequence structure maps by exhaustive enumeration: Neutral networks. Monatchefte fur Chemie 127, 355–374.

Harvey, I., 1997. Artificial evolution for real problems. In: Gomi, T. (Ed.), Evolutionary Robotics: From Intelligent Robots to Artificial Life (ER'97). AAI Books, Tokyo.

Harvey, I., Thompson, A., 1996. Through the labyrinth evolution finds a way: A silicon ridge. In: Proceedings of the first International Conference on Evolvable Systems: From Biology to Hardware. pp. 406–422.

Holland, J. H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press.

Hordijk, W., 1994. A measure of landscapes. Evolutionary Computation 4 (4), 335–360.

Horn, J., Goldberg, D. E., 1995. Genetic algorithm difficulty and the modality of fitness landscapes. In: Whitley, L. D., Vose, M. D. (Eds.), Foundations of Genetic Algorithms 3. Morgan Kaufmann, San Francisco, CA, pp. 243–269.
URL `citeseer.ist.psu.edu/horn94genetic.html`

Huynen, M., Stadler, P., Fontana, W., 1996. Smoothness within ruggedness: The role of neutrality in adaptation. In: Proc. Natl. Acad. Sci. Vol. 93. USA, pp. 397–401.

Jones, T., Forrest, S., 1995. Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In: Eshelman, L. J. (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, California, pp. 184–192.

Jörg-Peter, E., 1976. NEUROETHOLOGY. Springer-Verlag, Berlin Heidelberg, New York.

Kallel, L., 1998. Inside GA dynamics: Ground basis for comparison. In: Proceedings of the 4th Conference on Parallel Problem Solving from Nature. Lecture Notes in Computer Science, pp. 57–66.

Katada, Y., Ohkura, K., Ueda, K., 2003. Artificial evolution of pulsed neural networks on the motion pattern classification system. In: *Proceedings of 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation(CIRA)*. pp. 318–323.

Kauffman, S. A., 1993. The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, New York.

Kimura, M., 1983. The Neutral Theory of Molecular Evolution. Cambridge University Press, New York.

Kimura, M., Ohta, T., 1974. On some principles governing molecular evolution. Proceedings of the Academy of Natural Sciences 71 (7), 2848–2852.

Knowles, J. D., Watson, R. A., September 2002. On the utility of redundant encodings in mutation-based evolutionary search. In: Merelo, J., Admidis, P., Beyer, H.-G., Fernandes-Villacanas, J.-L., Schwefel, H.-P. (Eds.), Proceedings of Parallel Problem Solving from Nature - PPSN VII, Seventh International Conference. LNCS 2439, Granada, Spain, pp. 88–98.

Komoriya, K., Oyama, E., Tani, K., 1993. Planning of landmark measurement for the navigation of a mobile robot. Journal of the Robotics Society of Japan 11 (4), 533–540 (in Japanese).

Maass, W., Bishop, C., 1998. Pulsed Neural Networks. MIT press.

Manderick, B., Weger, M. D., Spiessens, P., 1991. The genetic algorithm and the structure of the fitness landscape. In: Belew, R. K., Booker, L. B. (Eds.), Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo, California, pp. 143–150.

Mitchell, M., Forrest, S., Holland, J. H., 11–13 1992. The royal road for genetic algorithms: Fitness landscapes and GA performance. In: Varela, F. J., Bourgine, P. (Eds.), Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, 1991. A Bradford Book, The MIT Press, Paris, pp. 245–254.
URL citeseer.nj.nec.com/mitchell91royal.html

Mühlenbein, H., 1993. Predictive models for the breeder genetic algorithm. Evolutionary Computation 1 (1), 25–49.

Nei, M., 1972. Genetic distance between populations. The American Naturalist 106, 283–292.

Newman, M., Engelhardt, R., 1998. Effect of neutral selection on the evolution of molecular species. In: Proc. R. Soc. London B. pp. 256:1333–1338.

Nolfi, S., Floreano, D., 2000. Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. MIT Press.

Ochoa, G., Harvey, I., Buxton, H., 1999. Error thresholds and their relation to optimal mutation rates. In: European Conference on Artificial Life. pp. 54–63.
URL citeseer.nj.nec.com/202913.html

Ohkura, K., Ueda, K., 1999. Adaptation in dynamic environment by using GA with neutral mutations. International Journal of Smart Engineering System Design 2, 17–31.

Ohta, T., 1992. The nearly neutral theory of molecular evolution. Annu. Rev. Ecol. Syst., 23:263–286.

Ohta, T., 1997. Role of Random Genetic Drift in the Evolution of Interactive Systems. Journal of Molecular Evolution 44, S9–S14.

Ohta, T., 1998. Evolution by nearly-neutral mutations. Genetica 102/103, 83–90.

Pfeifer, R., Scheier, C., 1999. Understanding Intelligence. MIT Press, Cambridge.

Reidys, C., Stadler, P., Schuster, P., 1997. Generic properties of combinatory maps - neutral networks of RNA secondary structures. Bull. math. biol 59, 339–397.

Rothlauf, F., Goldberg, D. E., 2003. Redundant representations in evolutionary computation. Evolutionary Computation 11 (4), 381–415.

Schuster, P., Fontana, W., Stadler, P., Hofacker, I., 1994. From sequences to shapes and back: A case study in RNA secondary structures. Proceedings of the Royal Society of London B (255), 279–284.

Shackleton, M., Shipman, R., Ebner, M., 6-9 2000. An investigation of redundant genotype-phenotype mappings and their role in evolutionary search. In: Proceedings of the 2000 Congress on Evolutionary Computation CEC00. IEEE Press, La Jolla Marriott Hotel La Jolla, California, USA, pp. 493–500.
URL `citeseer.ist.psu.edu/409243.html`

Shipman, R., Shackleton, M., Ebner, M., Watson, R., 2000. Neutral search spaces for artificial evolution: A lesson from life. In: Bedau, M., McCaskill, J., Packard, N., Rasmussen, S. (Eds.), Proceedings of Artificial Life VII. MIT Press, pp. 162–169.

Smith, T., 2002. The evolvability of artificial neural networks for robot control. Ph.D. thesis, School of Biological Sciences, University of Sussex.

Smith, T., Husbands, P., Layzell, P., O'Shea, M., 2002a. Fitness landscapes and evolvability. Evolutionary Computation 10 (1), 1–34.

Smith, T., Husbands, P., O'Shea, M., 2001a. Neutral networks and evolvability with complex genotype-phenotype mapping. In: Proceedings of the European Conference on Artificial Life: ECAL2001. Springer, pp. 23–36.

Smith, T., Husbands, P., O'Shea, M., 2001b. Neutral networks in an evolutionary robotics search space. In: Proceedings of the 2001 IEEE Congress on Evolutionary Computation: CEC2001. IEEE Press, Piscataway, New Jersey, pp. 136–145.

Smith, T., Philippides, A., Husbands, P., O'Shea, M., 2002b. Neutrality and ruggedness in robot landscapes. In: Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002). CEC, IEEE Press, pp. 1348–1353.

Stadler, P. F., 1996. Landscapes and their correlation functions. J. Math. Chem. 20, 1–45.
URL citeseer.ist.psu.edu/43461.html

Thompson, A., 1996. An evolved circuit, intrinsic in silicon, entwined with physics. In: Proceedings of the first International Conference on Evolvable Systems:From Biology to Hardware. pp. 390–405.

van Nimwegen, E., Crutchfield, J. P., 1998. Optimizing epochal evolutionary search: Population-size dependent theory. SFI Working Paper 9810-090.

van Nimwegen, E., Crutchfield, J. P., Mitchell, M., 1999. Statistical dynamics of the royal road genetic algorithm. Theoretical Computer Science 229 (1), 41–102.
URL citeseer.nj.nec.com/nimwegen98statistical.html

Vassilev, V. K., Miller, J. F., 2000. The advantages of landscape neutrality in digital circuit evolution. In: Miller, J. e. a. (Ed.), Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware (ICES'2000). Lecture Notes in Computer Science, Springer, Berlin, Germany, pp. 1801:252–263.
URL citeseer.nj.nec.com/vassilev00advantages.html

Vassilev, V. K., Miller, J. F., Fogarty, T. C., 2000. Information characteristics and the structure of landscapes. Evolutionary Computation 8 (1), 31–60.

von. Uexküll, J., 1926. Theoretical Biology. Kegan Paul, Trench, Trubner & Co, London.

Weinberger, E. D., 1990. Correlated and uncorrelated fitness landscapes and how to tell the difference. Biological Cybernetics 63, 325–336.

Wright, S., 1932. The role of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones, D. (Ed.), Proceedings of the Sixth International Congress on Genetics. Vol. 1. pp. 356–366.

Yasuda, Y., Ohkura, K., Ueda, K., 2001. The balance beam function : A new test function for the real world problems. In: Asama, H., Inoue, H. (Eds.), Proceedings of Intelligent Autonomous Vehicles (IAV01). Elsevier Science, pp. 133–138.