Multi-Objective Evolutionary Algorithms for Ecological Process Models

Rie Komuro

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2005

Program Authorized to Offer Degree: Department of Applied Mathematics

University of Washington

Abstract

Multi-Objective Evolutionary Algorithms for Ecological Process Models

by Rie Komuro

Chair of the Supervisory Committee:

Professor E. David Ford
College of Forest Resources

Fitting an ecological process model to a set of data is frequently done by minimizing the residual sum of squares (RSS) between data and model output. However, we may need to consider many component elements of an ecological process when simulating a model rather than just one, and the RSS may not be an appropriate metric for simulation assessment.

For this dissertation, a multi-objective Evolutionary Algorithm (EA) was used to fit a complex ecological process model. As an example, a model of successive hourly shoot growth of a forest tree was used. First, single-objective methods were tried with the RSS; however, the simulation results did not capture the measured data well, especially contraction periods. The current simulated growth is affected by that of the previous hours because the model includes a regression term. Thus, the fitting result could be improved if there was information about the relation between each data.

The multi-objective optimization method allows us to consider contraction and extension periods separately, and these are an important phenomenon in shoot growth. Since we can set more than one objective function, each focused on particular data features. Also, if there is difficulty in achieving some criteria at the same time, analysis of differential effectiveness in capturing contraction and extension, if it occurs, could help to find what and where the

deficiency of the model is. These effects of considering more than one objective function motivated using a multi-objective optimization method.

Since the model is complex, many objective functions were required. I implemented elitism, a process to keep the best individuals to the next generations, but this needed to be different from that used in other EAs and obtained the following results:

- crossover/mutation rate should be determined dynamically for an efficient search.
- from analysis of the results, deficiencies of the model were identified;
- with the revised model, accuracy of achievement at contraction periods was improved;
- the model reduced bias, but the error did not become small; more biological information about contraction and expansion is needed.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number | Page

LIST OF TABLES

ACKNOWLEDGMENTS

First of all, I wish to express sincere appreciation to my Ph.D. thesis advisor, Professor E. David Ford, for his constant support and unending patience. His scientific guidance has always inspired my motivation. I would also like to thank Dr. Joel H. Reynolds. His advice has always encouraged me to go one step further. My appreciation also goes to Professor Mark Kot for his feedback on both research and English expression in my dissertation. I am very grateful to my committee members, Professor E. David Ford, Professor Mark Kot, Professor Paul D. Sampson, Dr. Joel H. Reynolds, and Professor Donald E. Marshall, for their support. Finally, I would like to acknowledge the people in the Department of Applied Mathematics, my fellow students in the Ford lab, professors in my undergraduate and graduate programs as well as all of my friends and family.

Chapter 1

INTRODUCTION

Fitting an ecological process model to a set of data is frequently done by minimizing the residual sum of squares (RSS) between data and model output. However, we may need to consider many component elements of an ecological process when simulating a model rather than just one, and the RSS may not be an appropriate metric for simulation assessment. In this work, I used a multi-objective *Evolutionary Algorithm* (EA) to fit an ecological model to data. Generally, EAs are a type of algorithm used to solve optimization problems. The algorithms imitate evolution in the natural world; to find solutions, EAs evolve the optimal value of a predetermined number of parameterizations, iterating processes called selection and reproduction (crossover and mutation) (Schoenauer et al. [41]).

As an example, I used a model for the successive hourly shoot growth of a forest tree. To fit the model to data, first, I tried the single-objective optimization method with the RSS, but the resulting parameterizations very much depended on the initial ones; some sets of initial parameter values resulted in reasonable solutions, but others did not converge. Also, the simulated data did not fit the measured data very well for some periods. I decided to use a multi-objective optimization method, instead of a single-objective optimization method because solutions by this search method can consider more than one criterion at the same time, which might help simulated data fit the measured data better.

I used a type of multi-objective EA program, *Pareto_Evolve*, for the shoot growth model. Pareto_Evolve is a *Genetic Algorithm*, a type of EA. I analyzed the results of *Pareto_Evolve*, and introduced several techniques to improve the search software. In my example, I used more assessment criteria than is usual for general multi-objective optimization problems. Deb et al. [10] used two objective functions as a test problem. Zitzler et al. [45] compared the results of the 0/1 Knapsack Problem from several different multi-objective EAs, and the maximum number of knapsacks are four. In general, we need to consider many factors for complex ecological models. These factors are usually used as objective functions to assess how good simulation results are. The higher the dimension of the objective function space, the more variety of solutions obtained. Because of the larger number of objective functions, I decided it would be more effective if the rate of preserving good individuals (*elites*) and of reproducing individuals from individuals in two different ways (*crossover* and *mutation*) were determined dynamically, which are usually fixed.

The task is to achieve as many criteria as possible with reasonable parameterizations; i.e., the search should not only end in a good fit, but resulting parameter values should also be reasonable biologically. The first problem I encountered in using Pareto_Evolve was degeneration of the solution set as the search proceeded. In order to resolve this problem, I introduced the process called *elitism* into the original Pareto_Evolve. Elitism allows the "best" individuals to be carried over from iteration to iteration of the search. The best individuals perform better in a given environment, and this is measured by *fitness* (Section 2.2.1). However, the elitism that I found to be suitable to my problem is considerably different from the elitism generally used. I choose elites depending directly on their assessment vector calculated from the objective function values. I explain this in Chapter 5.

From the results of searches with elitism, I found drawbacks to the initial approach I had used such as no guarantee of better assessment vectors than those of the previous generation and keeping too many elites. Refining the method revealed that *crossover* and *mutation*, which are the main procedures of EAs for evolving the population of parameterization, need to be assigned by different rates from ones in classical EAs for the efficient search; the rate is determined dynamically, instead of being a constant.

This thesis discusses the changes in methods I introduced and how I analyzed results to achieve these improvements. Chapter 2 introduces EAs and how they work. Then the explanation of the software Pareto_Evolve, which I used to solve the shoot growth simulation problem, follows. Chapter 3 explains the data and shows the results of single-objective methods, which motivated use of the multi-objective EAs. Chapter 4 defines the problem that I worked on. Chapter 5 explains elitism and how it works. Chapters 5-8 show refinements to the software Pareto_Evolve for effective searches and the results of each modification. From those results, some problems were found, and what they were is explained in Chapter 9.

I modified Pareto_Evolve as well as changed the search criteria. However, I could not obtain simulation results capturing the measured data well with many achieved criteria. From the results, I concluded that the model was deficient. The last chapter (Chapter 10) shows the results of a modified model and introduces what should be done for the more appropriate model.

Chapter 2

EVOLUTIONARY ALGORITHMS AND PARETO_EVOLVE

In this Chapter, I describe *Evolutionary Algorithms* (EAs) and how they work. The EA software Pareto_Evolve, which was developed to assess and fit complex ecological models, is introduced.

2.1 Different Types of Evolutionary Algorithms

Evolutionary Algorithms (EAs) are algorithms to solve optimization problems imitating evolution in the natural world. In the 1960's, three different types of implementation appeared independently: *Genetic Algorithms* (GAs), *Evolution Strategies* (ESs) and *Evolutionary Programming* (EP). *Genetic Programming* (GP) was developed relatively recently (Peña-Reyes et al. [34]). Although there are similarities in structure between all EAs, these four types developed separately.

There are similarities in structure between all EAs. First, the generation number is set $g = 0$, and population P_0 is initialized. The two basic procedures of stochastic optimization, *selection* and *reproduction* (*crossover* and *mutation*) are repeatedly applied to population P_g of potential solutions until a maximum allowed number of generations is reached or the objective values used in the optimization satisfy conditions predetermined by the user. Crossover is used to produce one or more offspring from two or more parents, which were individuals selected by the selection procedure. This is not implemented in EP. There are different types of crossover depending on which encoding and which EAs are used. In GA, ES, and GP, after crossover, mutation takes place. The purpose is to make a small change in individual information. In EP, mutation is the only genetic operator. The terminologies for EAs are explained in next section.

GAs were introduced by Holland [21]. In order to apply this algorithm to a problem, we need to decide how each individual is encoded to a chromosome (genotype) and how their information is represented (phenotype). The encoding and representation very much depend on the problem. First, the initial set (population), consisting of individuals with "genes", is created. The vector of parameter values for each individual is in genotype space and produces model results giving a new vector in phenotype space. The two fundamental procedures for GAs, selection and reproduction, repeatedly take place until the solutions are obtained (Holland [22]); selection is implemented in phenotype space, and reproduction is implemented in genotype space. The genotype was originally expressed as a binary string. The *Multi-Objective 0/1 Knapsack Problem*, was solved by various types of GAs (Zitzler et al. [45]).

In contrast to GAs, each individual used in ESs is expressed by a real-valued vector consisting of parameter values and associated standard deviations (Bäck et al. [1]). Reproduction is by crossover and mutation. For (μ, λ)-ES, λ offspring are generated from μ parents, and the μ population members for the next generation is chosen only among λ offspring. Since it is not guaranteed that offspring are better than parents, the population of the next generation may not be as good as that of the current generation. Thus, this is a "non-elitist" method. On the other hand, for $(\mu + \lambda)$-ES, μ population members for the next generation are chosen among both μ parents and λ offspring generated from the μ parents. Therefore, this case is an "elitist" method. A multi-objective version of ESs is explained in Kursawe [27].

EP was designed to develop artificial intelligence and to evolve finite state machines (L. Fogel [14]). As for ESs, each individual used is expressed by a real-valued vector, but reproduction is only by mutation. Parents are selected by tournament selection; several randomly selected individuals are compared to determine how well they perform in the environment, that is, how good they are as solutions. Then, the best individuals become the population of the next generation. (Bäck et al. [2], D. Fogel [13]). Some transportation problems were solved by EP (D. Fogel [11]).

GP was originally developed as an application of GAs for computer programming. In GAs, genotype space is fixed, that is, each genotype has a fixed length. However, on solving a problem using a computer, a hierarchical program is preferable to a fixed length string because the size and shape of the genotype for a given problem is not known in advance; it is desirable that they can be changed while a search proceeds. In order to achieve this, each individual in the population has a tree-structure like LISP, instead of a set of strings as in GAs. This hierarchical structure allows each individual to change its size and shape easily. (Banzhaf et al. [4], Koza [26]).

The border lines between these algorithms is becoming less clear. The most recent EAs may not fit completely into any of these four original EA types. In next section, the general idea of EAs is discussed, and then Pareto_Evolve, software to solve multi-objective optimization problems, is introduced. Pareto_Evolve can not be classified as any of the original four EAs completely; however, it was developed based on a mix of ideas from GA and ESs, so I will outline GAs as an example of EAs in next section.

2.2 General Flow of Genetic Algorithms

In this section, the terminologies for GAs, such as *fitness*, *selection*, *crossover* and *mutation*, are explained. First, I outline how a GA solves a single-objective optimization problem:

> Find an m-component vector of $X = (x_1, \dots, x_m) \in \mathbb{R}^m$ that minimizes the value of the function $F(X) \in \mathbb{R}$.

Function $F(X)$ is called the *objective function* of this problem. At the beginning of a search, an initial population of N individual X's is randomly created: $\mathrm{P}_0 = \{X_i^0 | i = 1, \dots, N\}$.

Figure 2.1 shows the general flow of GAs. *Fitness* is a measurement of how good X is as a solution to $F(X)$. Therefore the fitness value for each individual is usually determined directly by its objective function value. Since the problem is a minimization problem, the smaller the function value $F(X)$, the higher the fitness value.

Based on the fitness values, some individuals are probabilistically selected as parents for the next generation in the search. This process is called *selection*. The selected parents produce offspring (*reproduction*) by exchanging some parameter values between two parents and changing some parameter values in small amounts for a parent. These reproduction processes are called *crossover* and *mutation*, respectively. In GAs, these two basic procedures, selection and reproduction, are carried out for each *generation*, and the cycle is repeated until at least one individual achieves each of the objective values within its specified tolerance or a maximum allowed number of generations is reached. At generation g, each individual X_i^g $(i = 1, 2, \ldots, N)$ in population P_g does not have to have distinct values from all other individuals for all parameters; it can have the same value for one or more parameters (but not all) as another individuals.

2.2.1 Fitness

Fitness measures how well an individual performs in the environment. For a single-objective minimization problem, if individual X_1^g gives a smaller objective value than that of X_2^g, X_1^g is considered to be better than X_2^g; thus, the fitness value of X_1^g is higher than that of X_2^g. Therefore the smaller the function value, the higher the fitness value. The fitness value for each individual is usually determined directly by its objective function value for single-objective cases.

An individual with a higher fitness value will be considered as "better", and it has a higher chance to participate in reproduction. At each generation g, the fitness values for all population members X_i^g's are calculated preceding selection process, which is explained next.

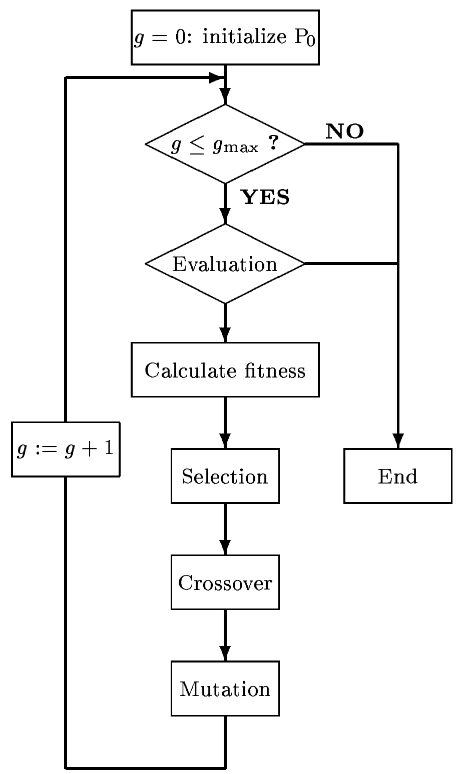


Figure 2.1: General flow of GAs. First, set generation number $g = 0$ and initialize a population P_0. Check whether $g \leq g_{\max}$, where $g_{\max}$ is the maximum allowed number of generations; stop if so; otherwise, go to the next step. Evaluate each individual; stop if the objective values used in the optimization satisfied the predetermined condition by the user; otherwise, go to the next step. Calculate fitness of each individual in the population. Select parents from the population depending on the fitness values. They are to reproduce offspring for the population at the next generation. Apply crossover operator to the parents selected to be crossed over. Then, apply mutation operator to the parents chosen to be mutated. Add one to the generation counter, and repeat the process.

2.2.2 Selection

Selecting individuals for the next generation corresponds, in the natural world, to selecting genotypes for breeding. After calculating the fitness value for each individual X_i^g in population P_g, individuals are selected as parents according to the fitness value to produce offspring. The higher the fitness value, the more chance the individual has to be selected as a parent. An individual can be selected multiple times. The selected individuals are then bred by crossover and mutation to produce new individuals that will make the next generation. If only "fitter genotypes" are chosen at this point, premature restriction to a local optimum could occur; that is, only individuals close to a local optimum could become parents. To avoid this, diversity within the population has to be maintained. Thus, some individuals with lower fitness are also stochastically chosen.

For selection, the process of passing "fitter" individuals into the next generation is often used. Since crossover and mutation are stochastic processes, the objective function value of a created offspring is not always better than that of its parent. Therefore, even if all individuals with higher fitness values are selected as parents to participate in reproduction, it is not guaranteed that objective values of the created offspring are as good, or better than, those of the parents. This means that the new population may not be as good as the old one. To avoid this, some individuals with relatively fitter genes, *elites*, are kept for the next generation. This procedure is called *elitism*. There are different types of elitism (Rudolph [39], Zitzler et al [45]), and I will introduce how it generally works later.

There are different types of selection methods. For *ranking selection*, parents are selected depending on just the rank of their fitness values. All N individuals in the population are ranked as 1 to N with respect to their fitness values. Depending on the rank, probability is assigned to each individual, and N parents are selected probabilistically (Baker [3]). For *tournament selection*, the individual with the best fitness is selected as a parent from a predetermined number of randomly selected individuals; this tournament process is repeated until the selection pool is filled (Goldberg et al. [18], Horn et al. [23]).

Roulette wheel selection (Srinivas et al. [42]), used in Pareto_Evolve, randomly chooses individuals for reproduction in proportion to their relative fitness (Fogel [13]); i.e., the probability that individual X_i is chosen as a parent is:

$$p_i = p(X_i) = \frac{f(X_i)}{\sum_{k=1}^{N} f(X_k)}, \tag{2.1}$$

where f is the fitness function, and N is the size of the population. After the probability is calculated for all X_i $(i = 1, 2, \ldots, N)$, range $[0, 1]$ is divided into N ranges: $[0, p_1]$, $(p_1, p_1 + p_2]$, …, $(\sum_{k=1}^{N-1} p_k, 1]$ (note: $\sum_{k=1}^{N} p_k = 1$). Then the "roulette" is constructed using this portion. Each range corresponds to an individual; X_i corresponds to range $(\sum_{k=1}^{i-1} p_k, \sum_{k=1}^{i} p_k]$. A number picked randomly on $[0, 1]$ indicates a position on the roulette, and the corresponding individual becomes one of the parents to be bred; e.g. if the picked random number is on $(p_1+p_2+p_3, p_1+p_2+p_3+p_4]$, then individual X_4 is selected as a parent.

Actually, the roulette wheel selection method may cause a rapid loss of diversity if there is a markedly unequal fitness among individuals. For example, if the probability of selection for individual X_1 is 0.9, then probability that any of $X_2, \ldots, X_N$ are chosen becomes very low. This is possibly prematurely restricting diversity of the population.

After parents are selected, offspring are created from those parents by crossover and mutation. How many parents are crossed over or mutated also needs to be considered.

2.2.3 Crossover

Crossover is used to produce one or more offspring from two or more parents. In some GAs, not all parents are crossed over; for instance, if crossover probability is 0.25, each parent has a 25% probability of crossing over, that is, only an average 25% of parents are crossed over (Michalewicz [28]). Crossover probability is typically fixed for the entire search and commonly proposed setting is over 0.6 to 0.95 (Fogel [12]).

Different types of crossover are used depending on the Evolutionary Algorithm (EA) being used (Bäck et al. [2]). I introduce an example of one-point crossover to generate two offspring Z_1 and Z_2 from two parents Y_1 and Y_2. A position between two randomly determined consecutive parameters is used as the crossover point. In this example, it is between the k-th and $(k+1)$-th parameters. Offspring Z_1 is generated by concatenating the left substring of parent Y_1 and the right substring of parent Y_2 at the crossover point and offspring Z_2 is created by concatenating of the left substring of parent Y_2 and the right substring of parent Y_1 at the crossover point.

Before crossing over:

Parent Y_1: $x_1^1, x_2^1, \ldots, x_k^1, x_{k+1}^1, \ldots, x_m^1$

Parent Y_2: $x_1^2, x_2^2, \ldots, x_k^2, x_{k+1}^2, \ldots, x_m^2$

After crossing over:

Offspring Z_1: $x_1^1, x_2^1, \ldots, x_k^1, x_{k+1}^2, \ldots, x_m^2$

Offspring Z_2: $x_1^2, x_2^2, \ldots, x_k^2, x_{k+1}^1, \ldots, x_m^1.$

Multi-point crossover determines more than one crossover point. The number of the points is fixed or random. An example of two-point crossover generating two offspring from two parents follows:

Before crossing over:

Parent Y_1: $x_1^1, x_2^1, \ldots, x_{k_1}^1, x_{k_1+1}^1, \ldots, x_{k_2}^1, x_{k_2+1}^1, \ldots, x_m^1$

Parent Y_2: $x_1^2, x_2^2, \ldots, x_{k_1}^2, x_{k_1+1}^2, \ldots, x_{k_2}^2, x_{k_2+1}^2, \ldots, x_m^2$

After crossing over:

Offspring Z_1: $x_1^1, x_2^1, \ldots, x_{k_1}^1, x_{k_1+1}^2, \ldots, x_{k_2}^2, x_{k_2+1}^1, \ldots, x_m^1$

Offspring Z_2: $x_1^2, x_2^2, \ldots, x_{k_1}^2, x_{k_1+1}^1, \ldots, x_{k_2}^1, x_{k_2+1}^2, \ldots, x_m^2.$

2.2.4 Mutation

The purpose of mutation is to make a small change in parameter values of an individual. This is thought valuable for locally searching the objective space. It is done by changing the values of some parameters; for example, adding or subtracting small values or selecting a parameter randomly, picking up a random number on its defined range, and replacing the parameter value with the randomly picked value.

Before mutating:

Parent Y: $x_1, x_2, \ldots, x_m$

After mutating:

Offspring Z: $x_1 + \varepsilon_1, x_2, \ldots x_{k-1}, x_k + \varepsilon_k, x_{k+1}, \ldots, x_m,$

where $\varepsilon_i > 0$ and $x_i + \varepsilon_i$ is in the i-th parameter search range. In this example, only two parameters are randomly chosen for mutation, the first and the k-th parameters.

Deciding which individuals undergo mutation is part of the problem design. For example, only offspring produced by crossover may have a chance to be mutated (Beasley et al. [5]) or, after crossover operators are applied, all individuals have a chance to be mutated (Michalewicz [28]). Or, as in Pareto_Evolve, only individuals not undergoing crossover are mutated.

As for crossover probability, mutation probability is typically fixed for the entire search and is very small compared to crossover probability. Common settings are from 0.001 to 0.01 of the population size (Fogel [12]).

2.3 Multi-Objective Optimization and the Pareto Frontier

Multi-objective problems consider more than one objective function for each individual:

> Find an m-component vector $X = (x_1, \ldots, x_m) \in \mathbb{R}^m$ that minimizes the values of functions $F_1(X), \ldots, F_n(X) \in \mathbb{R}$.

Usually, multi-objective optimization problems cannot be solved because we cannot find a solution X that minimizes all of the functions $F_1, \ldots, F_n$ at the same time. I will explain what kind of solutions we need to look for. Before that, I introduce some terminology.

Definition: X *dominates* X', or X' is *dominated* by X *if and only if*

$$F_i(X) \leq F_i(X') \;\; \forall i, \;\; 1 \leq i \leq n \qquad \text{and} \tag{2.2}$$

$$\exists i, \; 1 \leq i \leq n, \;\; \text{such that} \;\; F_i(X) \neq F_i(X'). \tag{2.3}$$

Also, X is *codominant* to X' *if and only if*

$$\exists i, \; j, \; i \neq j \quad \text{such that} \quad F_i(X) < F_i(X') \quad \text{and} \quad F_j(X') < F_j(X). \tag{2.4}$$

The *Pareto set* is the set of all non-dominated solutions, that is, solutions which are mutually codominant and are not dominated by any other X on the search range. For a multi-objective optimization problem, the goal is to describe the Pareto optimal set, i.e., the *Pareto frontier*, also known as the tradeoff surface or *efficiency frontier* (Keeney et al. [24]).

For example, we consider the case with $n = 2$. We wish to minimize:

$$\begin{cases} F_1(x) = x^2 \\ F_2(x) = (x-2)^2. \end{cases} \tag{2.5}$$

As we can see in Figure 2.2, there is no solution that minimizes both functions at the same time. For $x < x' \leq 0$, $F_1(x') < F_1(x)$ and $F_2(x') < F_2(x)$; thus x' dominates x. On the other hand, for $2 \leq x < x'$, $F_1(x) < F_1(x')$ and $F_2(x) < F_2(x')$; thus, x dominates x'. However, for $0 \leq x \leq 2$, while F_1 gets larger if x increases, F_2 becomes smaller. Therefore, for this problem, $\{x \mid 0 \leq x \leq 2\}$ is the Pareto set, and $\{(F_1(x), F_2(x)) \mid 0 \leq x \leq 2\}$ is the Pareto frontier.

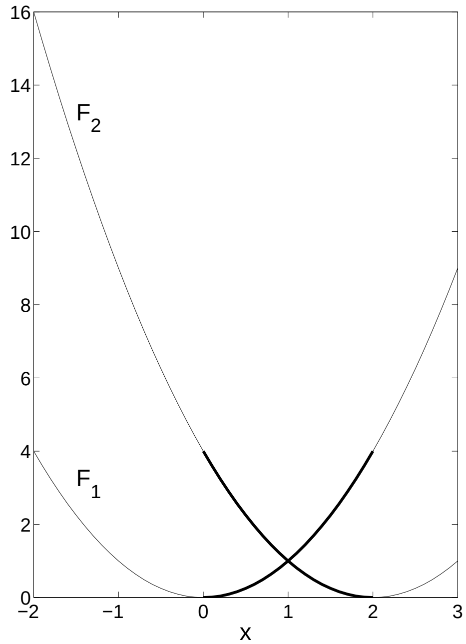


Figure 2.2: Functions $F_1(x) = x^2$ and $F_2(x) = (x-2)^2$.

2.4 Brief Explanation for Original Pareto_Evolve Software

Pareto_Evolve was developed to assess and fit complex ecological models. It solves multi-objective optimization problems using a type of Evolutionary Algorithm (EA), the Non-Dominated Sorting Genetic Algorithm (NSGA) (Srinivas et al. [42]). The software is designed to find the Pareto frontier and its corresponding Pareto set (the set of parameterizations). It was developed to assess and fit complex ecological models (Reynolds [37]).

In this section, I explain how the original Pareto_Evolve program works. The flowchart of the whole process is shown in Figure 2.3. The flow of Pareto_Evolve is similar to the case for the single-objective optimization case introduced in Section 2.2. First, population $\mathrm{P}_0 = \{X_i^0 | i = 1, \ldots, N\}$ (N is a population size) is initialized. In each generation g, fitness is calculated based on performance, the fitness value is assigned to each individual in population P_g, parents are then selected depending on their fitness values, and genetic operators (crossover or mutation) are applied to the parents to produce offspring. These

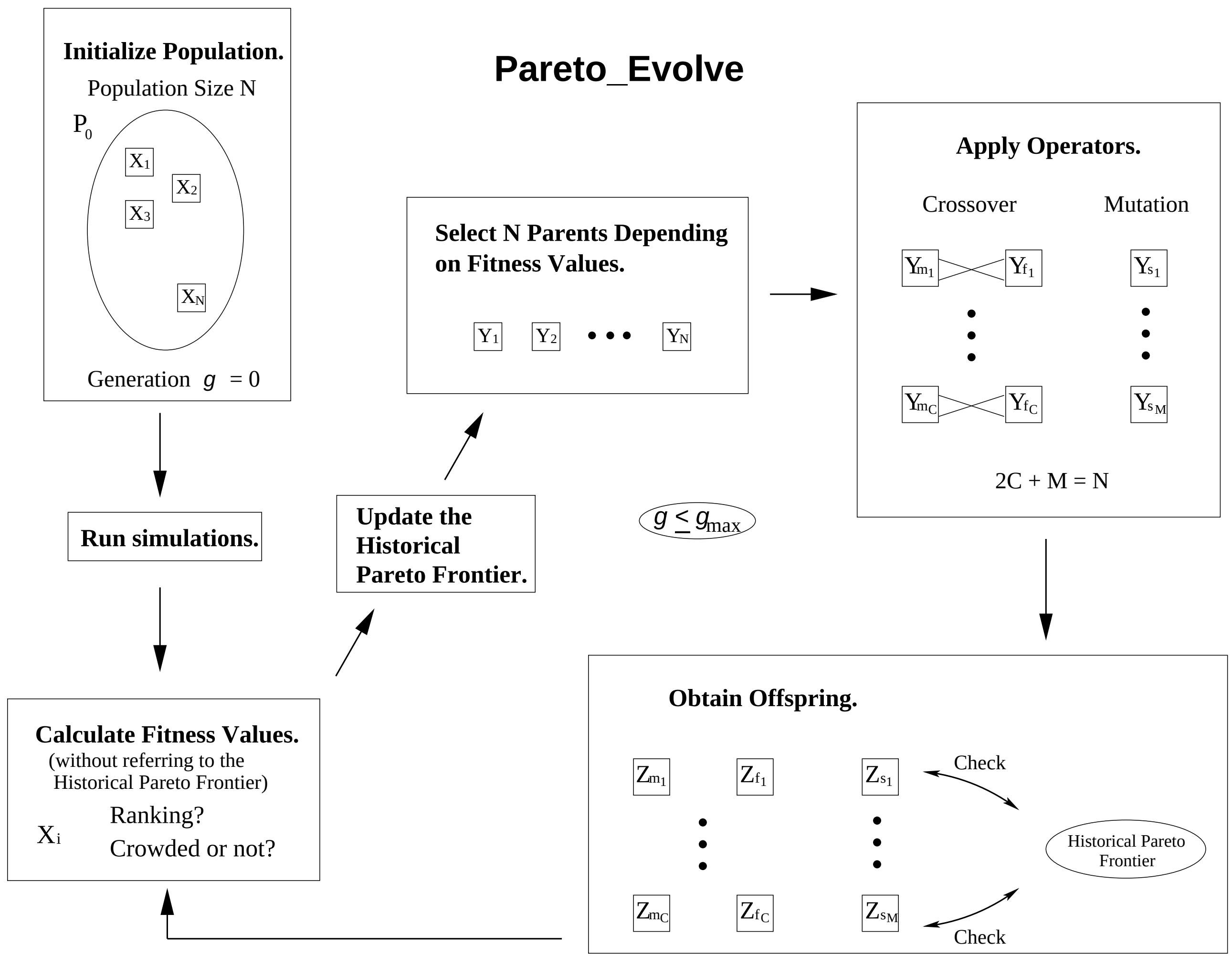


Figure 2.3: Flowchart of Pareto_Evolve.

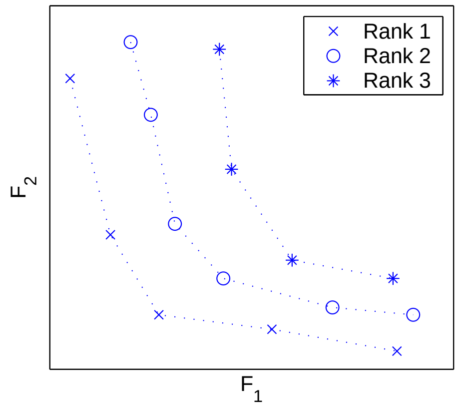


Figure 2.4: Example of Pareto-ranked solutions in the two-dimensional objective function space.

processes are repeated while $g \leq g_{\max}$ ($g_{\max}$ is the maximum allowed number of generation) and none of the population members satisfies the predetermined condition by the user.

While an objective function value is used for a fitness value for single-objective problems, for multi-objective case the fitness value of each individual is a function of both its *Pareto ranking* (or *non-dominated ranking*) and *niche count*. For individual X_{i_1} and X_{i_2} in the current population, if X_{i_1} dominates X_{i_2}, then it is natural that X_{i_1} should have a higher probability of participating in the reproduction process than X_{i_2}. Thus, all population members are ranked by domination. First, all non-dominated individuals are ranked as 1. Among the remaining non-ranked population members, i.e., the individuals excluding ones ranked as 1, non-dominated individuals are assigned rank 2. This procedure is repeated until all members are ranked. This ranking is called *Pareto ranking* or *non-dominated ranking* (Srinivas et al. [42]). Figure 2.4 shows an example of Pareto-ranked solutions in the two-dimensional objective function space.

Niche count, which defines crowding around an individual in the objective space, is also used to calculate the fitness value. In order to avoid convergence to local optimal solutions, it is desirable to have diversity within the population. Thus, to describe the full Pareto frontier, individuals whose objective vectors are close to each other's should have reduced fitness in order to force the search to spread out. Crowding around an individual in the objective space is calculated as its *niche count*, and the niche count will be used to decrease fitness if an individual is close to other individuals. To define niche count for each individual, a *sharing function* is first calculated (Goldberg et al.[19]):

$$Sh(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{\text{share}}}\right)^2 & \text{if } d_{ij} \leq \sigma_{\text{share}} \\ 0 & \text{otherwise,} \end{cases} \tag{2.6}$$

where d_{ij} denotes the Euclidean distance between objective vectors of two distinct individuals X_i and X_j, that is,

$$d_{ij} = \sqrt{\sum_{k=1}^{n} (F_k(X_i) - F_k(X_j))^2} \tag{2.7}$$

with

$$F = (F_1, F_2, \cdots, F_n),$$

and σ_{share} is the maximum allowed sharing distance (i.e., neighborhood radius), which is a predetermined user-defined number. Then, niche count, n_i for each individual X_i is determined as

$$n_i = \sum_{\substack{j \neq i \\ 1 \leq j \leq N}} Sh(d_{ij}). \tag{2.8}$$

In the original NSGA (Srinivas et al. [42]), distances d_{ij} are calculated between two individuals of the same rank only in terms of their distance in parameter space (genotype). On the other hand, in initial Pareto_Evolve software, d_{ij}'s are calculated for all pairs (X_i, X_j) of individuals and the distance is between objective vectors, instead of between parameterizations. It is useful to force objective function values to spread out along the frontier because this produces a variety of Pareto groups. Also, in general, the problems solved by the original Pareto_Evolve did not always have parameters that were commensurable with each other; they could not be compared directly since some parameters had different units from the others. On the other hand, objective vectors share a common scale, 0 or 1 for the binary case and $[0, 1]$ for the continuous case. Those two factors are the reasons why the distance is calculated between objective vectors, instead of parameterizations for all pairs of individuals.

After the maximum niche count $n_{\max}^{(r)}$ for individuals ranked in r is found, *baseline fitness* (dummy fitness in the NSGA) b_r for individuals ranked r is calculated as follows:

$$\begin{cases} b_1 &= 100 \\ b_r &= b_{r-1} \cdot W/n_{\max}^{(r-1)} \quad \text{for } r = 2, 3, \cdots, \rho, \end{cases} \tag{2.9}$$

where $0 < W < 1$ (WGT in the code). Then, fitness of each individual X_i in rank r is defined as

$$b_r/n_i. \tag{2.10}$$

Parameter W decreases the previous baseline. Fitness for individuals of rank r ($r = 2, 3, \cdots, \rho$) has to be lower than that for individuals of rank $r - 1$. To satisfy this, parameter W should be determined as $0 < W < 1$; if $W = 1$, then there could exist an individual of rank r whose fitness value is equal to that with the smallest fitness of rank $r-1$, because $b_r = b_{r-1}/n_{\max}^{(r-1)}$ is the smallest fitness value of rank $r-1$. Although $W = 0.5$ in the code, the user can change the value.

To consider the definition of fitness value in more detail, we consider an individual X_i and assume that there are many individuals X_j whose objective vectors are very close to that of X_i. Then, by equation (2.6), the values of $Sh(d_{ij})$ are close to 1 because $d_{ij} \ll \sigma_{\text{share}}$. If this holds, then the niche count takes a large value by equation (2.8). Therefore, by equation (2.10), we see that the fitness value for X_i is small. This shows that fitness is low if space is crowded around the objective function vector of X_i.

By the result of the fitness calculation, N individuals are randomly chosen from population P_g as parents for crossover and mutation. Each individual can be selected more than once. For this selection, roulette wheel selection method, which was introduced in Section 2.2.2, is used.

In Pareto_Evolve, each parent is assigned either a crossover operator or a mutation operator. Also, for Pareto_Evolve, it is preferred that crossover/mutation probability changes as a function of the generation number. Generally, population members at later generations are expected to approach reasonable solutions. Thus, mutation, which makes small changes in some parameter values of an individual for a local search, has been considered more important than crossover at later generations. In the Pareto_Evolve, the probability of receiving crossover, which is set at 2/3 at generation 0, decreases linearly as the generation number increases to reach to 0 at generation $g_{\max}$. On the other hand, the probability of mutation increases linearly from 1/3 to 1 with generation number.

Crossover occurs done by a method called uniform crossover (Syswerda [43]); the number of the crossover points, as well as crossover points themselves, is chosen randomly. Nonuniform mutation, introduced in Michalewicz [28], is used: each parameter x_k is determined randomly whether the value is changed or not; if yes, x_k is changed to x'_k, which is defined as

$$x'_k = \begin{cases} x_k + \Delta(g, UB - x_k) & \text{if addition} \\ x_k - \Delta(g, x_k - LB) & \text{if subtraction} \end{cases} \tag{2.11}$$

with the lower bound LB and upper bound UB of the domain, and function Δ. The probability of the step size defined by function Δ gets closer to 0 as generation g becomes larger so that the larger the generation becomes, the more locally the space is searched by mutation. Michalewicz [28] defined Δ as:

$$\Delta(g, y) = y \cdot \left(1 - q^{(1-g/g_{\max})^{\beta}}\right), \tag{2.12}$$

where q is randomly chosen from $[0, 1]$ whenever Δ is called, $g_{\max}$ and g are the maximal and current generation numbers, respectively, and β is a system parameter determining the degree of dependency on generation number g. The larger β, the smaller Δ, so for a large value of β, the change of the parameter value by mutation is small. The value of β is set at 5, which was used by Michalewicz [28]; however, it can be changed by the user. Whether addition or subtraction is applied is determined by a pseudo-coin-toss.

After reproduction, the generation number is checked. If it has reached the maximum, the search is over. Otherwise, the algorithm goes back to the procedure to calculate fitness.

The original Pareto_Evolve code has an external memory, *Historical Pareto Frontier*. This is a set of non-dominated individuals found from the initial population up to the previous population, and all individuals are codominant with the others in the set. As shown in Figure 2.3, after calculating the fitness, the set is updated; if there is a individual in the Historical Pareto Frontier dominated by a non-dominated individual of the current generation, it is replaced by the non-dominated individual in the current population. This process was designed to maintain a complete record of the full Pareto frontier as discovered during the search, rather than to simply rely on the Pareto frontier retained in the final generation search results. Also, if there are offspring matching any of the individuals in the Historical Pareto Frontier, they are mutated until they differ from all of the individuals in the Historical Pareto Frontier to reduce redundant searches.

Since I considered the simulation result captured the measured data better if the more criteria were achieved, I defined elitism to keep individuals achieving many criteria. What I emphasized was a program to retain and utilize the best individuals throughout the search rather than to simply keep a record of external to the search. I refined Pareto_Evolve so that an individual better than those of the previous generation can survive as an elite (Chapter 5), which allows us to keep some individuals without any modification. I added this new process and removed the process of the Historical Pareto Frontier from the original code.

Chapter 3

INTRODUCTION TO THE MODEL

My goal is to fit an ecological model to shoot growth data using a multi-objective Evolutionary Algorithm (EA), Pareto_Evolve. This chapter first introduces the data collection and then explains the model construction. Then, I show the results of fitting using three different types of single-objective methods; this motivates the use of the multi-objective method.

I tried the simplex methods and simulated annealing method to minimize the residual sum of squares (RSS), but the results depended on the initial parameterizations. Also, the simulated data did not fit the measured data very well for some periods. These two issues motivated the use of a different method. I chose multi-objective EAs because they can consider more than one criterion, potentially improving the fit of the simulated data.

This problem was chosen because while the data has very high measurement accuracy, it shows some complex variation. The question I ask is whether the proposed model can explain the important features of the data. I may consider changing the model and/or changing the assessment criteria for features of the data, but, as I will show, the principal requirement is develop a more effective search by Pareto_Evolve and to make the model the data well.

3.1 Collected Data

The response of plants to environmental change can be complex. One reason is that environmental change often involves a number of factors. For example, a change from sunny to cloudy weather is likely to produce decreased radiation, decreased temperature and decreased vapor pressure deficit. A second reason is that biological systems comprise a set of interacting processes that may respond in different ways to change. For example, a change from sunny to cloudy weather on a plant may affect photosynthesis through a change in light, growth and through a change in temperature, and transpiration through a change in vapor pressure deficit. Furthermore, the rates of these processes may interact. A change in transpiration can affect the internal water potential of a plant and affect both the growth process and photosynthesis. The plant-environment system is dynamic and nonlinear.

While direct experimentation can produce useful results, it has important limitations. Experimentation is most effective when a few components of a system are manipulated and this can limit what is discovered about dynamic interactions. Furthermore, for the plant-environment system, experiments are most effective when made with small plants, but important components of dynamics are directly affected by plant size, as in the water relations of trees. For these reasons, constructing models of the direct responses of plants in naturally fluctuating environments is important in the scientific investigation of the plant-environment system. Practical problems of model development and assessment of an environment-to-plant response provide the motivation for the EA system developed in this work.

Investigations were made of the day-to-day increment of shoot extension in a Sitka spruce plantation in southwest Scotland (Ford et al. [16] [17]). Measurements were made from a marked point on non-growing wood to the shoot apex at 9 am BST every day with a ruler. At the same time, daily totals for radiation and mean daily temperature were made. The advantage of this simple measurement system was that many shoots could be measured, and so variation in response within trees could be defined. Ford et al. [16] showed that shoots at the lowest levels in the canopy started growth earliest by just a few days, but shoots in the upper levels in the canopy continued growth for substantially longer. From the end of May, leading shoots grew for some 90 days while branches within the canopy only grew for 40 days. However, as far as the accuracy of the measurements could show, the patterns of day-to-day fluctuation in growth were similar between all shoots although the amplitude of change was greater for those shoots that grew most overall, i.e., the leading shoots and shoots of the upper branches.

An exploratory investigation of daily weather changes on shoot extension rates was made using time series analysis. Serial auto and cross-correlation analysis showed variation in detrended shoot increments was related to previous changes in solar radiation and temperature. For the first half of the growing season, autoregressive models were fitted (Box et al. [6]). These showed delays in the influence of temperature and solar radiation. A typical model, using parsimony as a guiding principle in model identification, and where T_t is daily mean temperature in centigrade, R_t radiation in Mega Jules, and S_t is detrended shoot extension at time t, is

$$S_t = 0.133 \cdot T_{t-1} - 0.042 \cdot T_{t-2} + 0.0107 \cdot R_{t-2} + 0.0150 \cdot R_{t-3}. \tag{3.1}$$

This model raised many questions. How could the effect of temperature at $t-2$ be negative? Was the lag of solar radiation really over a 2 and 3 day period? These are important questions for tree physiologists because answers to them tell things about the dynamic process of tree response to the environment. However, a difficulty with this work was that the time interval and accuracy of measurement was felt to limit what could be found out. Scientists measuring the shoots felt that considerable contraction of shoots was taking place on sunny days, and this could be a response to transpiration and did not affect "real" growth. Indeed the lag in positive response to solar radiation might be due to a recovery time for this contraction.

To obtain better data, a shoot extension sensor was constructed (Milne et al. [31]). This system sensed the top of the growing shoot using a light emitting diode and gave measurement accurate to better than 1 mm. The sensor was positioned on the shoot continuously and was scanned every 20 minutes. The results used are hourly sums based on those 20 minutes scans (Figure 3.1). Concurrent with the measurements of shoot extension, measurements of temperature, radiation and vapor pressure deficit were also made (Figure 3.2 (a) and (c)). From research into canopy transpiration (Milne et al. [30]), canopy conductance was

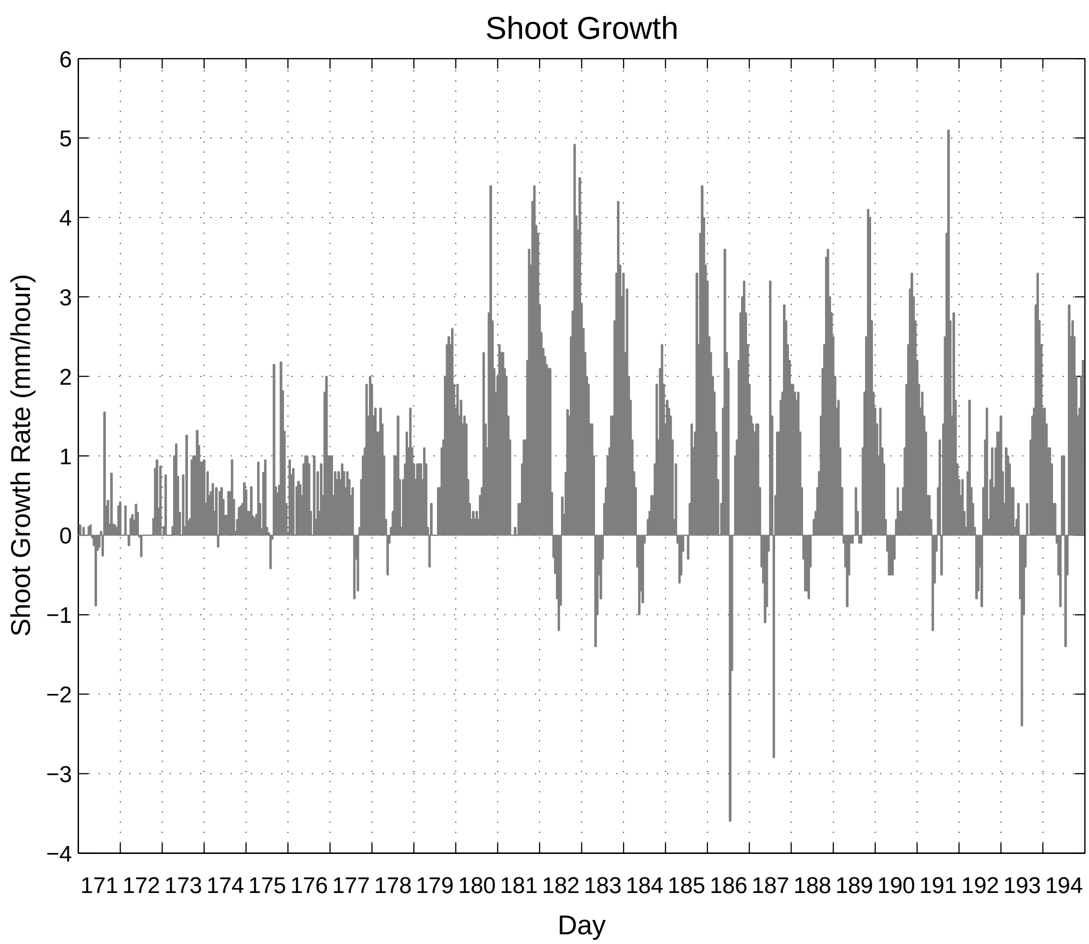


Figure 3.1: Hourly measured shoot growth rate of the terminal shoot of a 14 year plantation growth tree of Sitka spruce in southwest Scotland from Julian day 171 through 194 in 1976. During the period from day 177 onwards, the shoot has substantial contractions (negative growth) during the middle of the day and most rapid growth during the night.

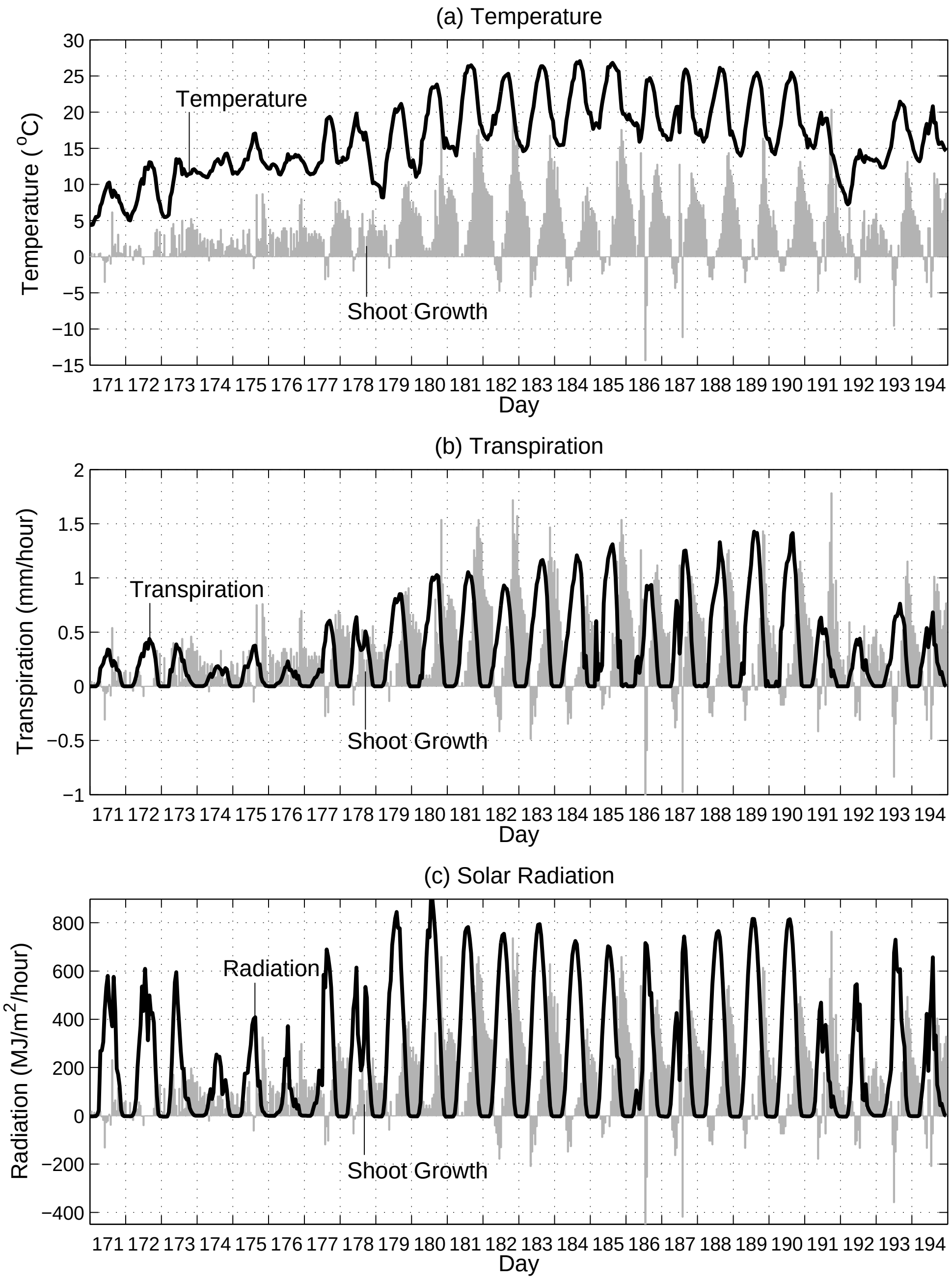


Figure 3.2: Hourly (a) measured temperature, (b) calculated transpiration and (c) solar radiation from Julian day 171 through 194. Hourly measured shoot growth rate are shown for reference (gray bars).

estimated and so hourly estimates of transpiration (Figure 3.2 (b)) could be calculated for the period that shoot extension was measured. Additionally, measurements of soil water potential were made, which in conjunction with moisture release curves were used to calculate soil water loss. The most apparent fluctuation in the hourly estimates of shoot extension is the diurnal cycle of contraction and expansion, which is related to changes in tree water potential (Milne et al. [29] [31]).

In this model, the water lost due to transpiration is balanced by water taken up from the soil. This was studied at this site by Deans [9]. Secondly, there is a lag between water being lost from the needles and shoots and it being replaced from the soil and possibly from the trunk (Milne et al. [29] [31]). Two things need to be considered in calculating the effects of water deficit on shrinking and contraction:

1. an increase in deficit has a more rapid effect on contraction than the equivalent decrease in deficit has an expansion; there is a hysteresis in the drying and rewetting cycle effect on tissue size. Consequently, two parameters x_5 for the effect during contraction and x_6 for the effect during remodel of the deficit were estimated.

2. the shrinkage and expansion of tissue does not affect the complete length of shoot that has grown because secondary shrinking of xylem trochoids stiffens the shoot progressively from the base. Here the water deficit change is applied to the amount of shoot that grew over the last twenty-four hours.

From the perspective of modeling, I have a dynamic system, but one with a number of components. Clearly, a starting point in model construction should be the model constructed for daily changes, and I should add to this a dynamic for the removal of water from the tree during the day due to transpiration and the recharge from the soil at night.

3.2 Model Construction

The model I used to find the best fit to the shoot growth data by Pareto_Evolve was constructed based on the previous section:

$$\begin{aligned} S_t &= x_1 \cdot \left(\sum_{k=1}^{24} T_{t-k} \right) \Big/ 24 \; - \; x_2 \cdot \left(\sum_{k=25}^{48} T_{t-k} \right) \Big/ 24 \\ &+ \; x_3 \cdot \sum_{k=25}^{48} R_{t-k} \; + \; x_4 \cdot \sum_{k=49}^{72} R_{t-k} \\ &- \; \begin{cases} x_5 \cdot \Delta_t D \cdot \sum_{k=1}^{24} S_k^* & (*) \\ x_6 \cdot \Delta_t D \cdot \sum_{k=1}^{24} S_k^* & (**) \end{cases} \end{aligned}$$

with

$$\Delta_t D \equiv D_t - D_{t-1} = W_t - U_t = W_t - x_7 \cdot D_{t-1},$$

where variables are defined in Table 3.1 and parameters are explained in Table 3.2. This shows the shoot growth at time t. The coefficients of the first and second terms are the average temperature for the previous 24 hours and two 24-hour periods before, respectively. The coefficients of the third and fourth terms are the total solar radiation for two and three 24-hour periods before, respectively. The last sums are the total growth for the previous day, and the previously calculated simulated rates are used. For $t < 24$, I used the measured data instead since the simulated ones are not available. Term (*) is selected if current hourly water deficit is less than one for the previous hour, i.e., $\Delta_t D < 0$, and (**) otherwise.

3.3 Single-Objective Optimization Methods

Before I introduce the results of Pareto_Evolve using the model constructed in the previous section, I show the results from fitting using the single-objective optimization methods. I

Table 3.1: Variables used in the shoot growth model for forest trees.

t	hour(s) from when simulation started; for example, if current time is hour h on day d, and simulation started from hour h_0 on day d_0, then $t = (d - d_0) * 24 + (h - h_0)$;
S_t	hourly simulated growth rate (mm/hour) at time t;
T_t	temperature (^oC) at time t;
R_t	hourly solar radiation (MJ/m^2/hour) at time t;
D_t	hourly water deficit (mm/hour) at time t,
S_t^*	hourly simulated growth rate (mm/hour) at time t on the previous day,
W_t	hourly water transpiration (mm/hour) at time t;
U_t	hourly water uptake (mm/hour) from the soil at time t.

Table 3.2: Definitions of the parameters $x_1, \ldots, x_7$ used in the model and their approximate feasible ranges.

x_1	rate of extension per average temperature for the past 24 hours; the value is about four times as x_2 (Ford et al. [17]); unit: mm/^oC; search range: $[0, 0.4]$;
x_2	rate of extension per average temperature between the past 48 hours through 24 hours; unit: mm/^oC; search range: $[0, 0.1]$;
x_3	rate of extension per total radiation between the past 48 hours through 24 hours; unit: $10^6(\text{mm})^3/\text{MJ}/360$; search range: $[0, 0.1]$;
x_4	rate of extension per total radiation between the past 60 hours through 48 hours; unit: $10^6(\text{mm})^3/\text{MJ}/360$; search range: $[0, 0.1]$;
x_5	rate of extension per increment of product of previous hourly water deficit and previous daily growth increment when the current hourly deficit is less than one for the previous hour; $x_5 \leq 1$ since x_5 is to downsize $\Delta_t D \cdot \sum_{k=1}^{24} S_k^*$ to calculate S_t; unit: hour/mm; search range: $[0, 1]$;
x_6	same rate as x_5, but when current hourly deficit is greater than one for the previous hour; unit: hour/mm; search range: $[0, 1]$;
x_7	rate of water deficit for the previous hour; since $U_t = x_7 \cdot W_{t-1}$, and $0 \leq U_t \leq W_{t-1}$; no unit; search range: $[0, 1]$.

solved the problem by three different methods: the Nelder-Mead simplex method, the Powell's version of the simplex method and the simulated annealing method.

Using the three methods, I minimized the residual sum of squares (RSS) between the measured and simulated data. I chose the RSS as an objective function because it is commonly used and I considered it is the simplest single-objective function. For the simulated data, I used the model introduced in the previous section and the period of seven days, Julian day 178 through 184. The shorter the period to be simulated, the easier the search. Thus, I worked on the seven day period, instead of the entire twenty-four day period. I decided to use this period since it contains both contraction and extension parts. The results show that it is not appropriate to use single-objective methods for this particular type of problem.

3.3.1 Nelder-Mead Simplex Method

The Nelder-Mead simplex method (Nelder et al. [33]) solves optimization problems without constraints. It does not use any derivatives, but it requires continuity of the objective function. Strictly speaking, I should not have used this method for the problem; the model is not defined by a continuous function, so the objective function RSS is not continuous either. However, since the method requires only continuity of the objective function, and the RSS is piecewise continuous, I tried it to see how well the model fit the data.

I used the program for the Nelder-Mead simplex method by Press et al. [36]. For an m-dimensional minimization problem, the method generates $m + 1$ points that give smaller objective values than those at current $m + 1$ points. To start a search, $m + 1$ initial points are required. For this case, we need eight different sets of initial points for search since there are seven distinct parameters.

I set the eight initial points to

$$Q_1^0 = (0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0)$$
$$Q_2^0 = (0.0,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)$$
$$Q_3^0 = (\tilde{x}_1,\ 0.0,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)$$
$$Q_4^0 = (\tilde{x}_1,\ \tilde{x}_2,\ 0.0,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)$$
$$Q_5^0 = (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ 0.0,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)$$
$$Q_6^0 = (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ 0.0,\ \tilde{x}_6,\ \tilde{x}_7)$$
$$Q_7^0 = (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ 0.0,\ \tilde{x}_7)$$
$$Q_8^0 = (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ 0.0),$$

where

$$\tilde{x}_1 = 0.4\alpha, \quad \tilde{x}_2 = 0.1\alpha, \quad \tilde{x}_3 = 0.1\alpha, \quad \tilde{x}_4 = 0.1\alpha, \quad \tilde{x}_5 = 1.0\alpha,$$
$$\tilde{x}_6 = 1.0\alpha, \quad \tilde{x}_7 = 1.0\alpha.$$

I set the value of each $\tilde{x}_i$ $(i = 1, \ldots, 7)$ to the product of α and the upper limit of the search range for Pareto_Evolve, where $\alpha = 10^{-1}, 10^{-2}$ or 10^{-3}.

The resulting eight sets of solutions, $Q_1, Q_2, \ldots, Q_8$, and the RSS for $\alpha = 10^{-1}, 10^{-2}, 10^{-3}$ were as follows:

$\alpha = 10^{-1}$	RSS
$Q_1 = (0.0710,\ 0.0413,\ 0.0410,\ 0.0235,\ 0.1064,\ 0.2709,\ 0.5668)$	76.2596
$Q_2 = (0.0709,\ 0.0412,\ 0.0410,\ 0.0236,\ 0.1065,\ 0.2710,\ 0.5670)$	76.2596
$Q_3 = (0.0709,\ 0.0411,\ 0.0410,\ 0.0235,\ 0.1065,\ 0.2708,\ 0.5669)$	76.2596
$Q_4 = (0.0709,\ 0.0412,\ 0.0409,\ 0.0236,\ 0.1065,\ 0.2710,\ 0.5671)$	76.2596
$Q_5 = (0.0711,\ 0.0414,\ 0.0410,\ 0.0235,\ 0.1065,\ 0.2706,\ 0.5667)$	76.2596
$Q_6 = (0.0709,\ 0.0413,\ 0.0410,\ 0.0236,\ 0.1065,\ 0.2711,\ 0.5672)$	76.2596
$Q_7 = (0.0710,\ 0.0413,\ 0.0410,\ 0.0236,\ 0.1065,\ 0.2710,\ 0.5673)$	76.2596
$Q_8 = (0.0711,\ 0.0414,\ 0.0410,\ 0.0235,\ 0.1066,\ 0.2709,\ 0.5669)$	76.2596

$\alpha = 10^{-2}$	RSS
$Q_1 = (0.0709,\ 0.0413,\ 0.0410,\ 0.0236,\ 0.1063,\ 0.2710,\ 0.5669)$	76.2595

$Q_2 = (0.0710,\ 0.0413,\ 0.0409,\ 0.0237,\ 0.1065,\ 0.2709,\ 0.5669)$	76.2595
$Q_3 = (0.0709,\ 0.0413,\ 0.0410,\ 0.0236,\ 0.1064,\ 0.2711,\ 0.5671)$	76.2595
$Q_4 = (0.0707,\ 0.0411,\ 0.0409,\ 0.0237,\ 0.1063,\ 0.2710,\ 0.5668)$	76.2595
$Q_5 = (0.0709,\ 0.0413,\ 0.0410,\ 0.0237,\ 0.1062,\ 0.2706,\ 0.5663)$	76.2595
$Q_6 = (0.0706,\ 0.0410,\ 0.0410,\ 0.0237,\ 0.1062,\ 0.2709,\ 0.5669)$	76.2596
$Q_7 = (0.0709,\ 0.0413,\ 0.0410,\ 0.0236,\ 0.1064,\ 0.2711,\ 0.5669)$	76.2596
$Q_8 = (0.0707,\ 0.0410,\ 0.0411,\ 0.0236,\ 0.1063,\ 0.2710,\ 0.5669)$	76.2595

$\alpha = 10^{-3}$	RSS
$Q_1 = (0.1444,\ 0.0636,\ 0.0157,\ -0.0112,\ 0.1149,\ 0.2190,\ 0.4930)$	86.8733
$Q_2 = (0.1443,\ 0.0636,\ 0.0157,\ -0.0112,\ 0.1150,\ 0.2191,\ 0.4929)$	86.8734
$Q_3 = (0.1444,\ 0.0637,\ 0.0157,\ -0.0111,\ 0.1148,\ 0.2188,\ 0.4926)$	86.8734
$Q_4 = (0.1444,\ 0.0636,\ 0.0157,\ -0.0111,\ 0.1149,\ 0.2188,\ 0.4927)$	86.8734
$Q_5 = (0.1444,\ 0.0636,\ 0.0157,\ -0.0111,\ 0.1149,\ 0.2190,\ 0.4926)$	86.8734
$Q_6 = (0.1445,\ 0.0637,\ 0.0157,\ -0.0111,\ 0.1148,\ 0.2190,\ 0.4929)$	86.8734
$Q_7 = (0.1444,\ 0.0636,\ 0.0157,\ -0.0111,\ 0.1149,\ 0.2189,\ 0.4927)$	86.8734
$Q_8 = (0.1444,\ 0.0637,\ 0.0157,\ -0.0111,\ 0.1147,\ 0.2187,\ 0.4925)$	86.8734.

The numbers of iterations to get those results are 602, 988, 781 for $\alpha = 10^{-1}, 10^{-2}, 10^{-3}$, respectively.

The errors, that is, the ratios of the RSS to the total sum of squares of hourly measured data for the period of seven days are 13.36% for $\alpha = 10^{-1}, 10^{-2}$ and 15.22% for $\alpha = 10^{-3}$. The parameterizations of the first two results ($\alpha = 10^{-1}, 10^{-2}$) very similar to each other. However, the parameterizations of $Q_1, \ldots, Q_8$ for $\alpha = 10^{-3}$ are quite different from those of the other two results. This happened probably because the search reached the local optimal solution before the space was searched widely enough; since all initial points were too close to each others the searched range was narrower than the other two cases. As is generally known for many optimization methods, the solution depends on the initial points. Thus, if α is small, or the initial points are close to each others, then the RSS may not be minimized

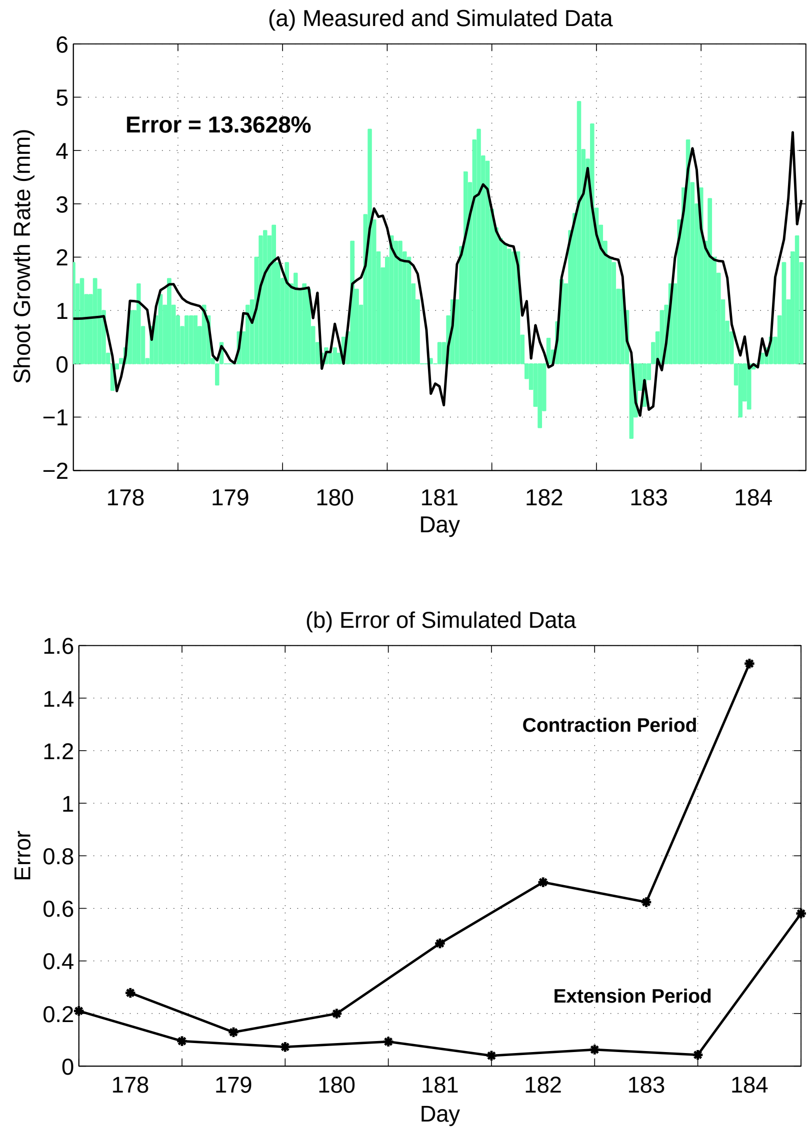


Figure 3.3: Results of the Nelder-Mead simplex method with $\alpha = 10^{-1}$. (a) The simulated data using the RSS for the period of seven days, from Julian day 178 through 184 (solid line), and the measured data (bar). (b) The ratio of the RSS between the measured and simulated data to the total sum of squares of the measured data for the period of seven days, day 178-184. Errors for contraction periods (6:00-18:00) and extension periods (18:00-6:00+) are plotted separately. Extension periods on the first and last day are 0:00-6:00 and 18:00-0:00, respectively. The overall error is quite small (13.3628%), but we can see that the model is biased to the extension periods; the errors for contraction periods are larger than those for extension periods.

efficiently.

The fit of the search for $\alpha = 10^{-1}$ are shown in Figure 3.3(a). Figure 3.3(b) shows that the simulated data does not fit well for the contraction periods, which means the model is biased against the contraction period when I use the simplex method.

3.3.2 Powell's Version of Simplex Method

There is a modified version of the Nelder-Mead simplex method, which solves nonlinear optimization problems with constraints (Powell [35]). I tried this method to see if I would obtained different results.

For this method, I used subroutine "NLPNMS" of software "SAS/IML" to find the solution minimizing the RSS (SAS Institute [40]). NLPNMS requires us to set only one initial point and allows us to set constraints. The constraints for the problem are the boundary conditions, and they are the ones shown in Table 3.2.

I found that the method is quite robust because no matter what the initial point was, the results were the same. The set of parameterizations I obtained as the solution is

$$(0.070962, 0.041340, 0.041006, 0.023611, 0.106384, 0.270735, 0.566824).$$

This is very similar to those obtained using the Nelder-Mead simplex method. The value of the RSS and the ratio of the RSS to the squared sum of hourly measured data for the period of seven days, Julian day 178 through 184, are 76.259439197 and 13.3627%, respectively. These values show that the results of this method are as good as those of the Nelder-Mead simplex method.

For Powell's version of the simplex method, if the step size for parameters is too small, a search requires more iteration and I sometimes could not obtain solutions. In addition to this problem, the method still requires continuity of the objective function, and so is not appropriate for this problem.

3.3.3 Simulated Annealing Method

The last method I tried is the simulated annealing method (Kirkpatrick et al. [25]). When a substance such as metal is liquefied with high temperature, molecules move freely. If the substance cools down slowly from this condition, the atoms form a pure crystal. This condition is stable with minimum energy, that is, the substance is strong. On the other hand, quick cooling does not bring this state but instead produces an amorphous state with higher energy. The simulated annealing method adopts this principle for finding global minima.

We consider an m-dimensional minimization problem. Search starts from high temperature. At fixed temperature, like the simplex method, the generates $m+1$ points which give smaller objective values than those at the current $m+1$ points. After it reaches the thermal equilibrium, temperature is dropped a little, and the same processes are repeated. If we could make temperature decrease infinitesimally and repeatedly keep up this process, the global minimum would certainly be found. However, in reality, both the decrement of temperature and the number of repetition of the process have to be finite. Therefore, we need to set practical cooling schedule.

This method is often used for the traveling salesman problem for finding the shortest way to visit designated cities. The method can solve minimization problems not only for discrete parameter spaces but also continuous ones (Press et al. [36]). As for the simplex method, it does not require any derivatives.

For the method, I again used a program by Press et al. [36]. As I mentioned above, for an m-dimensional minimization problem, $m+1$ initial points are required. Therefore I needed eight different sets of the initial points to solve my problem using the simulated annealing method because there are seven distinct parameters.

I tried initial temperature 10^4 with two different sets of the initial parameterizations:

$$\begin{aligned}
Q_1^0 &= (0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0)\\
Q_2^0 &= (0.0,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)\\
Q_3^0 &= (\tilde{x}_1,\ 0.0,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)\\
Q_4^0 &= (\tilde{x}_1,\ \tilde{x}_2,\ 0.0,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)\\
Q_5^0 &= (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ 0.0,\ \tilde{x}_5,\ \tilde{x}_6,\ \tilde{x}_7)\\
Q_6^0 &= (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ 0.0,\ \tilde{x}_6,\ \tilde{x}_7)\\
Q_7^0 &= (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ 0.0,\ \tilde{x}_7)\\
Q_8^0 &= (\tilde{x}_1,\ \tilde{x}_2,\ \tilde{x}_3,\ \tilde{x}_4,\ \tilde{x}_5,\ \tilde{x}_6,\ 0.0),
\end{aligned}$$

where

$$\begin{aligned}
&\tilde{x}_1 = 0.4\alpha, \quad \tilde{x}_2 = 0.1\alpha, \quad \tilde{x}_3 = 0.1\alpha, \quad \tilde{x}_4 = 0.1\alpha, \quad \tilde{x}_5 = 1.0\alpha,\\
&\tilde{x}_6 = 1.0\alpha, \quad \tilde{x}_7 = 1.0\alpha,
\end{aligned}$$

with $\alpha = 10^{-2}, 10^{-3}$.

The resulting RSS's are 76.259492 and 230.430525 for $\alpha = 10^{-2}, 10^{-3}$, respectively, and the parameterizations are as follows, respectively:

$$(0.070972,\ 0.041459,\ 0.041037,\ 0.023700,\ 0.106354,\ 0.270681,\ 0.566628),$$

$$(0.058095,\ 0.044864,\ 0.041309,\ 0.017170,\ 0.141898,\ 0.014240,\ 0.004390).$$

The resulting parameterization for $\alpha = 10^{-2}$ is again very similar to those by the simplex methods. For two different sets of results with $\alpha = 10^{-2}, 10^{-3}$, we again see that the solutions depended on the initial points. As for the Nelder-Mead simplex method, the search reached a local optimal solution before the space was searched widely enough. This is probably because all optimal points were too close to each others. Also, with $\alpha = 10^{-2}$, I tried initial temperature 10^3, and the RSS was 235.187483, which implies the initial temperature is too low. Thus, this cooling schedule is not appropriate for the problem.

3.3.4 Motivations of Using Multi-Objective Methods

As we saw, the results of the three methods I tried were affected by the initial points, which is commonly known for single-objective optimization methods. Also, the parameterizations of the best result for each of the three methods were similar to each others.

Since the model includes a regression term, the current growth is affected by that of the previous hours. Thus, I considered that the fitting result could be improved if I had the information about the relation between hourly data. The RSS is not informative enough to express this. Also, from the result shown in Figure 3.3(b), if I think about contraction and extension periods separately, I may find a set of parameterizations that makes the simulation result capture the measured data for contraction periods better. The multi-objective optimization method allows us to consider contraction and extension periods separately since we can set more than one objective function, each focused on particular data features. Also, if there is difficulty in achieving some criteria at the same time, analysis could help to find what and where the deficiency of the model is. These effects of considering more than one objective function motivated using a multi-objective optimization method.

Chapter 4

DEFINITION OF THE PROBLEM

In order to use Pareto_Evolve for a simulation model, I need to define the criteria to be used. Then, using these criteria, Pareto_Evolve solves the problem or finds the fit with as many achieved criteria as possible, constructing iteratively a Pareto set, i.e., a set of non-dominated individuals. The Pareto set is expected to satisfy more criteria as the search evolves.

In Section 3.2, I attempted to fit the model using single-objective methods and the results revealed drawbacks: dependency on the initial points and no relation between hourly data. These results motivated using multi-objective methods and led to selection of particular criteria. This chapter introduces the criteria and, after discussing some aspects about Pareto_Evolve, gives results calculated using the original Pareto_Evolve (Reynolds et al. [38]).

4.1 Criteria for the Problem

As criteria, I use the differences between measured and simulated growth data at the times when the measured growth data take their daily maximums and minimums, because the single-objective methods could not express the relation hourly data, which I considered to be helpful for better fitting. I used the same period of seven days as for the single-objective methods. We might consider it is more informative for a search if we could use more objective functions; thus, all 168 (24 hours $\times$ 7 days) hourly differences could be used as criteria. However, the search would take longer, and the number of *Pareto groups* (members of the Pareto frontier) might be very large because the more objective functions, the more Pareto groups. Figure 4.1 shows that increasing the number of objective functions makes the trade-

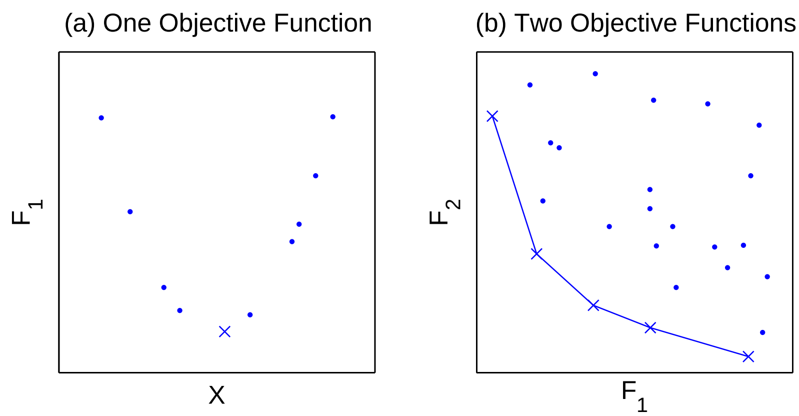


Figure 4.1: Pareto frontiers for minimization problems. Points "x"'s represent the members of the Pareto frontiers. (a) For a single-objective case, the solution set is one-dimensional. (b) For an n-objective ($n > 1$) case, the Pareto frontier can be n-dimensional. In this example, $n = 2$.

offs more complicated. Therefore, I decided to take only two objective functions for each day.

The ideal solution for my example is that all fourteen differences between measured and simulated data are 0. Thus, I could have set all fourteen target values at zero for the search and used a continuous error measure (distance) for each criteria. However, we might imagine that it is hard to make all fourteen differences zero, and I can ignore some small differences in outputs for my problem. Therefore, I decided to use binary error measures (Reynolds et al. [38]). Under a binary error measure, if the prediction output is within the predetermined objective target range, the criterion corresponding to that difference is considered to be achieved; otherwise, the criterion is unachieved.

Initially, and to examine how the method could be used, twelve-hour sums were used as objective functions, which would capture the pattern of diurnal change and the trends over time. The periods for the sum were:

period 1: 6:00 on day 178 $\rightarrow$ 18:00 on day 178

period 2: 18:00 on day 178 $\rightarrow$ 6:00 on day 179

$$\begin{array}{llcl} \text{period 3:} & \text{6:00 on day 179} & \to & \text{18:00 on day 179} \\ \vdots & & & \\ \text{period 14:} & \text{18:00 on day 184} & \to & \text{6:00 on day 185.} \end{array}$$

Thus, the following are the objective functions:

$$\sum_{j=6}^{17} d_{1j}\,,\ \left(\sum_{j=18}^{23} d_{1j} + \sum_{j=0}^{5} d_{2j}\right),\ \sum_{j=6}^{17} d_{2j}\,,\ \ldots,\ \left(\sum_{j=17}^{23} d_{7j} + \sum_{j=0}^{5} d_{8j}\right),$$

where d_{ij} is the absolute difference between measured and simulated growth data from time j to $j+1$ on the i-th day. When time $j = 23$, we consider $j+1$ as time 0 on the next day.

The trend of the sums of measured data for twelve-hour period with starting time 6:00 is shown in Figure 4.2. These sums capture the diurnal cycle and the general trend over the period. The value increases and decreases by turns, showing the contraction and expansion phases.

For each of the objective functions for twelve-hour sums, I used the target range $[0, 5]$. If the sum of absolute differences between measured and simulated data is less than five, the zigzag pattern remains detectable. That was the reason why I set the target range at $[0, 5]$. terms of the hourly difference, I divided five (width of the target range) by twelve (hours for the sum), and then obtained the value 0.42. I considered the sum of absolute differences, so the new objective search could be set at $[-0.42, 0.42]$. However, I set it with half a $[-0.21, 0.21]$ for each criteria, which is more stringent, since not all absolute differences were positive or negative at the same time.

I introduced parameter search ranges in Table 3.2 and set the step size for the search for all parameters at 10^{-3} because the objective ranges are to two decimal places.

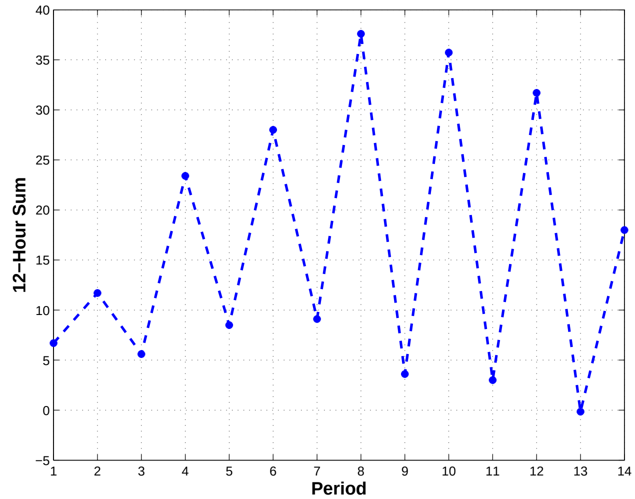


Figure 4.2: Twelve-hour sums of measured shoot growth data. The first period is from 6:00 through 18:00 on day 178 and the last period is from 18:00 on day 184 through 6:00 on day 185. The upper and lower parts include expansion and contraction periods, respectively.

4.2 Aspects of Pareto_Evolve that Need to Be Investigated

There are a number of choices for the user of Pareto_Evolve: population size, generation number, fitness parameters defined in Section 2.4 such as σ_{share}, W, and mutation parameter β. There are also a number of features that the user cannot modify in the original version of the software: the definition of fitness, crossover/mutation probability, methods of crossover and mutation, and selection method. All of these features likely influence the resulting solutions, Pareto set, but to different degrees. However, as I will show, if I modify the selection method and crossover/mutation probability, the search can more efficient, that is, Pareto groups achieving many criteria will be found faster and kept to the next generations. After showing the results from the original version of Pareto_Evolve in next section, I will show in the later chapters how these modifications were executed.

The population size and maximum generation number require careful thought. Small population size can result in low "genetic" diversity. The population may not change much with just a few generations. On the other hand, too large a population or too many generations may not be realistic since run time could become prohibitive. The relation between population size and generation number is the task we need to consider. The result of several different combinations of population size and generation number are shown in next section. Therefore, increasing the number of criteria increases the dimension of the Pareto frontier; more codominant individuals are obtained (Figure 4.1). We can assume that the population size needs to be increased if a criterion is added.

In my example, all criteria are commensurable, so if more criteria are achieved with reasonable parameterizations, I expected the resulting simulation would capture the measured data well. However, I found that the number of achieved criteria did not result in a good fit to the measured data. Actually, exploring the Pareto frontier revealed a deficiency of the model (Chapter 10).

Other selection methods are also worth examining since, as discussed in Section 2.2.2, roulette wheel selection, which is defined by the fitness for each individual, could have a drawback on the probability of selecting parents if there is an individual with very high selection probability. In selecting parents, elitism is also an issue to be considered: how to define elites and how long to set the proportion of elites within the population.

Whether I should hardcode parameter crossover/mutation probability may depend on the problem being investigated. Since we expect that the population converges to the solution set as the generation number increases, the probability of making a smaller change rather than a larger change in a value of an individual should increase with generation number. That is why the crossover probability decreased while the mutation probability increased in the original version of Pareto_Evolve as the generation number increased. However, if the crossover/mutation rate is changed dynamically depending on the performance of the Pareto frontier, search may become more effective. I investigated this thought, and the results are shown in Chapter 6.

4.3 Results of the Original Pareto_Evolve

We saw that parameterizations by single-objective methods did not fit the model very well. I could not tell if that happened because of selection of the search method, the objective function or the model itself. I chose the RSS as the objective function since it is commonly used for single-objective methods, but some other function could be used; such as the sum of hourly differences between measured and simulated data with weights. I tried three different single-objective methods, and as long as the initial points were chosen properly, similar results were obtained; thus, the search methods were probably not the problem in this case. Another possibility is inadequacy of the model. This problem will be discussed with an analysis of the Pareto frontiers that result from Pareto_Evolve in Chapter 10.

Actually, the search by the original Pareto_Evolve did not work well either. I encountered two problems when using the original Pareto_Evolve. The first concerned the resulting Pareto frontier and this is illustrated by considering the trends in the maximum and minimum number of achieved criteria of two trials (Figure 4.3). For my example, if a Pareto group achieves many criteria, I wish to keep it because it may produce a good fit. I succeeded in finding Pareto groups that achieved ten criteria out of fourteen. However, the maximum number of the achieved criteria by a Pareto group was not guaranteed to keep or increase its value when the generation number increased. As shown in Figure 4.3, the maximum number sometimes decreased, which means that the Pareto frontier degenerated. For example, consider a typical case where the maximum number of achieved criteria is nine at generation 50, that is, each of the Pareto groups achieves at most criteria, and there is at least one Pareto group that achieves nine criteria. At the next generation, generation 51, there is no guarantee that at least one Pareto group achieves nine or more criteria; it could happen that all Pareto groups achieve at most only eight criteria, i.e., the "best" one could be lost from the previous generations.

I executed the original Pareto_Evolve ten times with a population size of 100 and a generation number of 500. Figure 4.4 shows it is not guaranteed that the more criteria achieved,

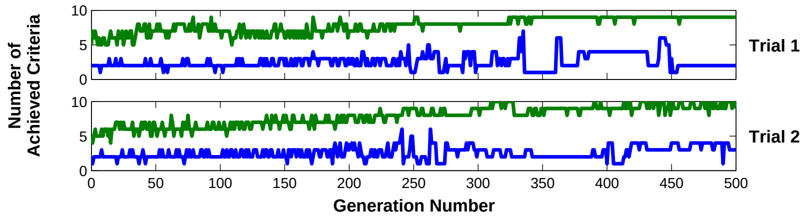


Figure 4.3: Maximum number of achieved criteria by a Pareto group (upper lines) and minimum number (lower lines) out of fourteen criteria for two trials; if the maximum and minimum are 3 and 9, respectively, then each Pareto group achieves at most nine and at least three criteria, and there is at least one Pareto group that achieves nine criteria and one that achieves three criteria. The y-axis indicates the number of the achieved criteria. Population size is 100. The maximum number sometimes decreases, which means that the Pareto frontier degenerates.

the better the simulated data fit the measured data. Simulated growth data by an individual corresponding to a Pareto group achieving only four criteria (Figure 4.4(b)) had a smaller error (ratio of the RSS between the measured and simulated data to the total sum of squares of the measured data) than that achieving the most criteria (Figure 4.4(a)). For example, in Figure 4.4(a), the eleventh criterion, where the measured growth takes the minimum on day 183, was achieved, but the simulated results did not capture the measured data during the contraction period on day 183. This is because achieving criterion 11 did not guarantee a good fit at the next time to criterion 11. Also, the smallest error among all ten trials was 26.6689%, which is much larger than that of the single-objective methods.

Second, the obtained parameter values were not stable. Figure 4.5 shows values for each of the seven parameters of non-dominated individuals at each generation for trial 1. Values of x_1, x_5, x_6 and x_7 became stable before or around generation 300. On the other hand, values of parameter x_2, x_3 and x_4 did not become stable even after a large number of generations; the search did not stop showing considerable fluctuation of their values. I show a result of only one trial in Figure 4.5, but the trend was similar to those of other trials; although the range of values of each parameter around the last generation was different from trial by trial, values of x_1, x_5, x_6 and x_7 became stable, but values of parameter x_2, x_3 and x_4 did

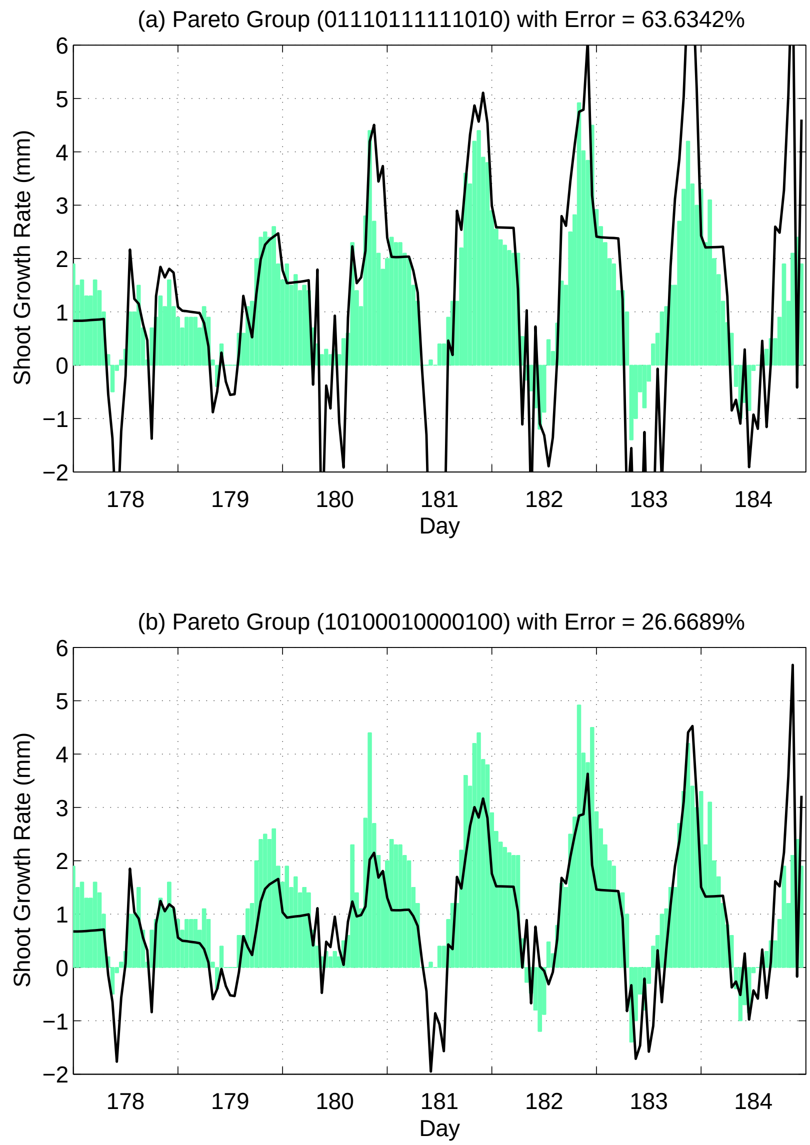


Figure 4.4: The simulated data for the period of seven days (solid line), and the measured data (bar). Population size is 100, and generation number is 500. (a) Simulated data by the individual corresponding to the Pareto group which achieved the number of criteria with the smallest error (ratio of the RSS between the measured and simulated data to the total sum of squares of the measured data) among all ten trials. (b) Simulated data by the individual corresponding to the Pareto group with the smallest error among all ten trials.

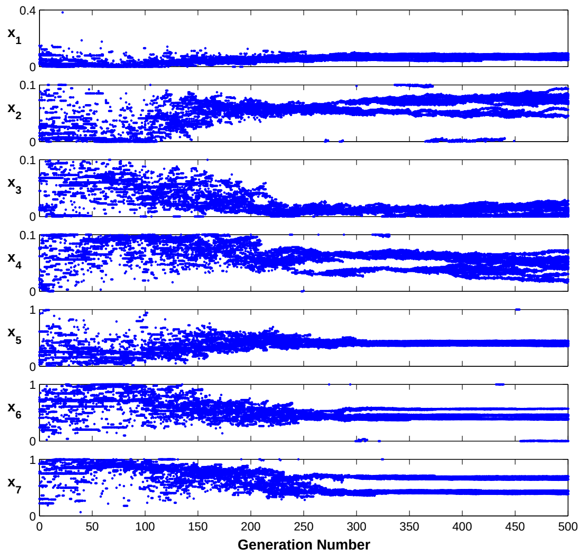


Figure 4.5: Parameter values of all non-dominated individuals found, by generation, for trial 1. Population size is 100. Values of x_1, x_5, x_6 and x_7 stabilize before or around generation 300, or they do not change values; however, values of parameter x_2, x_3 and x_4 do not become stable, even after a large number of generations. Although there are two distinct ranges of values for parameters x_6 and x_7, they may represent values belonging to different Pareto groups.

not become stable.

I also doubled and halved the error measure intervals for all fourteen criteria ($[-0.42, 0, 42]$ and $[-0.105, 0.105]$, respectively) to see how that impacts the stability of the parameter values. While values of x_1, x_5, x_6 and x_7 became stable before or around generation 300, values of parameter x_2, x_3 and x_4 did not become stable even after a large number of generations although the range of values of each parameter around the last generation was from trial by trial. Thus width of error measure intervals does not affect the stability of the parameter values.

In an attempt to fix degeneration of the Pareto frontier and instability of some parameter values , I introduced *elitism* into the original Pareto_Evolve, where the "best" individuals are carried over from generation to generation; they are just copied to the next generation until better ones join the population. The best individuals are usually considered to be ones having the highest fitness values that are reasonable for univariable optimization problems. However, one of my goals for elitism differs in that it concentrates on making the Pareto frontier non-degenerate; the maximum number of achieved criteria never decreases. Therefore, I choose elites depending directly on their objective values, instead of their fitness values. By this elitism, the Pareto frontier increases the number of achieve criteria or at least keeps it, and once the Pareto frontier reaches the state that it cannot be improved anymore, we expect that the Pareto groups do not change much. Then, the parameter values become stable. Therefore, with this technique, I expect to obtain a Pareto frontier, with as many achieved criteria as possible, faster.

To see how the Pareto frontier changes depending on population size and generation number, I ran Pareto_Evolve with five different sets of population size and generation number; the product of population size and generation number is fixed at 200000:

Population size	10	50	100	200	500
Generation number	20000	4000	2000	1000	400.

The results are plotted in Figure 4.6. I tried five times for each of the five combinations (5×5 runs). Box and whisker plots show the distributions of the number of achieved criteria by a Pareto group for each the five trials (Figures 4.6(a)-(e)). If the size was very small, 10, then even though the search took place longer, each of the Pareto groups achieved only a small number of criteria (Figure 4.6(a)). If the population was large enough, 50 or more, then the larger the population size, the larger the maximum number of achieved criteria by a Pareto member (Figures 4.6(b)-(e)). The maximum number of achieved criteria was affected by population size more than by generation number since it increased as the population size became larger, but not as the generation number became larger. On the other hand, minimum number of achieved criteria was not much affected by the population size or generation number.

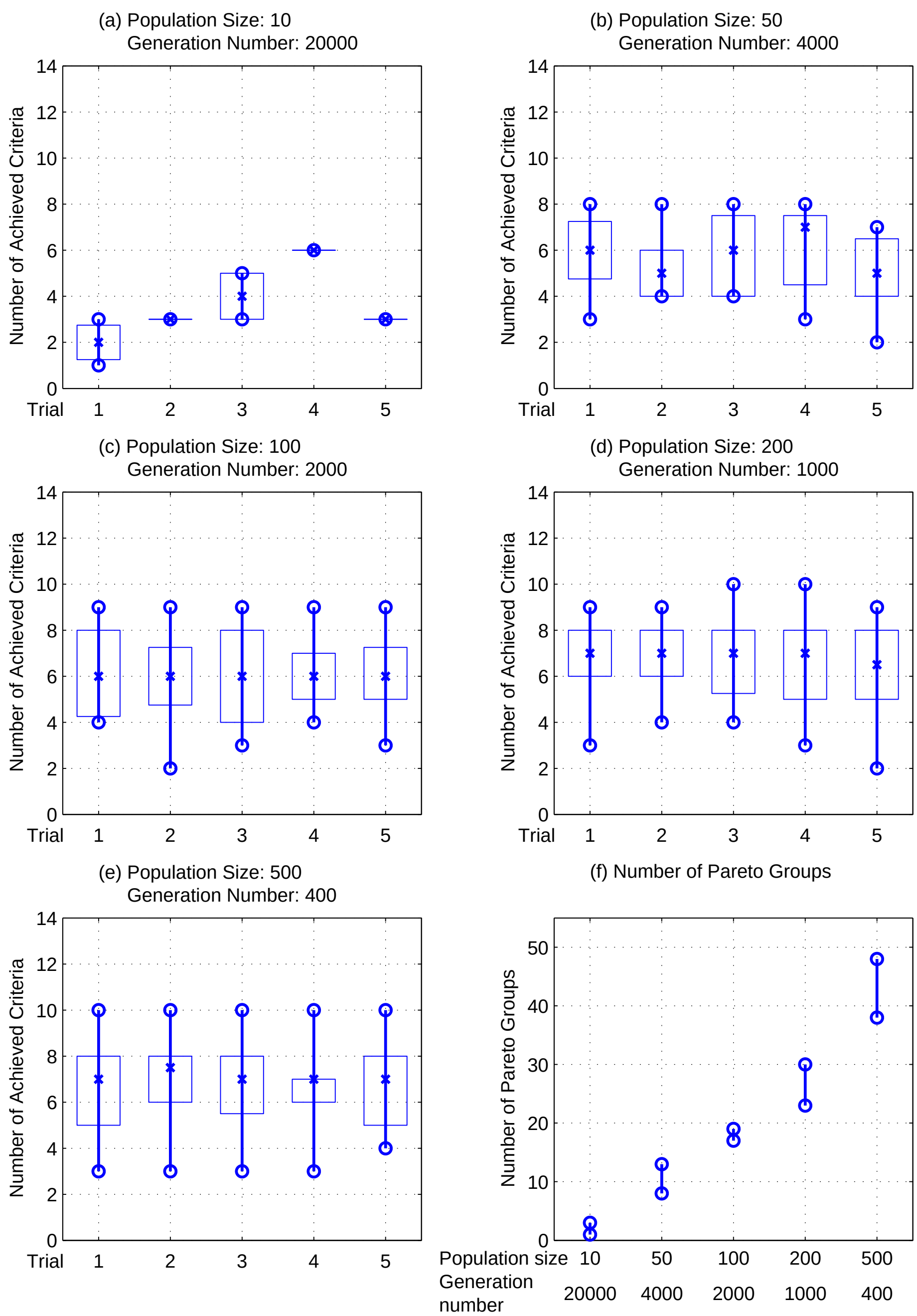


Figure 4.6: Change of the Pareto frontier for five different sets of a population size and generation number; the product of population size and generation number is fixed at 200000. (a)-(e) Box and whisker plots of the number of achieved criteria by a Pareto group for each of the five different trials. Each box represents a range between the first and third quartiles. Symbol "o"'s shows the smallest and largest values, and "x"s represent the median. (f) Range of the number of distinct Pareto groups by a trial.

Also, the number of distinct Pareto groups by a trial increased as population size increased (Figure 4.6(f)). Up to population size 100, the number of distinct Pareto groups was almost proportional to population size; however, from population size 100 to 500, the ratio became smaller. This may have happened because the number of Pareto groups reached the limit. Since both the number of criteria achieved by a Pareto group and the number of distinct Pareto groups by a trial were affected by population size more than by generation number, I concluded that increasing population size is more effective than running for many generations.

Chapter 5

INTRODUCING ELITISM TO PARETO_EVOLVE

5.1 Elitism

As I explained in Section 2.2.2, an elite is a “good” individual for the next generation. For the elitism used in some single-objective optimization methods, the ratio of elites to the population size is fixed for all generations, and the population members that have the highest fitness are copied to the next generation; for Das et al. [7] set the ratio at 20%, which means 20% of the population will be copied to the next generation as elites, and the ratio varies between 0% and 70% in Grefenstette [20]. In elitism by De Jong [8] and Moilanen [32], only the single best individual is copied to the next generation.

Bäck et al. [1] introduced the Evolutionary Strategies (ESs), which are also elitist methods, but with a different type of elitism. For a $(\mu + \lambda)$-ES, μ parents produce λ offspring, and among those $\mu + \lambda$ individuals, the best μ individuals are selected as parents for the next generation.

In contrast to the single-objective case, when comparing two individuals in multi-objective optimization, we cannot always determine which individual is better. For example, on minimizing two objective functions $F_1(x) = x^2$ and $F_2(x) = (x-2)^2$ (Figure 2.2), it is impossible to determine whether the superior solution is $x = 0.5$ or $x = 1.5$ because $F_1(0.5) < F_1(1.5)$ and $F_2(1.5) < F_2(0.5)$. Thus, if we assume that the number of elites is set to be μ and that there are ν non-dominated individuals with $\nu > \mu$, how should we choose the μ elites from among ν individuals? In order to use a constant number or proportional elitism method, we would need to order individuals having the same rank. If so, we could choose μ best individuals among ν non-dominated individuals as elites. For instance, an individual could

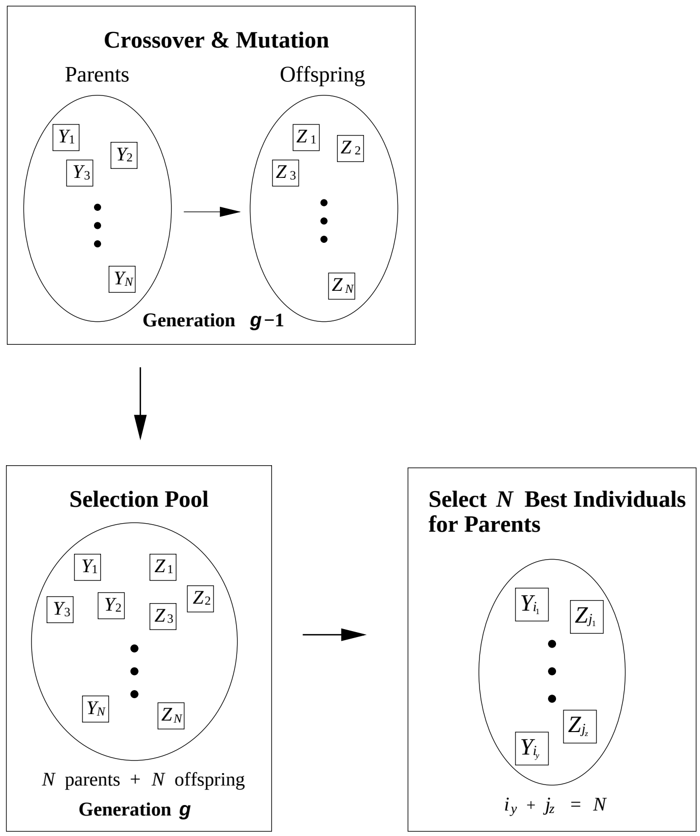


Figure 5.1: Elitism for the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II). At generation $g-1$, N parents produce N offspring, and then N best individuals are chosen from those $2N$ individuals (N parents and N offspring) at generation g. N parents are selected among those N best individuals by tournament selection, and crossover and/or mutation operators are applied to the parents to produce N offspring.

be assigned a high probability of being selected as an elite if the normalized Euclidean distance in the parameter space between it and its closest individual having the same rank is large; if the distance is small, then it has low probability.

Different types of elitism are used in multi-objective optimization. Here I introduce two examples. In the Elitist Non-Dominated Sorting Genetic Algorithm (NSGA-II) (Figure 5.1) introduced by Deb et al. [10], parents of the current generation are selected among parents and offspring of the previous generation. At generation g, first, the parents and offspring of generation $g-1$ are combined. Then the individuals in the combined set are divided into groups with respect to Pareto ranking (Section 2.4); F_r denotes the set of individuals with rank r. From $r=1$, all individuals in F_r are added to the pool for parents as long as the total number of individuals does not exceed population size N; $Q=\{F_1,\ldots,F_q\}$ is the pool for parents. If the size of Q is N, then Q becomes the set of parents. Otherwise, members with the largest "crowding distance" (Section 7.1) in F_{q+1} are added to Q until the parents population is fulfilled. From Q, individuals are selected by tournament selection, and then are crossed over and/or mutated to produce offspring. Since parent generation participates in selection, this is an elitist method.

The Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler et al. [45] uses another method (Figure 5.2). A number μ is chosen in advance, and the greatest μ of the non-dominated individuals that were created from generation 0 are stored as elites externally, like the Historical Pareto Frontier. At each generation g, all non-dominated individuals of the current population P_g are copied to the external non-dominated set P', and all dominated and duplicate individuals are removed from P'. If the size of P' exceeds μ, the extra individuals are removed by means of "clustering", which is defined depending on crowdedness. Fitness of a member X' of P' is proportional to the number of individuals of P_g that are dominated by or identical to X'. On the other hand, for an individual X of P_g, fitness is calculated by summing the fitness values of members of P' that dominate or are identical to X. One is added to the sum so that individuals of P' have better fitness values than those of P_g. Thus, fitness of an individual of the current population P_g is evaluated depending only on the external non-dominated set P' with μ elites. Parents are selected from P' and P_g. Thus, elites also participate in evolution, instead of just being copied as population members of the next generation. For test problems for the SPEA, only a few objective functions were used (Zitzler et al. [45] [44]). However, if there are many objective functions, then many different Pareto groups are obtained. In that case, it is hard to keep diversity if μ is too small.

For both NSGA-II and SPEA, elites participate in evolution. The difference in elitism between these two methods is in the selection pool for parents; while they are selected from

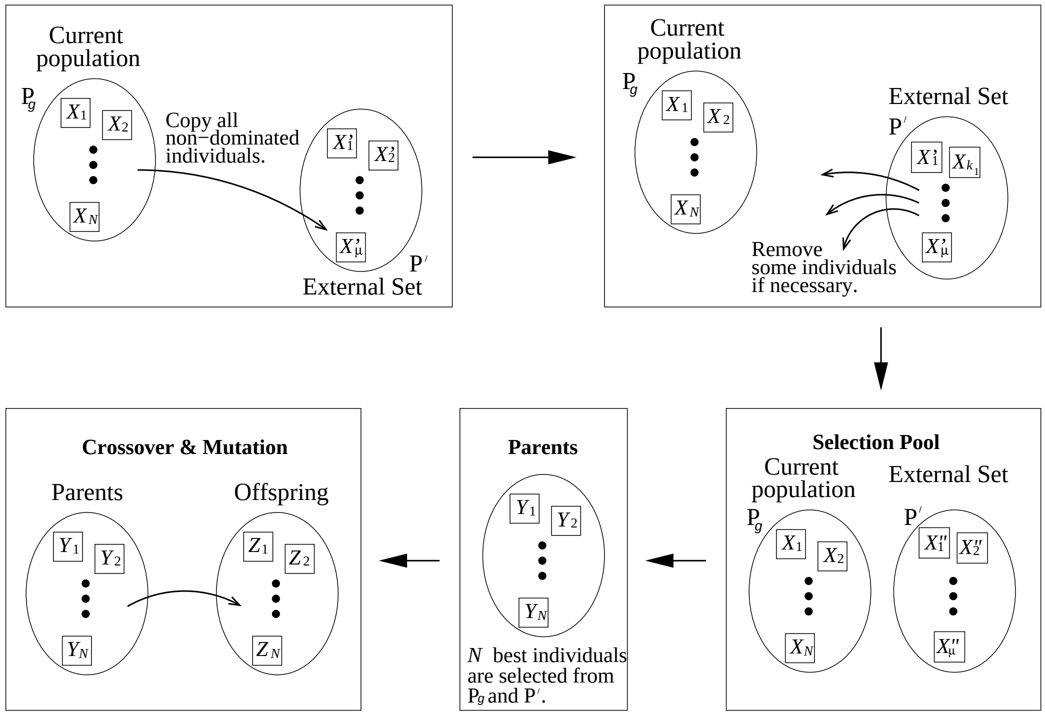


Figure 5.2: Elitism for the Strength Pareto Evolutionary Algorithm (SPEA). First, all non-dominated individuals in the current population P_g are copied to the external set P' for elites. Then, the dominated individuals in P' are removed. After that, if the number of individuals in P' exceeds prefixed number μ, some individuals are removed until the number of individuals in P' reaches to μ; this reduction is done using "clustering". N of the individuals are selected as parents from the current population P_g and the external set P'. Crossover and mutation operators are applied to the parents to produce N offspring.

parents and offspring of the previous generation for NSGA-II, for SPEA, they are chosen from the current population members and the external set.

Test problems for NSGA-II and SPEA did not have as many objective functions as my case. As I mentioned before, generally, the more objective functions used in a search, the more non-dominated individuals and Pareto groups occur due to the geometry of the objective space (Figure 4.1). My problem has fourteen objective functions, which means many Pareto groups with a small number of corresponding individuals could be obtained. If elitism of NSGA-II is used, there may be more than N non-dominated individuals in the selection pool

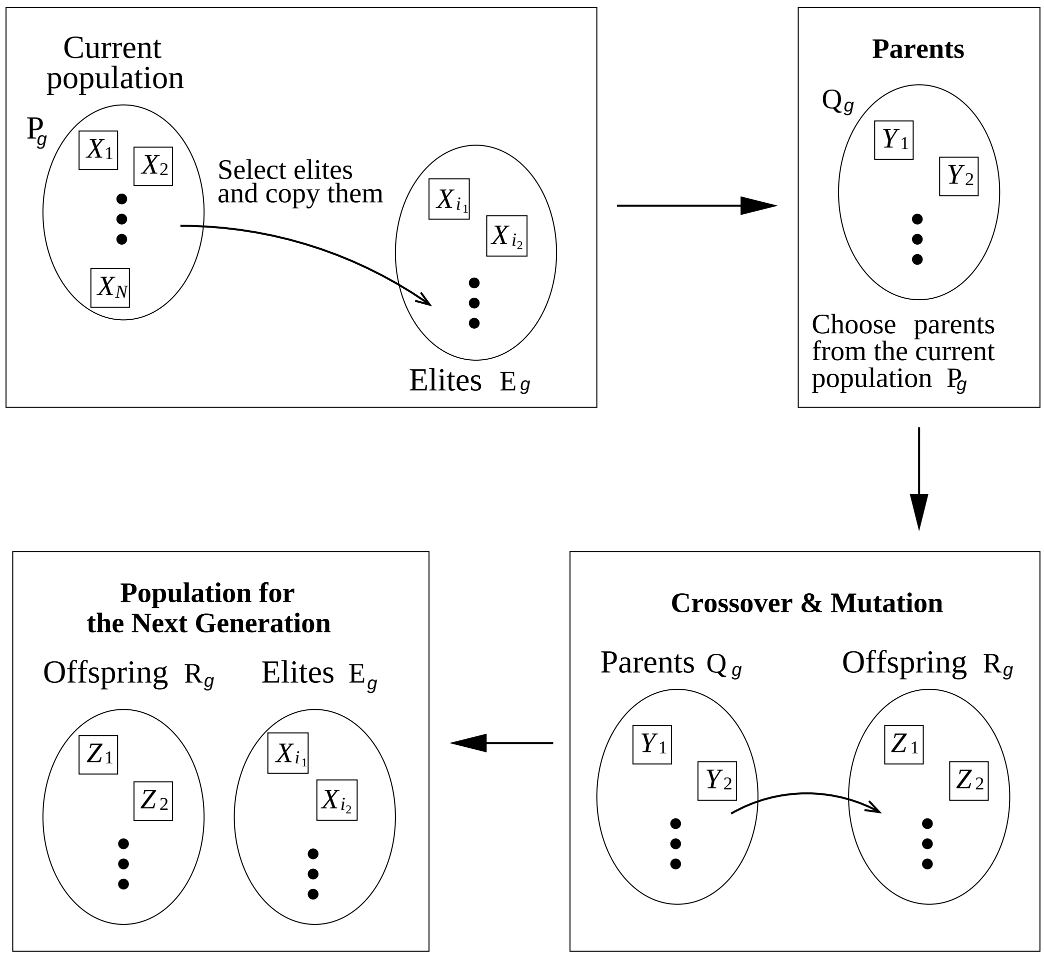


Figure 5.3: Elitism for Pareto_Evolve. Elites are selected from the current population P_g under certain conditions, which will be introduced later for each experiment. The number of elites is not restricted but is determined depending on performance of Pareto frontier. Parents are chosen from P_g so the total number of the elites and the parents becomes the population size N. Each of the parents is crossed over or mutated. The resulting offspring and elites become the population for the next generation.

with $2N$ individuals (N parents and N offspring), and individuals with only Pareto groups achieving a small number of criteria might be selected as elites. For elitism of SPEA, the similar thing could happen because N best individuals from the current population P_0 and the external non-dominated set P' again could be individuals with Pareto groups achieving only a small number of criteria. Thus, I defined the elitism depending on performance of Pareto groups. Unlike NSGA-II or SPEA, elites do not participate in reproduction. However, although elites are not bred, individuals chosen as elites still have a chance to participate in reproduction since they may be selected as parents. Figure 5.3 shows how the new elitism works; Q_g, R_g and E_g denotes sets parents, offspring and elites, respectively.

I started with elitism with simple conditions. I analyzed the results and refined the elitism so that better results could be obtained. In the first of the following experiments, I define three conditions for elitism; a non-dominated individual becomes an elite if it satisfies one of them. However, two of them turned out to produce almost no elites, and only several individuals were selected as elites under the third condition at each generation. Since this experiment did not work as effectively as I had expected, I revised it for the second experiment. Section 5.1.1 explains the problems of the first experiment, and Section 5.1.2 explains the revision.

5.1.1 Experiment One: Three Initial Criteria for Defining Elites

This procedure compared the current Pareto groups to the previous ones to see whether improvements in achieving assessment criteria had been made. If a Pareto group satisfied one of the following three conditions, I defined its corresponding individuals as elites:

1. It achieved a criterion that had not been achieved by any of the Pareto groups of the previous generation.

2. It achieved more criteria than the maximum number of criteria achieved by a Pareto group of the previous generation.

3. It is better than at least one of the Pareto groups of the previous generation. (Note: P_1 is better than P_2 means the corresponding individuals of P_1 dominate those of P_2, and that is denoted by $P_1 \succ_p P_2$.

It seems the third condition includes the second one. However, it is important to note that a Pareto group satisfying the second condition does not always meet the third one. For instance, at generation $g-1$, if the obtained Pareto frontier consists of three groups P_i^{g-1} $(i = 1, 2, 3)$:

Pareto group	Criterion 1	2	3	4	5	6	7	8	9	10	11	12	13	14
P_1^{g-1}:	X	X	X	0	X	X	0	0	0	X	0	0	0	X
P_2^{g-1}:	X	X	0	X	X	0	0	0	0	0	0	0	0	0
P_3^{g-X}:	X	0	0	X	X	0	0	0	0	0	X	0	0	0

where 0 and X denotes unachieved and achieved, respectively. At the current generation g, we have three members P_i^g $(i = 1, 2, 3)$:

Pareto group	Criterion 1	2	3	4	5	6	7	8	9	10	11	12	13	14
P_1^g:	0	X	X	X	X	X	0	0	0	X	X	0	0	X
P_2^g:	X	X	0	X	X	0	0	0	0	0	0	0	0	0
P_3^g:	X	0	0	X	X	0	0	0	0	0	X	0	0	0

In this case, neither P_2^g nor P_3^g is selected for elitism since $P_2^{g-1} = P_2^g$ and $P_3^{g-1} = P_3^g$, but P_1^g is selected by the second condition because the maximum number of criteria achieved by a Pareto group at generation $g-1$ is 7, and 8 for generation g.

Results of two trials from among ten using the three conditions for selecting elites are presented in Figure 5.4. Population size is 100 and generation number is 500 for both trials. In

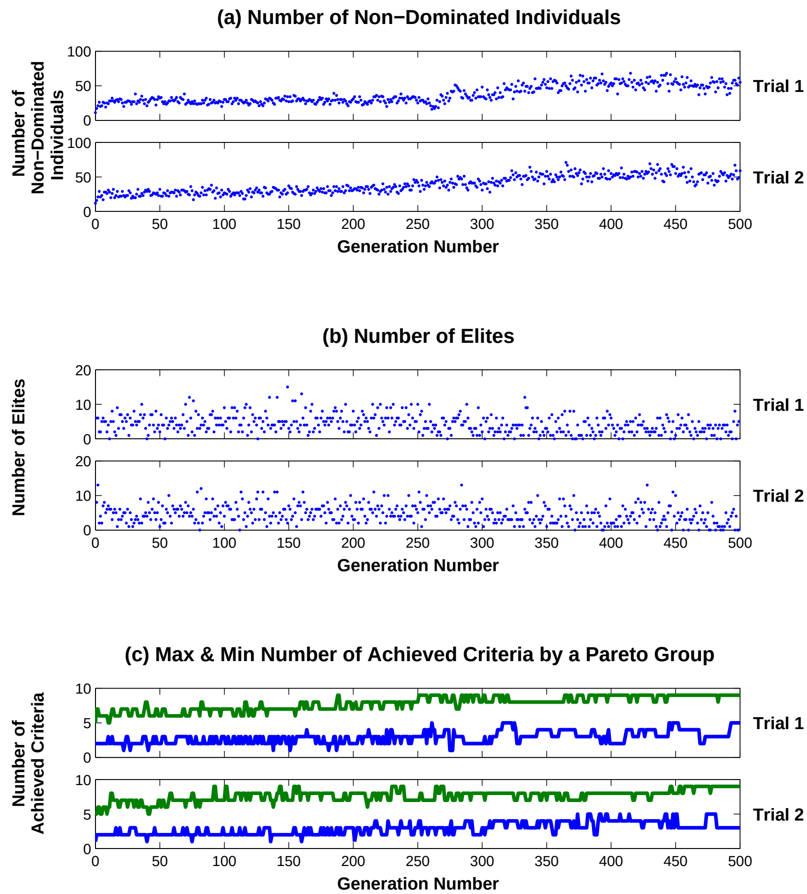


Figure 5.4: Results of two different trials in Experiment One with a population size of 100. (a) The number of non-dominated individuals changes generation by generation. (b) The number of elites fluctuates. I can infer from this that it is very likely that the Pareto frontier improves and degenerates alternatively because an improved Pareto frontier results in more elites. (c) As for the results of the search without elitism, the Pareto frontier often degenerates; i.e., the maximum number of achieved criteria by a Pareto group decreases.

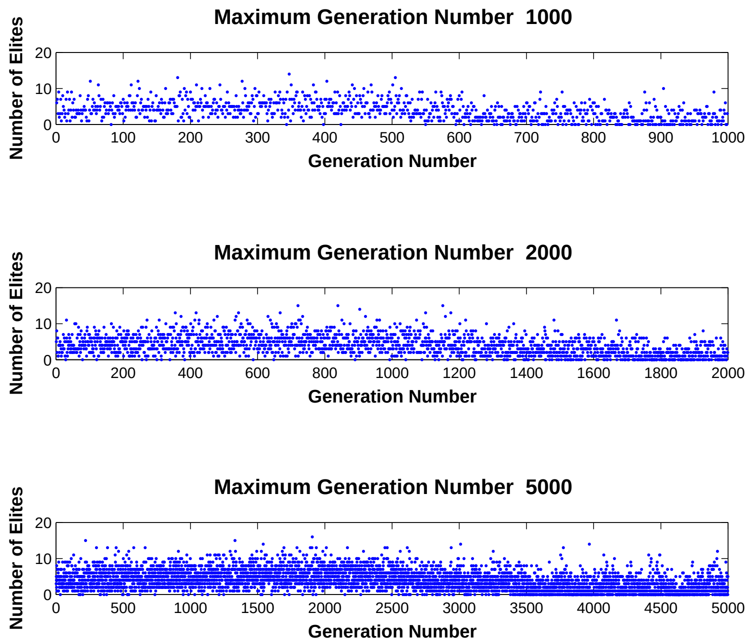


Figure 5.5: The number of elites fluctuates even though the maximum number of generation increases.

a successful search, parameterizations would converge, and the number of non-dominated individuals and elites would become stable. However, the number of non-dominated individuals changed generation by generation (Figure 5.4(a)), and the number of the elites fluctuated (Figure 5.4(b)) between 0 and 7. Thus, although some elites were succeeded from the previous generation but not many are selected as elites at the current generation, then the succeeded elites could die out. For the above example, corresponding individuals of P_1^g will be copied to generation $g+1$; however, P_1^g will not select for elitism, so even though P_1^g was good, it could die out. Even though I increased the maximum generation number to 1000, 2000 and 5000, the number of elites still fluctuated (Figure 5.5). Thus, I suspected that this elitism did not work effectively.

Also, in some cases, as for the result of the search without elitism, I found that the Pareto frontier degenerated; i.e., the maximum number of achieved criteria by a Pareto group sometimes decreased (Figure 5.4(c)). Actually, by the second condition, it is guaranteed that the Pareto frontier improves only if the maximum number of achieved criteria becomes larger than that at the previous generation. So, if there is no elite selected under condition 2, the Pareto frontier is not guaranteed to improve. As mentioned above, such elites were not obtained very often. That is why the degeneration happened. These fluctuations also suggest that this procedure for elitism does not work effectively.

The Pareto frontier and the Historical Pareto Frontier of trial 1 at the last generation, generation 500 are plotted in Figure 5.6. The Historical Pareto Frontier at generation 500 is the set of Pareto groups that were obtained at each generation up to generation 500. From those results, we can also see that the elitism did not work well because $P_4 \not>_p P_3^{500}, P_4^{500}, P_5^{500}$ and $P_5 \not>_p P_6^{500}$. Therefore, elitism in Experiment One does not guarantee that the better Pareto groups are kept.

As shown in Figure 5.7, the parameter values are also unstable with Experiment One. As for the search without elitism (Figure 4.5), values of parameter x_1, x_5, x_6 and x_7 became stable as generation increased, but for x_2, x_3 and x_4, values fluctuated. Even though, the maximum generation number is increased, this phenomena remain the same (Figure 5.8(a) and 5.8(b)). This implies that the elitism used in this experiment did not increase stability of parameter values.

Although I only presented two results here, all of the ten trials showed similar results. Thus, I considered these conditions for elitism to have problems. By direct examination, I found that the first condition did not contribute to the selection of many elites. At a very early generation, there may have been a few criteria unachieved by any Pareto groups, but soon there were none left unachieved. The second condition was rarely used because the maximum number of achieved criteria out of fourteen, did not increase very often. The third condition selected several non-dominated individuals as elites; so, only this condition was effective for Experiment One. I concluded that this method of elitism was ineffective. In next section, I experiment with a revised method.

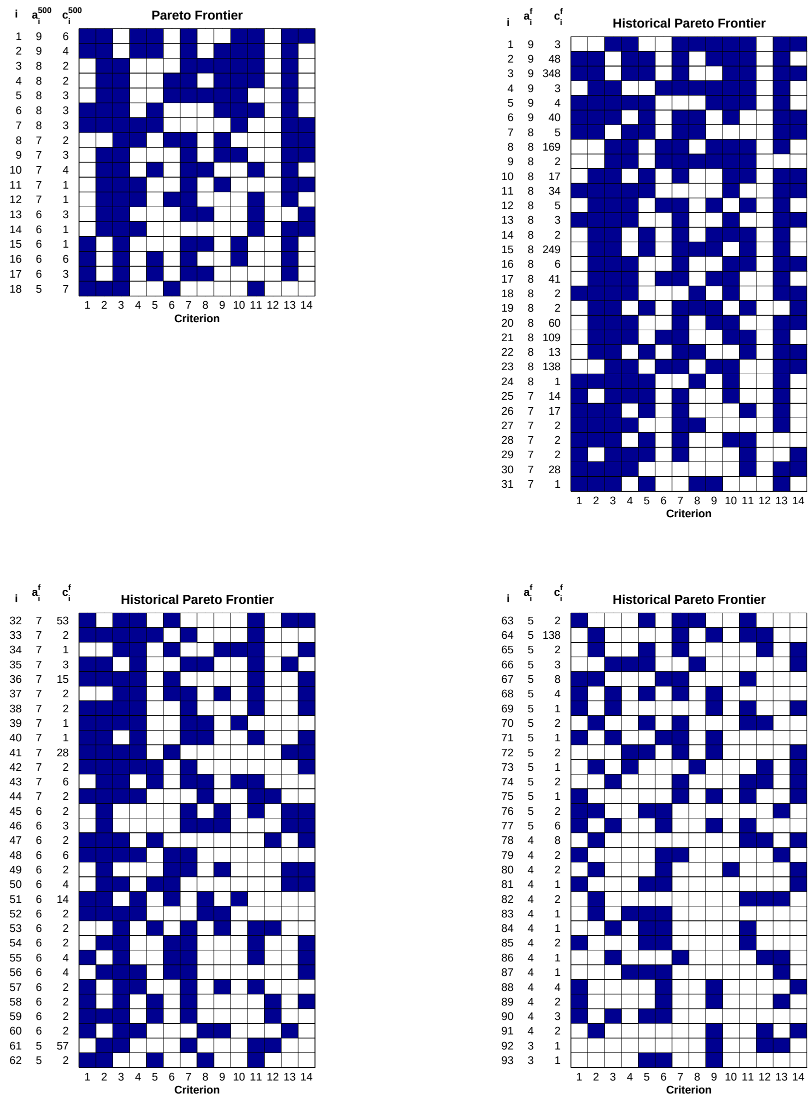


Figure 5.6: The Pareto frontier (top left) and Historical Pareto Frontier at generation 500 (other three) for trial 1 in Experiment One. Three numbers i, a_i^{500}, c_i^{500} presented on the left of the plot for the Pareto frontier are index of Pareto group P_i^{500}, the number of achieved criteria by P_i^{500}, the number of corresponding individuals to P_i^{500}, respectively. Three numbers i, a_i^f, c_i^f on the left of three plots for the Pareto Historical Frontier are similar to those on the plot for the Pareto frontier. Black and white squares represent achieved and unachieved, respectively.

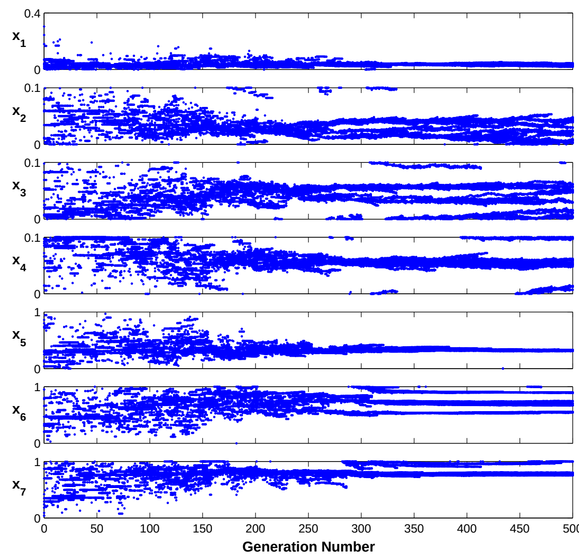


Figure 5.7: Parameter values of all non-dominated individuals found, by generation, for trial 1 in Experiment One. Population size is 100, and the maximum generation number is 500. As for the search without elitism (Figure 4.5), values of parameter x_1, x_5, x_6 and x_7 stabilize as generation increases, but values of x_2, x_3 and x_4 still fluctuate. This shows that the elitism used in Experiment One does not contribute stability of parameter values.

5.1.2 Experiment Two: Elitism to Produce a Non-Degenerate Pareto Frontier

Since I found that the first and second selection conditions for Experiment One did not work effectively, I changed them. The second condition accepted as elites only non-dominated individuals that corresponded to Pareto groups achieving more criteria than the maximum number of the criteria already achieved by an existing Pareto group of the previous generation. Since the maximum number of the criteria achieved by a Pareto group rarely increased, there usually existed no elites selected under condition 2. Also, condition 1 and 3 did not affect the number of achieved criteria. Therefore, if there are no individuals condition 2, we may obtain a degenerate Pareto frontier, i.e., the number of maximum achieved criteria by a Pareto group may decrease. So, I changed this condition for selecting elites so that all of the non-dominated individuals corresponding to the Pareto groups that achieve the

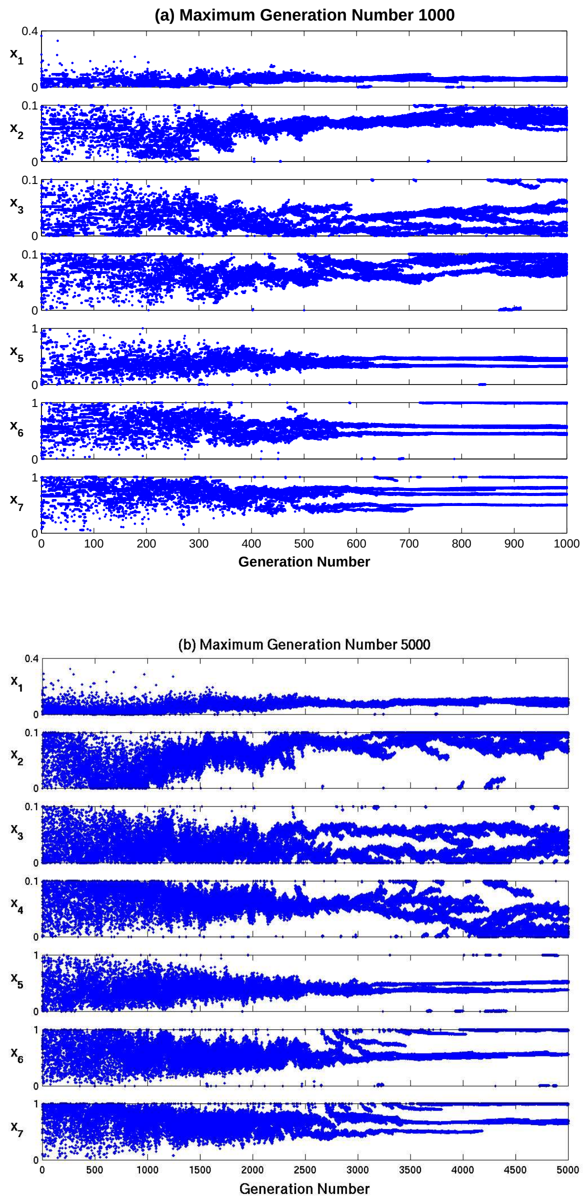


Figure 5.8: Parameter values of all non-dominated individuals found, by generation, for trial 1 in Experiment One. Population size is 100, and the maximum generation number is 1000 (top) and 5000 (bottom). As for the search with the maximum generation number 500 (Figure 5.7), values of parameter x_1, x_5, x_6 and x_7 stabilize as generation increases, but values of x_2, x_3 and x_4 still fluctuate. This shows that those three parameters values are not stabilized even though the maximum generation number is increased. This implies that increasing the maximum generation number does not help stability of those three parameters values.

most criteria at the current generation are selected. This condition guarantees that the Pareto frontier does not degenerate since the maximum number of the achieved criteria by a Pareto group at the current generation is larger than or equal to the maximum number at the previous generation. This is the first condition for Experiment Two. If there are more than one Pareto groups achieving the maximum criteria, all of them are selected for elitism. This condition brings less chance that the problem found on comparing the Pareto frontier and the Historical Pareto Frontier (Figure 5.6) happens; the Pareto groups of the Pareto frontier are no worse than those of the Historical Pareto Frontier as long as they achieved the most criteria. The second condition is the third condition of Experiment One, which had been found useful.

As the third condition for Experiment Two, I accept as elites the non-dominated individuals achieving the criteria unachieved by any elite selected under the first and second conditions. This guarantees that all criteria are achieved by at least one Pareto group at the next generation as long as they are achieved at the current generation.

Therefore, the definition of elitism for Experiment Two is as follows: if a Pareto group satisfied one of the following three conditions, I defined its corresponding individuals as elites:

1. It achieved the most criteria in the Pareto frontier of the current generation.

2. It is better than at least one of the Pareto groups of the previous generations.

3. It achieved a criterion unachieved by any Pareto groups that were chosen under the above conditions 1 and 2.

Experiment Two is different from Experiment One in that a selected Pareto group could keep being chosen for more than one generation because of the the first condition. For Experiment One, none of the selected Pareto group was chosen at the next generation.

After this refinement, the number of elites at each generation was at least one because at least one elite was selected under condition 1. With this method of elitism, Pareto groups satisfying the first and second condition are chosen first. Then, I checked if all criteria are achieved by at least one of the the selected Pareto groups. If not, we look for members meeting the third condition. So, if all criteria are achieved by at least one Pareto group at the current generation, it is guaranteed that all criteria are achieved at the next generation, too.

This selection system for elites was run in ten trials with population size 100 and generation number 500. Results of two of the trials are presented in Figure 5.9. As we expected, the Pareto set did not degenerate (Figure 5.9(c)); the maximum number of the criteria achieved by a Pareto member never decreased. However, both the number of non-dominated individuals and elites nearly or completely reached 100 (i.e., population size) around generation 300 (Figure 5.9(a) and 5.9(b)), so, almost all of the population became non-dominated individuals, and almost all of them were copied to succeeding generations. Therefore, after generation 300, the population did not evolve much or at all. This means that too many elites were selected. I solved this problem in Chapter 7.

As we can see in Figure 5.10, parameter values for trial 1 became stable before generation 300. This is because the search stopped around generation 300 due to selection of too many elites.

The simulated data with the smallest error (the ratio of the RSS to the total sum of squares of hourly measured data for the period of seven days) among ten trials is plotted in Figure 5.11. Compared to the results of the three single-objective optimization methods, the RSS is much larger. I will show how the simulated data changed by refinement of the search method.

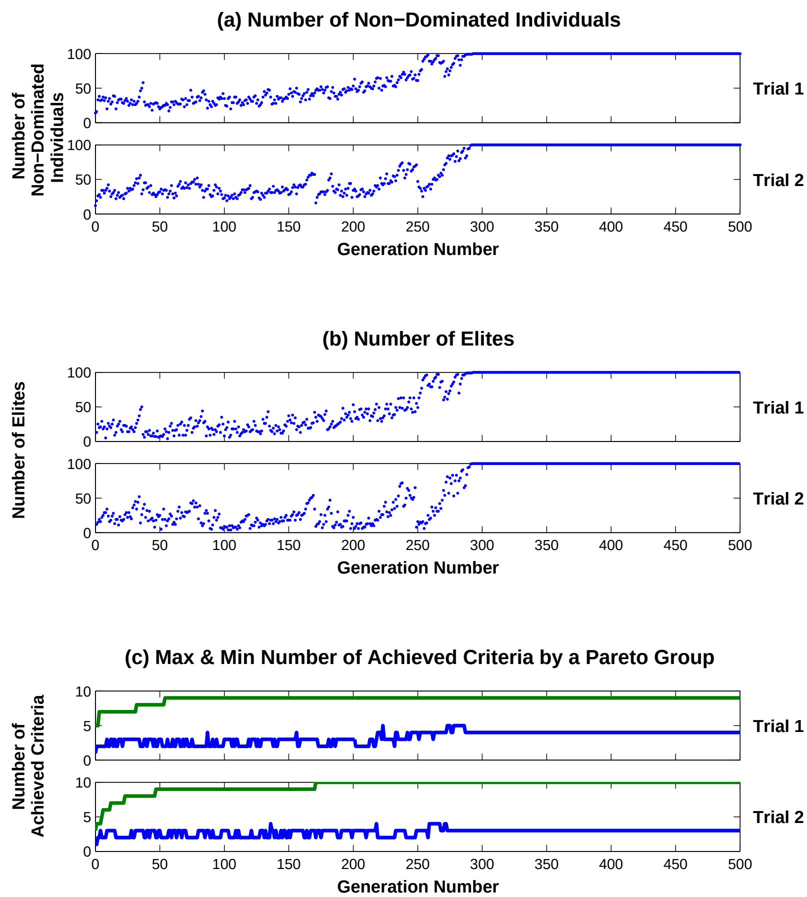


Figure 5.9: Results of two different trials in Experiment Two with a population size of 100. (a) Around generation 300, the number of the non-dominated individuals nearly or completely reaches 100; i.e., most of the population becomes non-dominated individuals. (b) Most of the non-dominated individuals are selected as elites around generation 300, so almost all of the population members are copied to succeeding generation. (c) As I designed, the maximum number of the criteria achieved by a Pareto group never decreases, which means that the Pareto frontier does not degenerate, i.e., the maximum number of achieved criteria by a Pareto group does not decrease.

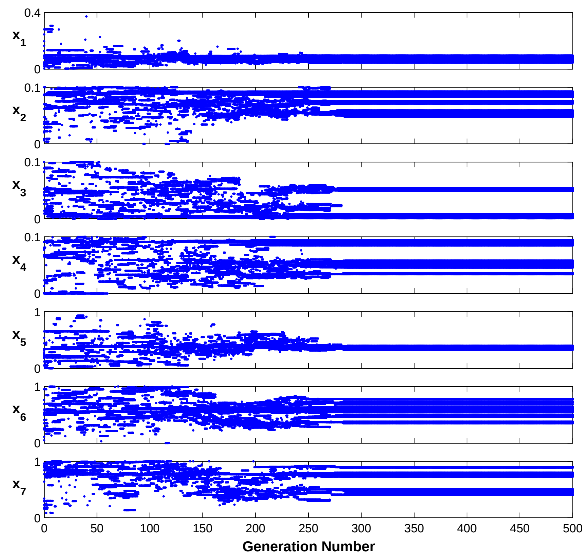


Figure 5.10: Parameter values of all non-dominated individuals found, by generation, for trial 1 in Experiment Two. Population size is 100. All of the parameter values stabilize before generation 300. This is because the search stopped around generation 300 since too many elites were selected.

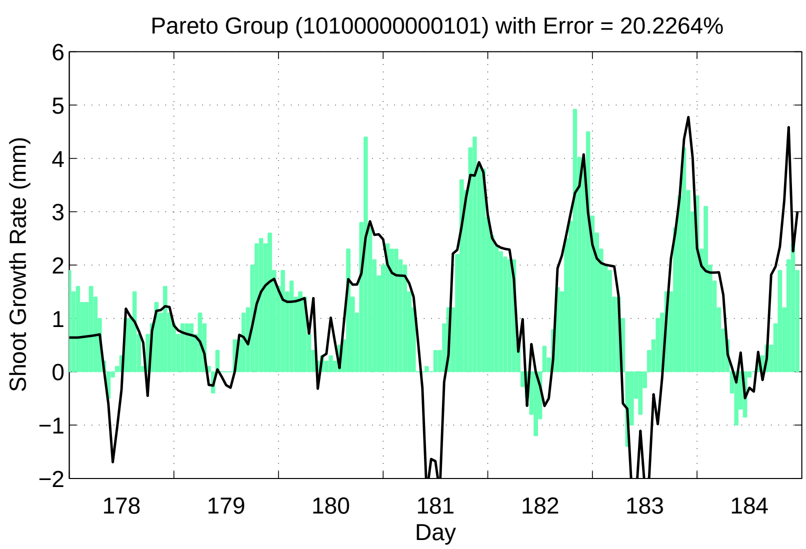


Figure 5.11: The simulated data by the individual corresponding to the Pareto group with the smallest error among all ten trials for Julian day 178 through 184 (solid line), and the measured data (bar). Population size is 100, and the maximum generation number is 500. The error is 20.2264%, which is much larger than those by the single-objective methods (about 13%).

5.2 Effect of Population Size

In Experiment Two in Section 5.1.2, all parameter values stabilized. This is because the search halted. I tried Experiment Two again with different population sizes to see how the population size affects the number of elites or the end of a search. The population size I used in Experiment Two in Section 5.1.2 was 100 for all ten trials. I examined population size 25, 50 and 200 to see a population of 100 is reasonable, not too large or small, for the search. The maximum generation number remains 500 because the number of elites reached the population size before generation 500. I have ten trials for each case.

With a population size of 25, all ten trials showed similar results. I present two trials here. Figure 5.12(a) and 5.12(b), respectively, show that both the number of elites and non-dominated individuals nearly or completely reached 25, the number of the population size, before generation 100. Since the population size is small, the number of Pareto groups is small, too. Thus, because of the definition, many Pareto groups could be accepted for elitism. This is probably the reason why the number of elites reached the population size at a very early generation. The maximum number of criteria achieved by a Pareto group (Figure 5.12(c)) was less than that with a population size of 100 (Figure 5.9(c)) because the search stopped before Pareto groups achieve as many criteria as those with population size 100 did.

I found another difficulty with this small population size. For some of the ten trials, more than twenty Pareto groups were obtained. Since there were only 25 individuals in the population, the correspondence between non-dominated individuals and Pareto groups was one-to-one in most of the cases. I cannot judge whether parameter values are stable local or global optima if I have only a few individuals per Pareto group. Thus, I concluded that population size 25 is too small to obtain reasonable solutions to this optimization problem although the search does not stop because the number of elites reaches the population size.

With a population size of 50, as for a population size 25, for trial 1, the number of elites reached the population size at an early generation; on the other hand, for trial 2, prob-

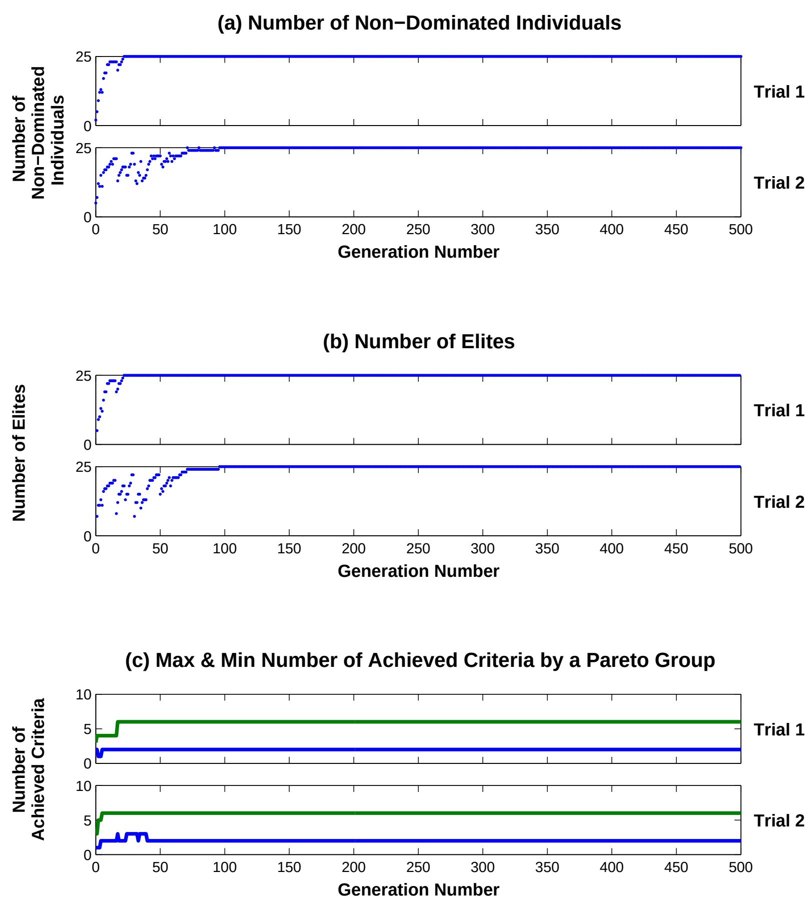


Figure 5.12: Results of two different trials with elitism as specified in Section 5.1.2 with a population size of 25. (a) As for a population size of 100 (Figure 5.9(a)), all population members become non-dominated, but this happens before generation 100, which is much earlier than the case with a population size of 100. (b) The number of elites reached 25 before generation 100, so all of the population are copied to succeeding generation. (c) The maximum number of criteria achieved by a Pareto group was less than that with a population size of 100 shown in Figure 5.9(c).

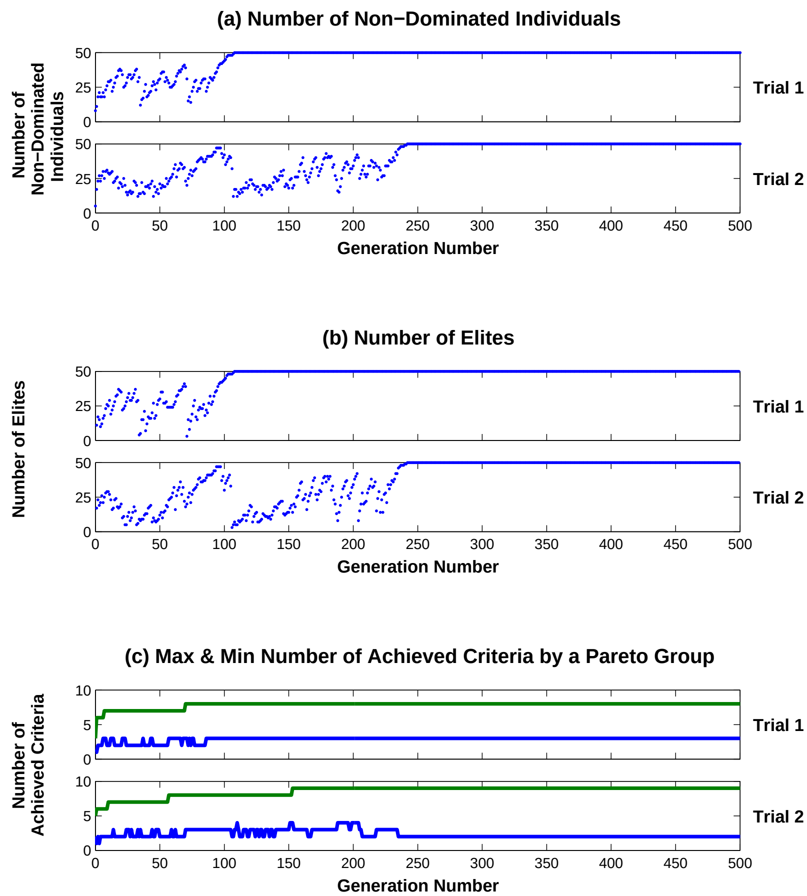


Figure 5.13: Results of two different trials with elitism as specified in Section 5.1.2 with a population size of 50. (a) For both of the two trials, the number of non-dominated individuals reached 50, but the time is different; one is at around generation 100, and the other is at around generation 250. (b) The number of elites behaves about the same way as the number of non-dominated individuals in Figure (a). It is considered that the difference depends on the time when the number of elites reaches the population size; as with a population size of 25, for trial 1,that happens at an early generation; for trial 2, individuals move in the search space more actively before the search stops because of the number of elites. It is probably hard to obtain consistent results with a population size 50. (c) For trial 1, the search stopped around generation 100; so the maximum number cannot increase. However, for trial 2, the search is active until around generation 250 as I see in Figure (a) and (b); thus, the maximum number can increase.

ably individuals moved in the search space more actively before that (Figure 5.13). For the case like trial 2, we have a large maximum number of achieved criteria by a Pareto group at a very early generation, so we may obtain the Pareto frontier whose member has a large number of achieved criteria; because it is guaranteed that the Pareto frontier does not degenerate. Compared to the case with a population size of 25, the population size was doubled. Thus, more combinations of parameter values were obtained, which could bring the Pareto frontier with the large maximum number of achieved criteria at an early generation. However, as we can see in trial 1 in Figure 5.13(c), we do not guarantee the Pareto frontier with a large number of achieved criteria at an early generation. Therefore, if the number of elites reaches the population size, then it is very likely that we do not obtain a Pareto frontier with the large maximum number of achieved criteria at an early generation.

The number of Pareto frontier members in the Pareto frontier varied between twenty-one and thirty-eight in the ten trials. As for a population size of 25, most Pareto groups had only one or a few corresponding individuals. Therefore, I concluded that population size 50 is also insufficient to obtain solution to the optimization problem. It is difficult to tell which solutions are robust.

For all ten trials with a population size of 200, the number of non-dominated individuals and elites nearly or completely reached the number of population size, at around generation 300 (Figures 5.14(a) and 5.14(b)). This is a slightly later generation number than with a population size of 100, which is probably because a larger population size brings more Pareto groups.

Figures 5.15(a)-(d) show the distributions of the number of achieved criteria by a Pareto group for each of the ten trials with population sizes 25, 50, 100, 200, respectively. The mean and the maximum numbers of achieved criteria by a Pareto group became larger as the population size was doubled up to 100, but from 100 to 200, there was not much difference compared to the previous two comparisons. In Figure 5.15(e), we see that the range of the number of Pareto groups with population size 50 is about double to that with population

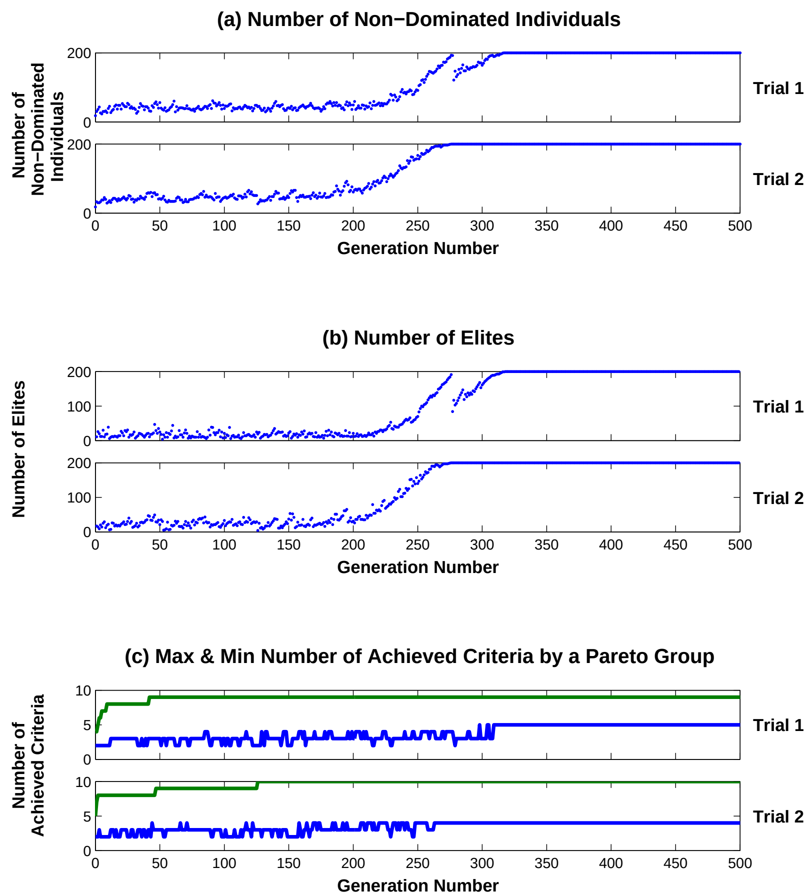


Figure 5.14: Results of two different trials with elitism as specified in Section 5.1.2 with a population size of 200. (a) The number of non-dominated individuals almost or completely reached 200, or all of the population members become non-dominated individuals at around generation 300, which is slightly later than that with a population size of 100. (b) The number of elites reached 200 at around generation 300, which is also slightly later than the case with a population size of 100. (c) The maximum number of achieved criteria by a Pareto group is as large as that with a population size of 100.

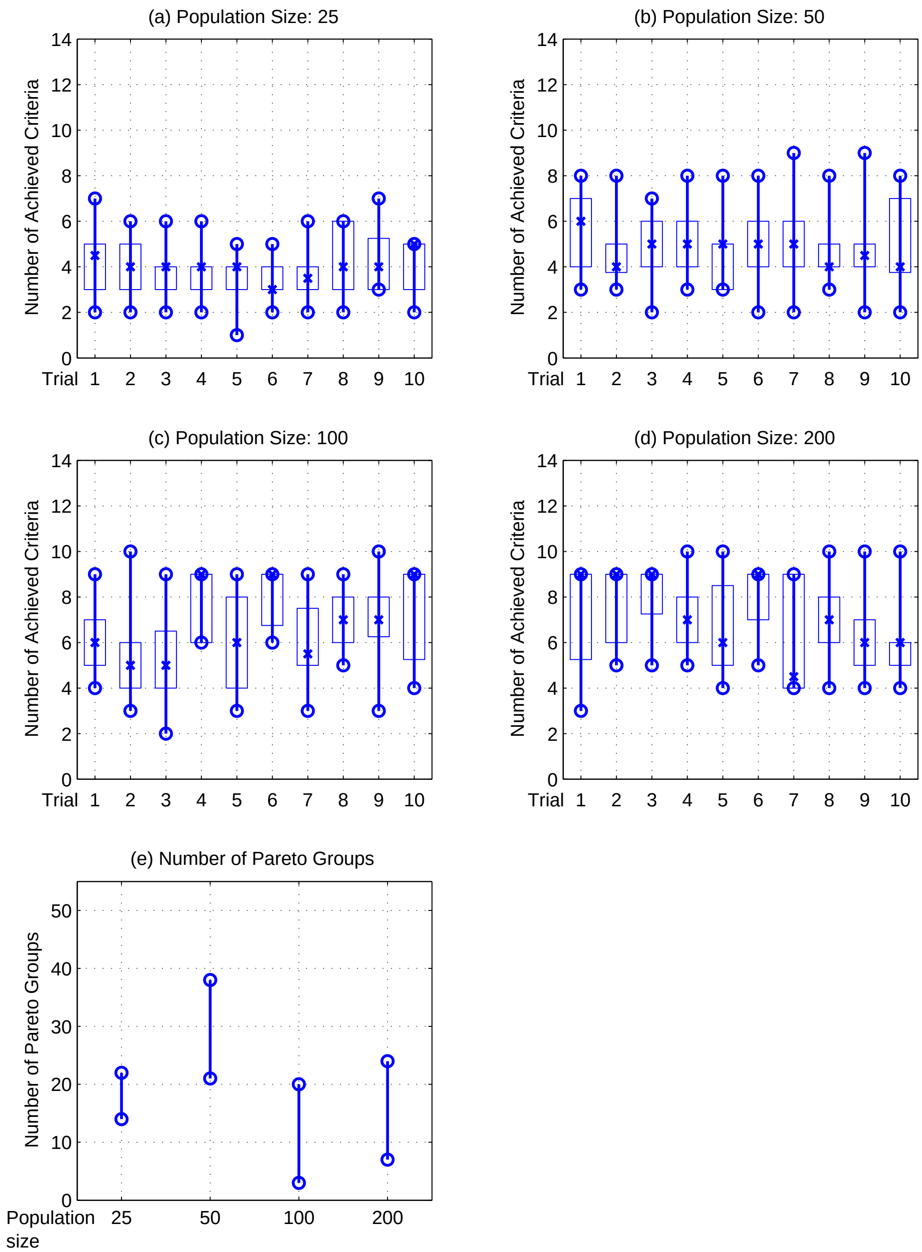


Figure 5.15: Change of the Pareto frontier for five different sets of a population size and the maximum generation number 500. (a)-(d) Box and whisker plots of the number of achieved criteria by a Pareto group for each of the ten different trials. Each box represents a range between the first and third quartiles. Symbol "o"'s shows the smallest and largest values, and "x"'s represent the median. (e) Range of the number of distinct Pareto groups by a trial.

size 100. This is because Pareto groups with more achieved criteria were obtained when the population size was 100, which made the total number decreased. For population size 200, the range did not became wider or larger although the population size was doubled. From these results, population size 100 is appropriate enough with the maximum generation 500.

The maximum number of achieved criteria by a Pareto group was as good as that with a population size of 100 (Figure 5.14(c)). The maximum and minimum numbers of Pareto groups were twenty-four and seven, respectively, among the ten trials, and with a population size of 100, twenty and three, respectively. The Pareto frontier of each of the ten trials varies, but, generally, I obtained more Pareto groups with a population size of 200 than those with a population size of 100. Although it is likely to obtain more Pareto groups on the Pareto frontier with a population size of 200, it is much less than double of those with a population size of 100. Thus, I use population size 100 in the next research problem.

Chapter 6

INFLUENCE OF CROSSOVER AND MUTATION ON A SEARCH

Introduction of elitism prevented reduction of Pareto groups maximizing the number of criteria achieved. However, the research is not complete because the maximum number of criteria in one member is ten, whereas there are fourteen criteria, and the fitting was not as good as that with the single-objective methods. The crossover/mutation probability I used so far is linear with generation number, and mutation probability becomes higher as the generation number increases. I considered that this probability could affect the result. In order to see the role of crossover and mutation, I ran Pareto_Evolve with crossover only and mutation only.

Up to this point, the probability of crossover and mutation decreased and increased linearly with the generation number, respectively. The idea behind this method was that crossover would produce wide exploration of the search space, and mutation would produce a gradual refinement of parameterizations selected as parents. I considered that the crossover and mutation probability should be made dynamic on the progress toward solutions, i.e., upon the development of the Pareto frontier. For example, if a parent selected to be bred was already chosen as an elite, it is probably more efficient to apply crossover rather than mutation to it. The probability of applying crossover also has to be designed so that an offspring locates its objective values where it is not close to those of parents in objective space if it reaches a locally optimal solution. Generally, it is difficult to judge where a local optimum is; however, if there is the field in the parameter search space where many individuals are clustered, it may be a local or global optimum. These clustered individuals should be given a high probability to be crossed over. If it is a global optimum, it is very likely that not all of the individuals die out because elitism is designed so that good genes can survive. On the other hand, if the clustered individuals are locally optimal, individuals

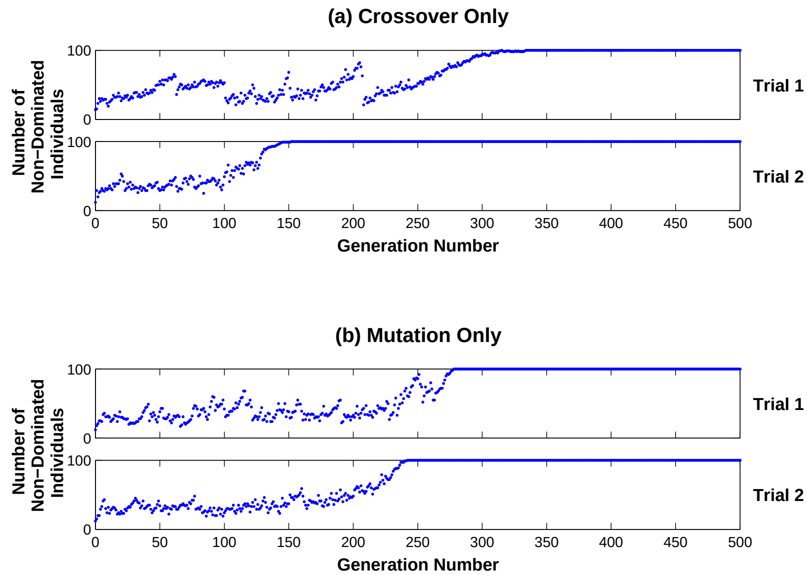


Figure 6.1: The number of non-dominated individuals by searches (a) crossover only and (b) mutation only. Population size is 100. The results of two trials among ten are presented for both cases.

whose assessment vectors are better than theirs may appear; they will not be elites, so they will be crossed over, and eventually, the clustered point should disappear because crossover allows individuals to jump to a different area from the original one.

Before modifying the crossover/mutation probability, I will illustrate the effect of the differences between crossover and mutation on the search process. I tried ten trials with population size 100 and generation number 500 for each of two experiments; search with crossover only and mutation only. The numbers of non-dominated individuals, elites and the maximum and minimum achieved criteria by a Pareto group for two representative trials among ten for both experiments are plotted in Figure 6.1, 6.2 and 6.3, respectively. The numbers of elites reached the population size (Figure 6.2), and around the same time, the numbers of non-dominated individuals did, too (Figure 6.1). This happened probably because the current elitism selects too many elites as I mentioned in Section 5.1.2.

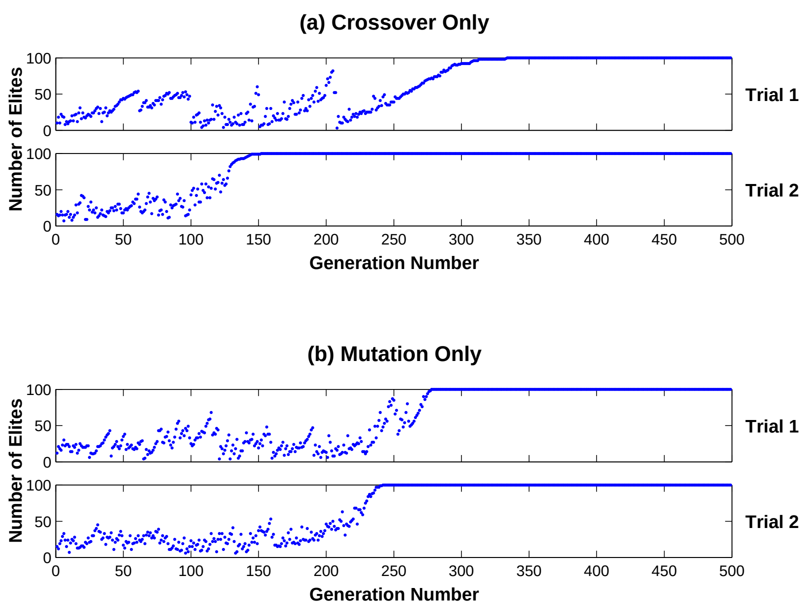


Figure 6.2: The number of elites by searches (a) crossover only and (b) mutation only. Population size is 100. The results of two trials among ten are presented for both cases. (a) The generation when the number of elites stabilize is quite different from trial by trial with crossover only (generation 150–320); the earliest and latest are shown above. Crossover exchanges some of parameter values between two parents, so the created offspring are not so close to their parents in the search space. The generation when the number reaches the population size varies depending on the "jump" in the search space. (b) The generation when the number of elites stabilize varies much less for mutation only (generation 240–275). Since mutation changes some of parameter values of a parent by small amounts, the parent and its created offspring are close to each other in search space. Hence, the solutions are very likely to converge to the field in search space where there exist many solutions. This is probably the reason why the rate of convergence varies much less than that for crossover.

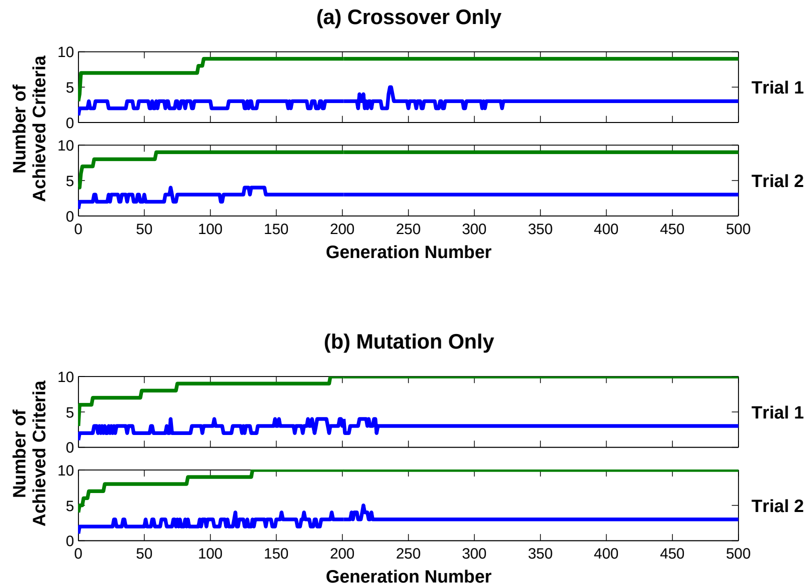
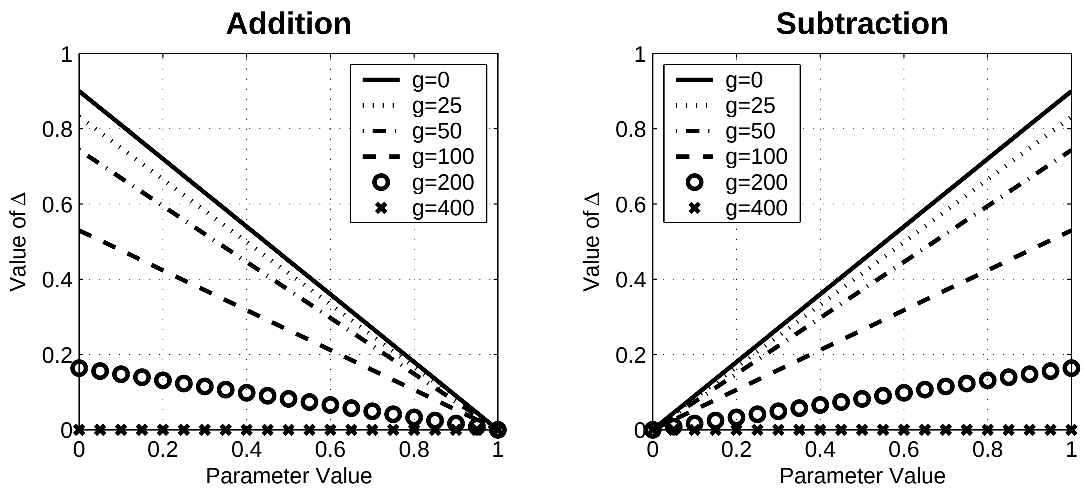


Figure 6.3: The numbers of maximum and minimum achieved criteria by a Pareto group by searches (a) crossover only and (b) mutation only. The maximum numbers are as large as the case shown in Section 5.1.2 (Figure 5.9(c)).

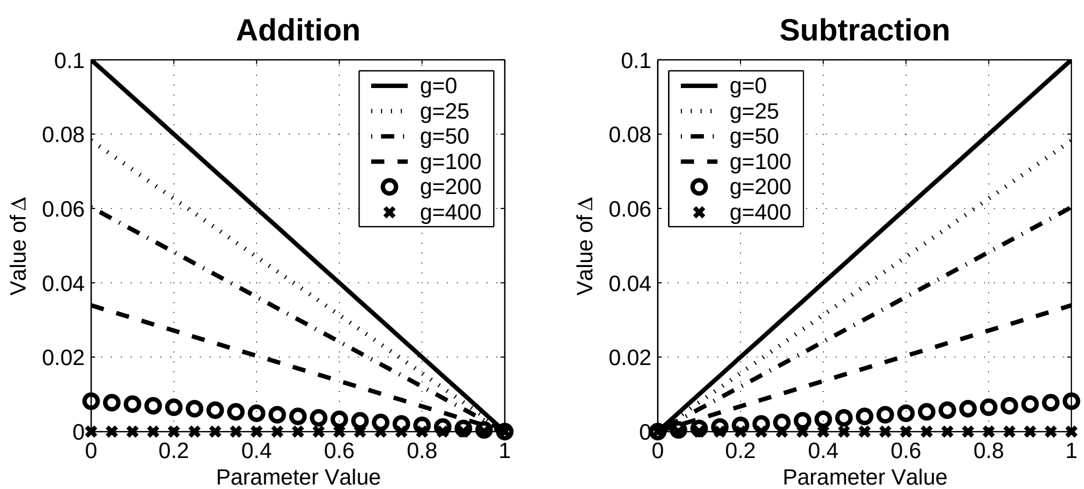
The first thing I should mention is the difference in generation number where the number of elites reached the population size. With crossover only, the generation where that happened differed considerably from trial to trial (generation 150–320), but varied much less for mutation only (generation 240–275).

As introduced in Chapter 2, crossover exchanges at least one parameter value, out of seven, but not all, between two parents. Thus, an offspring produced by crossover usually has a objective vector which is not so close to that for the parents. Thus, we can expect that even though the parents are dominated, offspring can be non-dominated. I use nonuniform mutation used in Pareto_Evolve, and it is defined as follows:

> For each parameter x_k of an individual selected for mutation, randomly determine whether the value is changed or not. If it is to be changed, x_k is changed



(a) $q = 0.1$



(b) $q = 0.9$

Figure 6.4: Function value of step size Δ for mutation with $[LB, UB] = [0, 1]$, $g_{\max} = 500$, $\beta = 5$ random number (a) $q = 0.1$ and (b) $q = 0.9$. The left plots are for addition and the right one is for subtraction. The plots are at six different generations; g is 0, 25, 50, 100, 200 and 400. The slope become smaller exponentially as generation increases. For other value of $q \in [0, 1]$, we obtain a different scale for Δ, but a similar shape. The value of Δ becomes smaller as the generation number increases, which shows that mutation in the Pareto_Evolve produces gradual refinement of parameterizations for selected parents.

to

$$x'_k = \begin{cases} x_k + \Delta(g, UB - x_k) & \text{if addition} \\ x_k - \Delta(g, x_k - LB) & \text{if subtraction} \end{cases}$$

with the lower bound LB and upper bound UB of the search range for parameter x_k, and function Δ:

$$\Delta(g, y) = y \cdot \left(1 - q^{(1-g/g_{\max})^{\beta}}\right),$$

where q is randomly chosen from $[0, 1]$, $g_{\max}$ is the maximum generation number, and β is a system parameter determined by the user. Whether addition or subtraction is applied, is determined randomly; the probability is fifty-fifty.

For the original Pareto_Evolve, $\beta = 5$ based on Michalewicz [28]. Currently, $LB = 0$ for all seven parameters and UB is 0.4 for x_1, 0.1 for x_2, x_3, x_4 and 1.0 for x_5, x_6, x_7. By definition, the step size defined by function Δ gets closer to 0 as $g \to g_{\max}$ generation number so that mutation searched more locally (Figure 6.4).

I explained the difference of the role for crossover and mutation. We can see it from Figure 6.5 and 6.6. Before the search stops, that is, the number of elites reached to the population size, values of each parameter are scattered for the case of crossover only because offspring produced by crossover are generally not close to their parents in search space. On the other hand, the parameterizations for mutation-only case are close to each other.

Figures 6.1 and 6.2 show the different behaviors of change of non-dominated individuals and elites by search with crossover only and mutation only. For both cases, the number of non-dominated individuals and elites sometimes decreases during the search. I will show next why this happened.

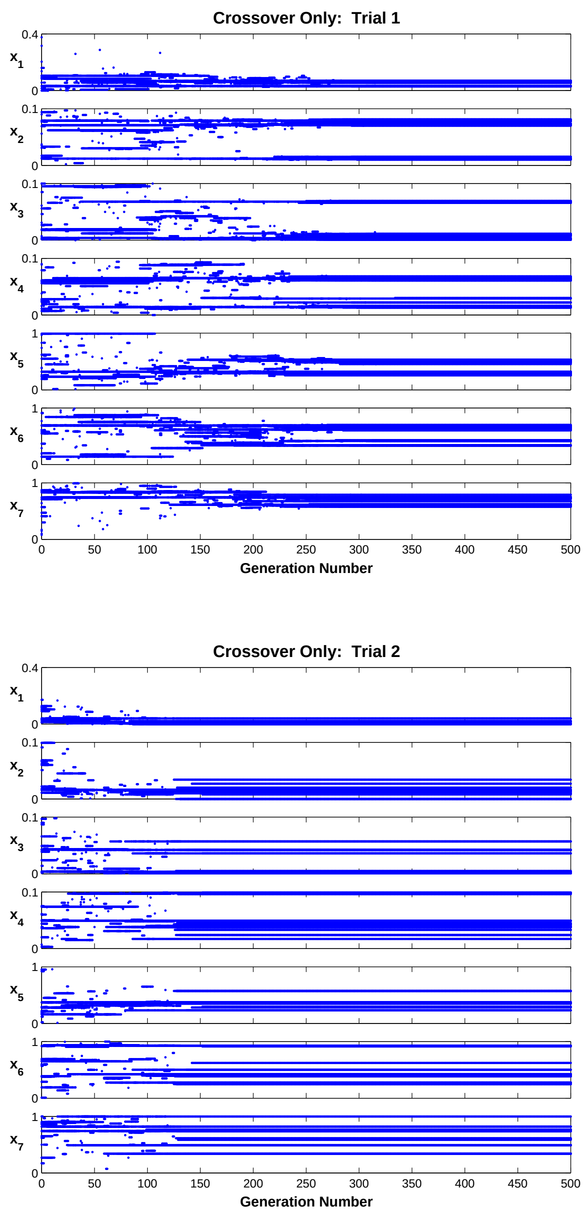


Figure 6.5: Parameter values of all non-dominated individuals found, by generation, for (a) trial 1 and (b) trial 2 by searches with crossover only. Population size is 100. Parameter values for trial 2 stabilized much earlier than those for trial 1. The difference comes from the generation when the number of elites reached the population size.

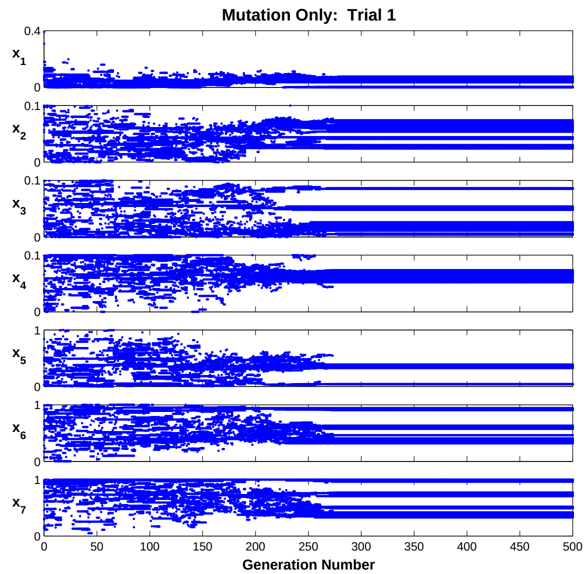


Figure 6.6: Parameter values of all non-dominated individuals found, by generation, for a search with mutation only. Population size is 100.

6.1 Search by Crossover Only

For trial 1 by the search with crossover only, the number of non-dominated individuals suddenly decreased around generation 210, and then increased until the search ended (Figure 6.1(a)). In this section, I investigate how the number of non-dominated individuals and elites changed around generation 210.

Recall the three conditions for elitism. The assessment vector of an elite has to

1. achieve the most criteria in the Pareto frontier of the current generation or

2. be better at least one of the Pareto groups of the previous generation or

3. achieve a criterion unachieved by any Pareto groups which were already chosen under above conditions 1 and 2.

Table 6.1: The number $\tilde{n}_g$ of non-dominated individuals and the number e_g of elites at generation $g = 205, \ldots, 209$ for trial 1 by the search with crossover only. Population size is 100. The number of elites under each of three conditions is also presented. We see that condition 3 is the main contributor to the total number, apart from at generation 208. The number $\tilde{n}_g$ of non-dominated individuals decreases after generation 205, and starts to increase again from generation 209.

Generation number	205	206	207	208	209
Non-dominated individuals $\tilde{n}_g$	82	76	63	21	29
Elites e_g	82	52	52	3	19
Condition 1 (max number of criteria)	1	1	1	1	1
Condition 2 (domination)	0	3	2	2	3
Condition 3 (unachieved criteria)	81	48	49	0	15

The number $\tilde{n}_g$ of non-dominated individuals and the number e_g of elites under each of the three conditions are shown in Table 6.1 for generation $g = 205, \ldots, 209$ of trial 1 (Figure 6.1(a)). The number $\tilde{n}_g$ of non-dominated individuals decreased from generation 206, and started to increase again from generation 209.

At generation 205, only Pareto group P_1^{205} was selected under the first condition for elitism, and it had only one individual (Figure 6.7). There did not exist any elites under the second condition. Since P_1^{205} unachieved criterion 1, 2, 5, 7 and 12, condition 3 required all the other Pareto groups to be selected for elitism because they achieved at least one of those five unachieved criteria. Thus, all 82 non-dominated individuals became elites.

At generation 206, more than 80% of the population members were copied as elites from the previous generation; thus I may expect that the population did not change much from

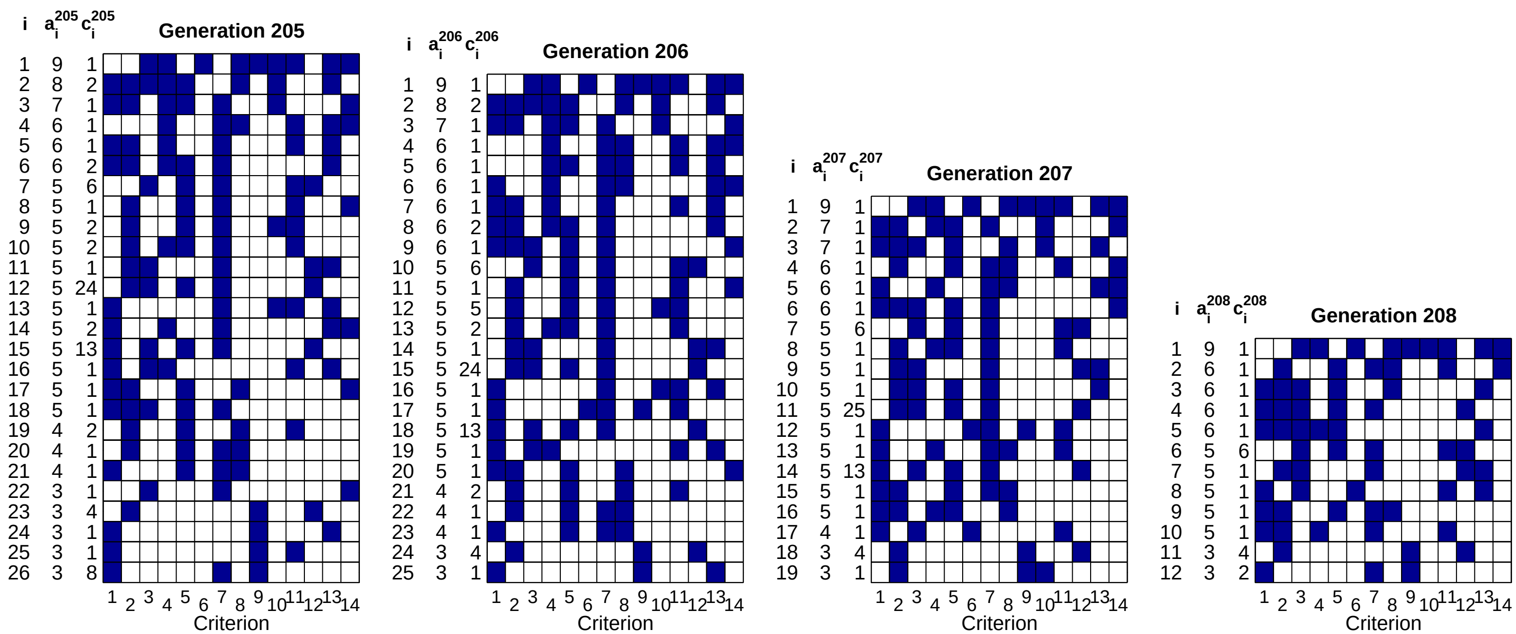


Figure 6.7: Pareto frontiers from generation 205 through 208 of trial 1, the search with crossover only. Population size is 100. Each plot corresponds to Pareto frontier $\{P_i^g | i = 1, 2, \ldots\}$ for generation $g =$205, 206, 207, 208. Black and white squares represent achieved and unachieved, respectively, for criterion 1 through 14. E.g., Pareto group P_1^{205}, which is the group plotted at the top row of generation 205, achieved nine criteria ($a_1^{205} = 9$): 3, 4, 6, 8, 9, 10, 11, 13 and 14. There was $c_1^{205} = 1$ individual (parameterization) in this group.

that at the previous generation. However, the Pareto frontier was improved by three Pareto groups. They are better than some Pareto groups of the previous generation: $P_{17}^{206} \succ_p P_{25}^{205}, P_{26}^{205}$, $P_6^{206} \succ_p P_{14}^{205}$ and $P_{24}^{206} \succ_p P_{18}^{205}, P_{22}^{205}$. The number of individuals copied from the previous generation is $\{(c_{25}^{205} + c_{26}^{205}) + c_{14}^{205} + (c_{18}^{205} + c_{22}^{205})\} = 13$, and those thirteen individuals were dominated by individuals with the three Pareto groups.

The three new Pareto groups were created by crossover. Assessment vectors of offspring created by mutation are generally the same as those of parents due to general continuity of the objective functions and the use of binary error ranges. In the original Pareto_Evolve, it is preferred that crossover probability decreases and mutation probability increases as the generation number becomes large. However, this may not be appropriate for this problem. Even though the generation number is large, a higher crossover probability can be considered more efficient after many elites are selected at the previous generation since elites are just copied to the next generation under the elitism I am using.

Each of the three new Pareto groups P_i^{206} $(i = 6, 9, 17)$ had only one corresponding individual. In addition, I obtained only four new non-dominated individuals; one belonged to a new Pareto group, P_5^{206}, and the other three belonged to $P_{12}^{206} = P_9^{205}$. Those three were copied as elites.

At this point, the numbers of individuals are

Copied as elites from generation 205		82
Copied as elites but dominated under condition 2		13
New		$4 + 3 = 7$
Non-dominated at generation	205	82
	206	$82 - 13 + 7 = 76$

At generation 207, only 50% of the population members were copied as elites. This allowed new individuals to join the population. There were eight new Pareto groups P_i^{207} $(i = 3, 4, 10, 13, 15, 16, 17, 19)$, each with one individual. P_4^{207} and P_{15}^{207} satisfied condition 2; $P_4^{207} \succ_p \{P_6^{206}, P_7^{206}, P_9^{206}\}$ and $P_4^{207} \succ_p P_9^{206}$. Those three Pareto groups were not selected for elitism at generation 206.

Pareto groups P_2^{207} and P_8^{207} each had one individual, and they appeared as P_3^{206} and P_{13}^{206}, respectively. However, since these corresponding individuals were not selected as elites, each parameterization for P_2^{207} and P_8^{207} was different from those for P_3^{206} and P_{13}^{206}. Also, one new individual belonging to P_{11}^{207} joined the population.

At this point, the numbers of individuals are:

Copied as elites from generation 206		52
New		$8 + 2 + 1 = 11$
Non-dominated at generation	206	76
	207	$52 + 11 = 63$

At generation 208, as for the previous generation, about half of the population was copied as elites. Since Pareto groups selected under condition 1 and 2 satisfied all criteria, there was no Pareto group selected under condition 3, making the number of elites only three. New Pareto group P_4^{208} with an individual selected under condition 2 is better than P_{11}^{207} and P_{14}^{207}, which had 25 and 13 individuals, respectively, decreasing the number of non-dominated individuals by thirty-eight. Pareto group P_8^{208}, selected for elitism under condition 2, is better than P_{17}^{207}, but P_{17}^{207} was not selected for elitism at generation 207, thus the number of individuals corresponding to P_{17}^{207} did not affect to the calculation of the number of non-dominated individuals shown below. Only seven parameterizations newly joined to the population with new Pareto groups.

At this point, the numbers of individuals are:

Copied as elites from generation 207		52
Copied as elites but dominated under condition 2		38
New		7
Non-dominated at generation	207	63
	208	$52 - 38 + 7 = 21$

Pareto group P_4^{208} helped the search escape from a local optimum by being better than two Pareto groups which had many individuals that had been copied as elites from the previous generation. However, the corresponding individuals of P_{11}^{208} are not guaranteed to survive after the next generation because P_4^{208} may not be selected for elitism under the current definition. If it is not selected as an elite, it could die out; disappear or change to a completely different parameterization by crossover. Therefore, if elitism is modified so that these small number of individuals can survive longer, the Pareto group can remain until a better one appears or some further convergence takes place. The results by this modified elitism is shown in Section 8.3.

6.2 Search by Mutation Only

Figure 6.1(b) shows that the number of non-dominated individuals for trial 1 by the search with mutation only increased and then decreased around generation 250. This happened because a Pareto group having a lot of individuals within the Pareto frontier stopped being selected as elites. I explain why next.

The number $\tilde{n}_g$ of non-dominated individuals at generation g and the total number e_g of elites selected using the three conditions for determining elites for generation $g = 251, \ldots, 255$ of trial 1 (Figure 6.1(b) and 6.2(b)) are presented in Table 6.2. As for the crossover-only search, e_g depends mainly on the third condition; e_g decreased by about half from generation 252 to 253 because the number of elites selected under condition 3 decreased by half. This is similar to the process described in Section 6.1, at generation 208 for trial 1 in the search with crossover only. Many individuals with Pareto groups from the previous generation were dominated by only one individual with a new Pareto group.

Figure 6.8, where Pareto groups are plotted for generation g = 251, 252, 253, 254, shows that Pareto group $P_9^{251} = P_9^{252} = P_6^{253} = P_8^{254}$ increased in size (number of corresponding individuals) one by one through generation 253, but lost most of them at generation 254; it changed $34 \rightarrow 35 \rightarrow 36 \rightarrow 5$. Actually, it lost all of them, and new individuals were created

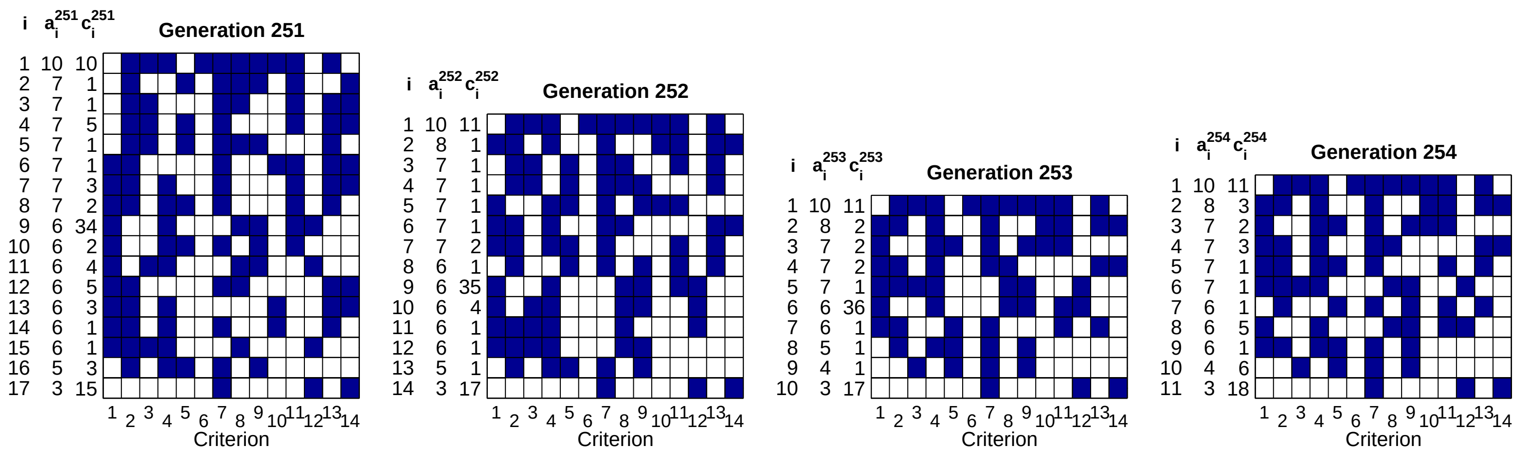


Figure 6.8: Pareto fontiers $\{P_i^g | i = 1, 2, \ldots\}$ from generation 251 through 254 of trial 1, the search with mutation only. Population size is 100. Three numbers i, a_i^g, c_i^g presented on the left of each plot are index of Pareto group P_i^g, number of achieved criteria by P_i^g, number of corresponding individuals to P_i^g. Black and white squares represent achieved and unachieved, respectively.

Table 6.2: The number $\tilde{n}_g$ of non-dominated individuals and the number e_g of elites at generation $g = 251, \ldots, 255$ for trial 1 by the search with mutation only. Population size is 100. The number of elites under each of three conditions is also presented. The number e_g of elites is driven by condition 3. The number $\tilde{n}_g$ of non-dominated individuals decreases after generation 251, increasing again from generation 255.

Generation number	251	251	253	254	255
Non-dominated individuals $\tilde{n}_g$	92	78	74	52	61
Elites e_g	66	77	38	43	59
Condition 1 (max number of criteria)	10	11	11	11	12
Condition 2 (domination)	2	3	1	2	1
Condition 3 (unachieved criteria)	54	57	26	30	46

as offspring. None of the individuals corresponding to P_6^{253} was copied because P_6^{253} did not satisfy any of the three conditions for elitism. However, since it had many individuals, some of them were probabilistically selected as parents. The resulting offspring had the same assessed vector as the parents because the current mutation changes parameter values only a little.

We saw that mutation with small function value of Δ defined in Section 2.4 brought an offspring which had the same assessed vector as its parent did. As I mentioned before, in the original Pareto_Evolve, as generation number increases, crossover probability becomes low and mutation probability becomes high. However, since my elitism just copies individuals to the next generation, an individual should be either crossed over or mutated with large value of Δ if it is selected not only as an elite but as a parent. Since the assessed vectors of the elites are copied to the next generation, it is undesirable to obtain the same assessed vector by evolution. I expect this can be avoided by crossover or mutation with large value of Δ because crossover makes an individual to jump out from its field, and mutation with large Δ changes its parameter values in quantity.

6.3 Conclusion about Crossover/Mutation Probability

In Section 2.4, I introduced crossover and mutation in Pareto_Evolve. For crossover, uniform crossover is used; the number of the crossover points, as well as crossover points themselves, are chosen randomly. For mutation, each parameter x_k is determined randomly whether the value is changed or not; if yes, x_k is changed to

$$x'_k = \begin{cases} x_k + \Delta(g, UB - x_k) & \text{if addition} \\ x_k - \Delta(g, x_k - LB) & \text{if subtraction} \end{cases}$$

with the lower bound LB and upper bound UB of the domain, and function Δ. Since $E(\Delta) \to 0$ as $g \to g_{\max}$ ($E(\Delta)$ denotes the expected value of Δ), the search becomes localized.

Sections 6.1 and 6.2 suggest some points that should be considered in deciding the probability of choosing genetic operator crossover/mutation.

- If many parents are crossed over at a late generation, their offspring may just keep jumping around in the search space.

- Elites are selected first, and then parents are chosen (Figure 5.3). Thus, if a selected parent has been already chosen as an elite, applying crossover or mutation with large value of Δ to it is more efficient because this produces offspring that are likely to have different assessment vectors from their parents.

- Since the probability of mutation (i.e., probability of selecting mutation operator) linearly increases with decreasing value of Δ, applying mutation to many individuals which are close to each other in parameter space and belong to the same assessment vector is inefficient. (This is for binary error measure, and for continuous measure case, it may help to obtain an offspring dominating the parent.) Since Δ becomes smaller and smaller as generation number increases, we can scarcely expect that at a late generation these individuals change their values in quantity or have different assessment vectors than their parents.

Therefore, I redefine the probability of taking crossover or mutation as follows.

- Parents to be bred which have been already selected as elites should have either high crossover probability or mutation probability with a large step size Δ so that they can produce offspring further away in parameter space.

6.4 Modification to Elitism

I have been trying to find how many criteria can be achieved among fourteen criteria. If it is determined, which criterion is hard to achieve with which criterion needs to be examined so that I can know the model is biased against some criteria. Therefore, Pareto groups achieving the criterion which is hard to achieve also should be kept. Thus, I need to define the elitism not only to keep Pareto groups with many achieved criteria but also not to ignore Pareto groups achieving criteria which is difficult to achieve.

Sections 6.1 and 6.2 showed that the number of elites was very dependent on the third condition of my definition of elitism: the individual achieves a criterion unachieved by any Pareto groups which were chosen under the other two conditions. If the number of criteria unachieved by any Pareto groups selected under the first and second condition is large, as is likely in the early generations of a search, most Pareto groups could be chosen under the third condition, and then their non-dominated individuals become elites, which means they are copied to the next generation. If this happens, then the population members may not change much because most of the non-dominated individuals can be copied to the next generation. Therefore, the current elitism should be modified so that not many individuals selected under the third condition as elites.

Also, at the end of Section 6.1, I found that elitism should be changed so that if the Pareto group that achieves most criteria corresponds to only small number of individuals, these should survive until a Pareto group achieving more criteria appears. In next section, I will show the results of Pareto_Evolve using this modified elitism.

Chapter 7

EXPERIMENT THREE: ELITISM PRESERVING SUPERIOR INDIVIDUALS

Experiment Two, with crossover only, demonstrated a drawback in my definition of elitism (Section 6.1). A Pareto group at the current generation which is better than those at the previous generation (condition 2) is not guaranteed to survive to the next generation. If I obtain a new Pareto group which has only one individual and is better than a Pareto group at the previous generation, it will pass to the next generation by elitism. We assume that, at the next generation, it remains a Pareto group with only one individual. Then it may not satisfy any of the conditions of elitism because its corresponding individuals have been just copied. Since it has only one corresponding individual, the probability that the individual will be selected as a parent and produce an offspring having the same assessment vector is low.

To solve this problem, I modified the selection of elites in the following way. A Pareto group at the current generation which is better than one at the previous generation should survive until a better one appears. However, we do not need too many individuals to represent one Pareto group. In order to achieve this, I will set the maximum number of individuals copied as elites per Pareto group. They should be chosen widely from within those members. In order to establish how to select individuals within a Pareto group, a distance is introduced.

I also had a problem in the third condition for the selection of elites. In Experiment Two (Section 6.1), all Pareto groups were selected at generation 205 by elitism. Therefore, too many individuals can be copied to the next generation by elitism. To avoid this, after Pareto groups achieving the criterion unachieved by any Pareto groups selected under condition 1 and 2 are chosen, only ones achieving the most criteria among them will be selected; e.g., we assume criterion 1 is not achieved by any Pareto group of generation 50 but it is achieved by three Pareto groups P_1^{51}, P_2^{51} and P_3^{51} of generation 51; if P_1^{51}, P_2^{51}, P_3^{51} achieves 4, 7, 10 criteria, respectively, then only P_3^{51} is considered to meet the condition. Since fewer Pareto groups tend to be chosen by this method than by the previous one, less individuals are copied to the next generation.

7.1 Definition of the New Procedure for Selecting Elites

If a Pareto group satisfies one of the following four conditions, some of its corresponding individuals are defined as elites:

1. It achieves the most criteria in the Pareto frontier of the current generation.

2. It is better than at least one of the Pareto groups of the previous generation.

3. It belongs to the *external pool* for the Pareto groups selected under condition 2 before the current generation; the external pool is to let an elite survive until a better one appears. All elites selected under condition 2 are stored in the external pool at each generation. They are compared with the Pareto frontier at the beginning of the next generation, and if there are better ones, then those are removed.

4. For the case where a criterion remains unachieved by any of the Pareto groups chosen under conditions 1, 2 and 3, and if there exist Pareto groups satisfying this criterion, the one that achieves the most criteria is chosen among them. For instance, we assume only P_1^{208} was selected for elitism under condition 1, 2 and 3 in Figure 6.7 in Section 6.1; it does not achieve criterion 1, 2, 5, 7 or 12; seven Pareto groups P_i^{208} $(i = 3, 4, 5, 8, 9, 10, 12)$ achieve criterion 1; while P_i^{208} $(i = 3, 4, 5)$ achieves six criteria, P_i^{208} $(i = 8, 9, 10, 12)$ achieves five; thus for criterion one, only P_i^{208} $(i = 3, 4, 5)$ are accepted under this condition.

While the Historical Pareto frontier is to keep the non-dominated individuals up to and including the previous generations, the external pool in this elitism is for Pareto groups with small numbers of corresponding individuals. Since we do not need many individuals that are numerically close to each other and have the same assessment vectors, I limited the number of corresponding to each Pareto group selected under one of the above four conditions by

$$l = \{\text{population size}\} \;/\; \{\text{number of Pareto groups in the current Pareto frontier}\}.$$

If the number of corresponding individuals to a Pareto group is smaller than or equal to the limit l, all of the individuals are accepted. Otherwise, only l individuals depending on crowdedness, which will be explained later, are chosen. Since I obtained many Pareto groups (Figure 7.7), l did not become too large.

Elites are created only when generation $2 \leq g < g_{\max}$, where g is the current generation and $g_{\max}$ is the maximum generation number. For Pareto frontier $\tilde{\mathrm{P}}_g$, set Q_g of external Pareto groups, set E_g of elites and set R_g of offspring, Pareto_Evolve is executed as follows:

Step 0 Set $g := 0$, randomly create the initial population P_g and initialize $\tilde{\mathrm{P}}_g$, Q_g and E_g; $\tilde{\mathrm{P}}_g := \emptyset$, $\mathrm{Q}_g := \emptyset$, $\mathrm{E}_g := \emptyset$.

Step 1 Select population, evaluate individuals, calculate assessment vector and calculate Pareto frontier $\tilde{\mathrm{P}}_g$.

Step 2 Update Q_g; delete all members in Q_g which are worse than any current Pareto groups in $\tilde{\mathrm{P}}_g$.

Step 3 Find the Pareto groups in $\tilde{\mathrm{P}}_g$ satisfying condition 1, and add them to E_g if they have not been added yet.

Step 4 Find the Pareto groups in $\tilde{\mathrm{P}}_g$ satisfying condition 2, and add them to Q_g.

Step 5 Add all of the members in Q_g to E_g if they have not been added yet. (condition 3)

Step 6 Find the Pareto groups in $\tilde{\mathrm{P}}_g$ satisfying condition 4, and add them to E_g if they have not been added yet.

Step 7 Calculate the crowding distance $d_i^{(g)}$ (explained later) for individual X_i^g corresponding to each Pareto group in E_g.

Step 8 Select l corresponding individuals as elites for each Pareto group in E_g. If there are l or fewer individuals, $d_i^{(g)} \leq l$, accept all of them as elites. Otherwise, l of X_i^g's with the first l largest $d_i^{(g)}$ are selected. (See below for $d_i^{(g)}$.)

Step 9 Selects $\nu = N - \mu$ ($|\mathrm{E}_g| = \mu$) of parents from population P_g, apply a crossover or mutation operator to produce ν of offspring for R_g.

Step 10 Set $\mathrm{P}_{g+1} := \mathrm{E}_g \cup \mathrm{R}_g$, $\mathrm{Q}_{g+1} := \mathrm{Q}_g$, $\mathrm{E}_{g+1} := \emptyset$, $\tilde{\mathrm{P}}_{g+1} := \emptyset$ and $g := g + 1$; then go back to *Step 1* if $g < g_{\max}$. Stop otherwise.

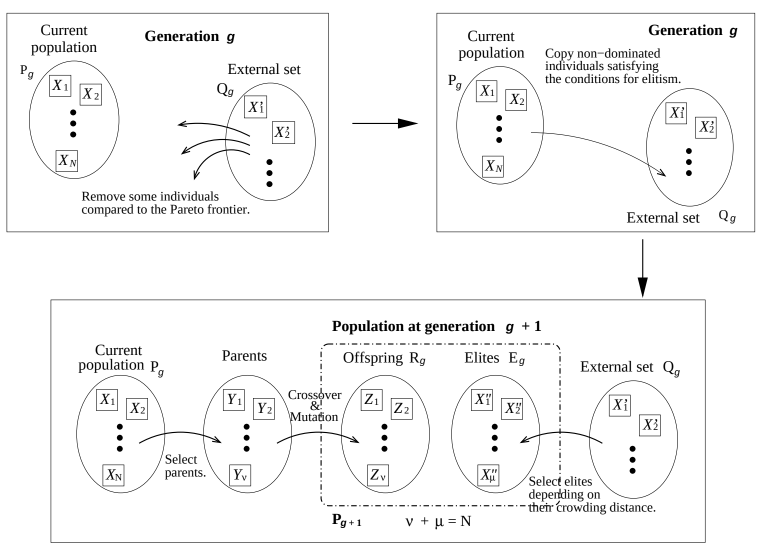


Figure 7.1: Elitism for Experiment Three. First, the non-dominated individuals satisfying the four conditions for elitism are selected from the current population P_g and are stored in external set Q_g (*Step 3* - *Step 6*). Crowding distance are calculated for individuals in Q_g (*Step 7*), and depending on the distance, some individuals are selected as elites from Q_g (*Step 8*). The number $\nu = N - \mu$ of parents are chosen from the current population P_g, and ν of offspring are produced (*Step 9*). The selected elites and produced offspring become the population of the next generation $g + 1$ (*Step 10*).

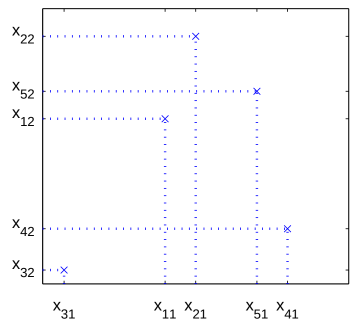


Figure 7.2: An example to calculate the crowding distance.

After calculating the new Pareto frontier $\tilde{\mathrm{P}}_g$, individuals of members in Q_g may not be non-dominated anymore. That is why Q_g has to be updated in *Step 2*. The selection of individuals in *Step 8* is done depending on "crowding distance" calculated in *Step 7*, which is introduced in Deb et al. [10] to select the exact number of elites for the method NSGA-II. Figure 7.1 shows how this elitism (*Step 3* - *Step 10*) works.

I now explain how crowding distance $d_i^{(g)}$ is calculated. It is the measurement for crowdedness around each individual in the parameter space; if the value is large, it is not crowded around the individual, and it is crowded if the value is small. We assume that there are ρ corresponding individuals $X_i^g = (x_{i1}, x_{i2}, \ldots, x_{im})$ $(i = 1, 2, \ldots, \rho)$ for a Pareto group in $\tilde{\mathrm{P}}_g$. In my case, $m = 7$ since there are seven different parameters. I rank those individuals depending on their crowding distance $d_i^{(g)}$.

Step 0 Set $j = 1$, and let $d_i^{(g)} = 0$ for $i = 1, 2, \ldots, \rho$.

Step 1 Sort x_{ij} $(i = 1, 2, \ldots, \rho)$ in ascending order, and let the sorted index i_k for $k = 1, 2, \ldots, \rho$.

Step 2 Calculate crowding distance:

$$d_{i_1}^{(g)} = d_{i_n}^{(g)} = \infty$$

$$d_{i_k}^{(g)} = d_{i_k}^{(g)} + \frac{x_{i_{k+1}j} - x_{i_{k-1}j}}{x_{[j]}^{\max} - x_{[j]}^{\min}} \qquad \text{for } k = 2, \ldots, \rho - 1.$$

Step 3 Set $j := j + 1$. If $j \leq m$, go back to *Step 1*. Stop otherwise.

In *Step 2*, $x_{[j]}^{\max}$ and $x_{[j]}^{\min}$ are maximum and minimum values of the j-th parameters, respectively; thus, the normalized distance between x_{ij} and its two neighboring points in the j-th space ($j = 1, 2, \ldots, m$) of the search space is calculated, and it is added to $d_i^{(g)}$ ($i = 1, 2, \ldots, \rho$). I show a simple example of calculation of crowding distance. As shown in Figure 7.2, there are five points in 2-dimensional plane ($m = 2$, $\rho = 5$), $X_i = (x_{i1}, x_{i2})$ for $i = 1, \ldots 5$, and I calculate the distance for $X_1 = (x_{11}, x_{12})$. First, x_{i1} and x_{i2} are sorted in ascending order, that is, $x_{31}, x_{11}, x_{21}, x_{51}, x_{41}$ and $x_{32}, x_{42}, x_{12}, x_{52}, x_{22}$, respectively (*Step 1*). From the figure,

$$
\begin{aligned}
x_{[1]}^{\min} &= x_{31} \\
x_{[1]}^{\max} &= x_{41} \\
x_{[2]}^{\min} &= x_{32} \\
x_{[2]}^{\max} &= x_{22},
\end{aligned}
$$

and two neighboring points of x_{11} and x_{12} are $\{x_{31}, x_{21}\}$ and $\{x_{42}, x_{52}\}$, respectively. Thus the crowding distance of $X_1 = (x_{11}, x_{12})$ is

$$d_1 = \frac{x_{21} - x_{31}}{x_{41} - x_{31}} + \frac{x_{52} - x_{42}}{x_{22} - x_{32}} \, .$$

In the same manner,

$$
\begin{aligned}
d_2 &= \infty, \\
d_3 &= \infty, \\
d_4 &= \infty, \\
d_5 &= \frac{x_{41} - x_{21}}{x_{41} - x_{31}} + \frac{x_{22} - x_{12}}{x_{22} - x_{32}} \, .
\end{aligned}
$$

Thus,

$$d_5 < d_1 < d_2 = d_3 = d_4 = \infty.$$

Since d_5 has the shortest crowding distance, (x_{51}, x_{52}) is the least crowding point among the five points.

Deb et al. [10] defined this crowding distance to rank each individual with respect to its crowdedness. In their *Step 2*, objective function values are used, while I am using parameter values. Since it is desirable to obtain parameterizations which are not close to each other in search space, I chose to use parameter values, instead of objective values.

7.2 Results of Experiment Three

In Experiment Three, I implemented Pareto_Evolve with elitism introduced in the previous section. As for Experiment Two, I executed ten trials with population size 100 and generation number 500. For all those trials, I obtained similar results; the number of elites were about the same as the number of non-dominated individuals, which means almost all non-dominated individuals were elites. Also, most of the population members became non-dominated. These phenomena were seen in Experiment Two. The results of two trials are shown in Figure 7.3.

The minimum number of achieved criteria became stable even if the number of non-dominated individuals and elites still increased (Figure 7.3(c)). This phenomenon was not seen in Experiment Two. I can infer that new Pareto groups achieving the small number of criteria were rarely obtained because of condition 4 of elitism. Thus, the minimum number of achieved criteria became stable before the number of non-dominated individuals and elites did.

Each parameter value over 500 generations for trial 1 is plotted in Figure 7.4. The result is very different from Experiment Two. The range of parameter values remained similar over 500 generations. This is because many or most individuals were copied to the next generation by elitism. The population does not change much. Therefore the obtained parameter values are considered to be dependent on the population at early generations.

Figure 7.5(a) shows the resulting simulated data. The smallest error of the simulated data

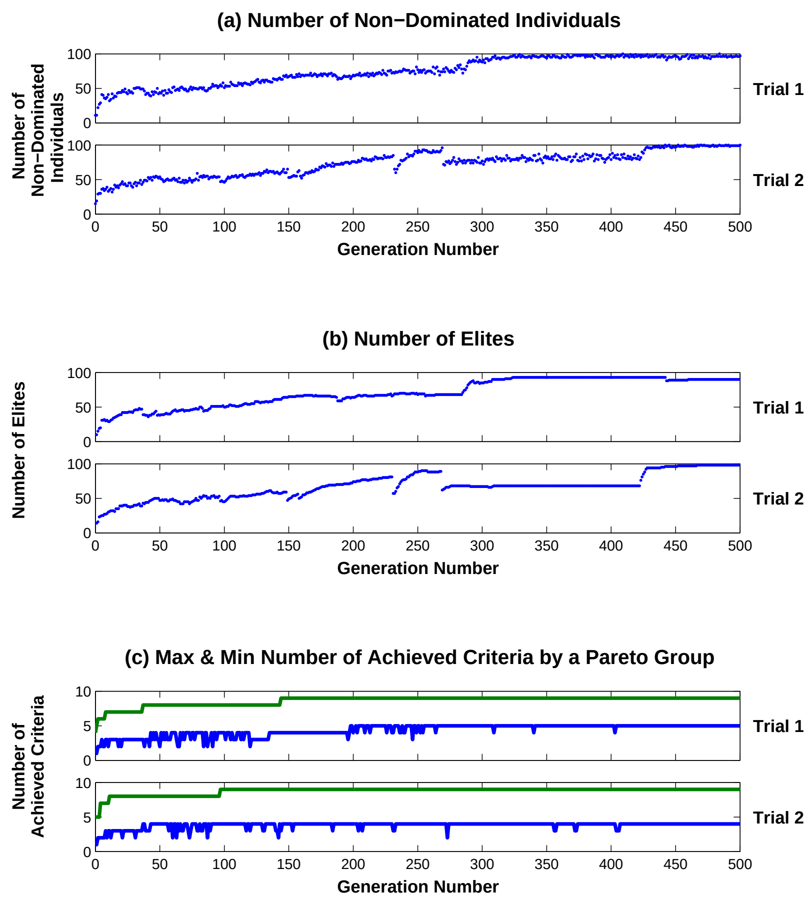


Figure 7.3: Results of two different trials in Experiment Three with a population size of 100. (a) Most of the population become non-dominated individuals before the maximum generation 500, but the time is different; around 300 for trial 1 and around 425 for trial 2. (b) Most of the non-dominated individuals are selected as elites around generation 300 for trial 1 and generation 425 for trial 2, so almost all of the population members are copied to the succeeding generation. This is same as the results of Experiment Two. (c) Even though the number of non-dominated individuals and elites still increase, the minimum number settles down, which is different from the results of Experiment Two. This is probably because new Pareto groups achieving the small number of criteria are rarely obtained because of condition 4 of elitism.

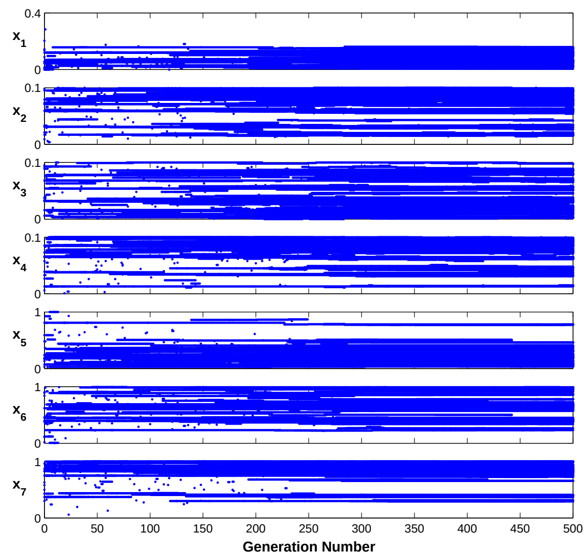


Figure 7.4: Parameter values of all non-dominated individuals found, by generation, for trial 1 in Experiment Three. Population size is 100. All of the parameter values stagnate before generation 300, around generation when the number of elites reached almost the population size. The range of each parameter value stays quite wide; it does not converge to a narrow range as for Experiment Two.

by a non-dominated individual is 15.663%, which is much better than that in Experiment Two (Section 5.1.2) but not as good as those by single-objective methods. The errors for the extension periods were as small as those by the Nelder-Mead simplex method except on the first and last days, but the errors for the contraction periods were fairly large compared to the results of the simplex method (Figure 7.5(b)).

The Pareto groups at generation 0, 100, 250 and 500 for trial 1 in Experiment Two and trial 1 in Experiment Three are shown in Figure 7.6 and 7.7, respectively. For Experiment Two, the number of elites completely reached the population size before generation 300; therefore the Pareto frontier did not change at all after that. For Experiment Three, the number of elites almost reached the population size, but not completely, so only one Pareto group joined to or disappeared from the Pareto frontier after the number of elites reached the maximum.

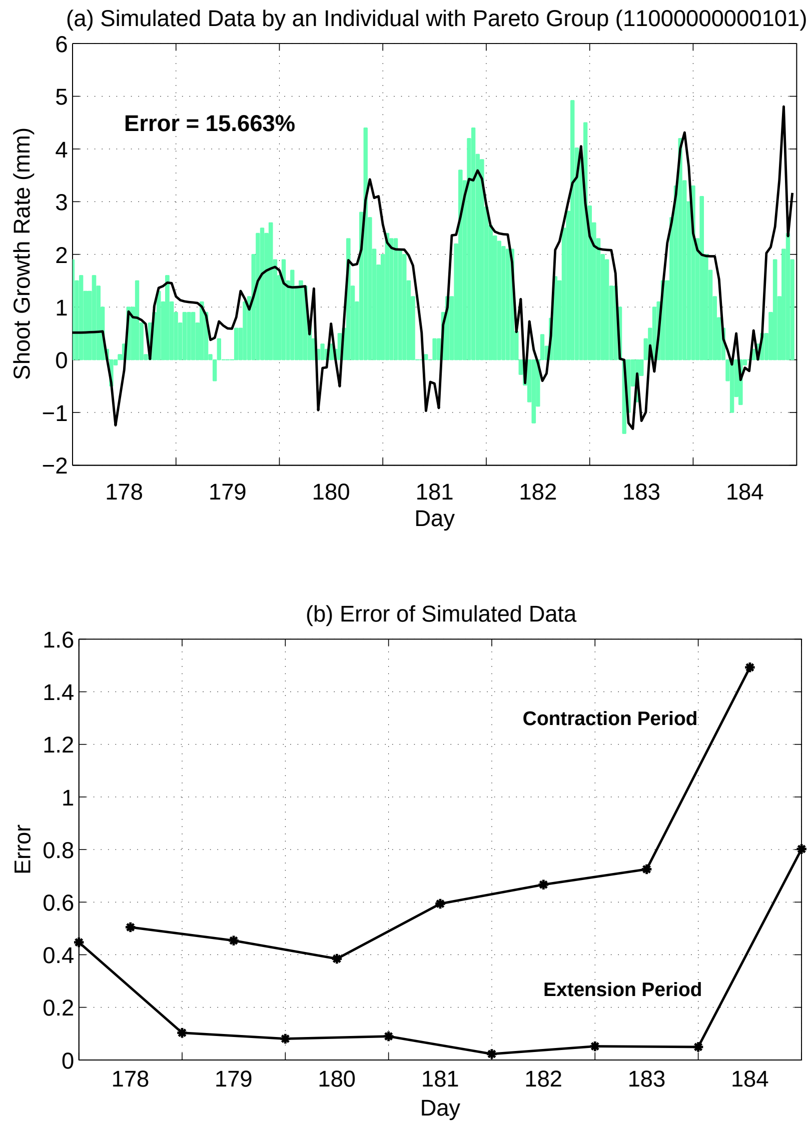


Figure 7.5: Results of Experiment Three. (a) The simulated data for the period of seven days, from Julian day 178 through 184 (solid line), and the measured data (bar). Population size is 100, and generation number is 500. Simulated data is by the individual corresponding to the Pareto group with the smallest error among all ten trials. (b) The ratio of the RSS between the measured and simulated data to the total sum of squares of the measured data for day 178-184. Errors for contraction periods (6:00-18:00) and extension periods (18:00-6:00+) are plotted separately. Extension periods on the first and last days are 0:00-6:00 and 18:00-0:00, respectively. Except those two, the errors are as small as those by the Nelder-Mead simplex method; however, the errors for the contraction periods are larger than those by the simplex method except on the last day (Figure 3.3(a)).

In Experiment Three, once a Pareto group was stored in the external pool, it was not removed as long as a better one appeared.; thus, all of the corresponding individuals were copied to the next generation as elites under condition 3. This is the reason why the number of parents to be crossed over was very small, which, made the slow. Therefore, the number of individuals preserved needs to be restricted for breeding (crossover and mutation). In next section, I will show how changed the elitism and the results of Pareto_Evolve with the modified elitism.

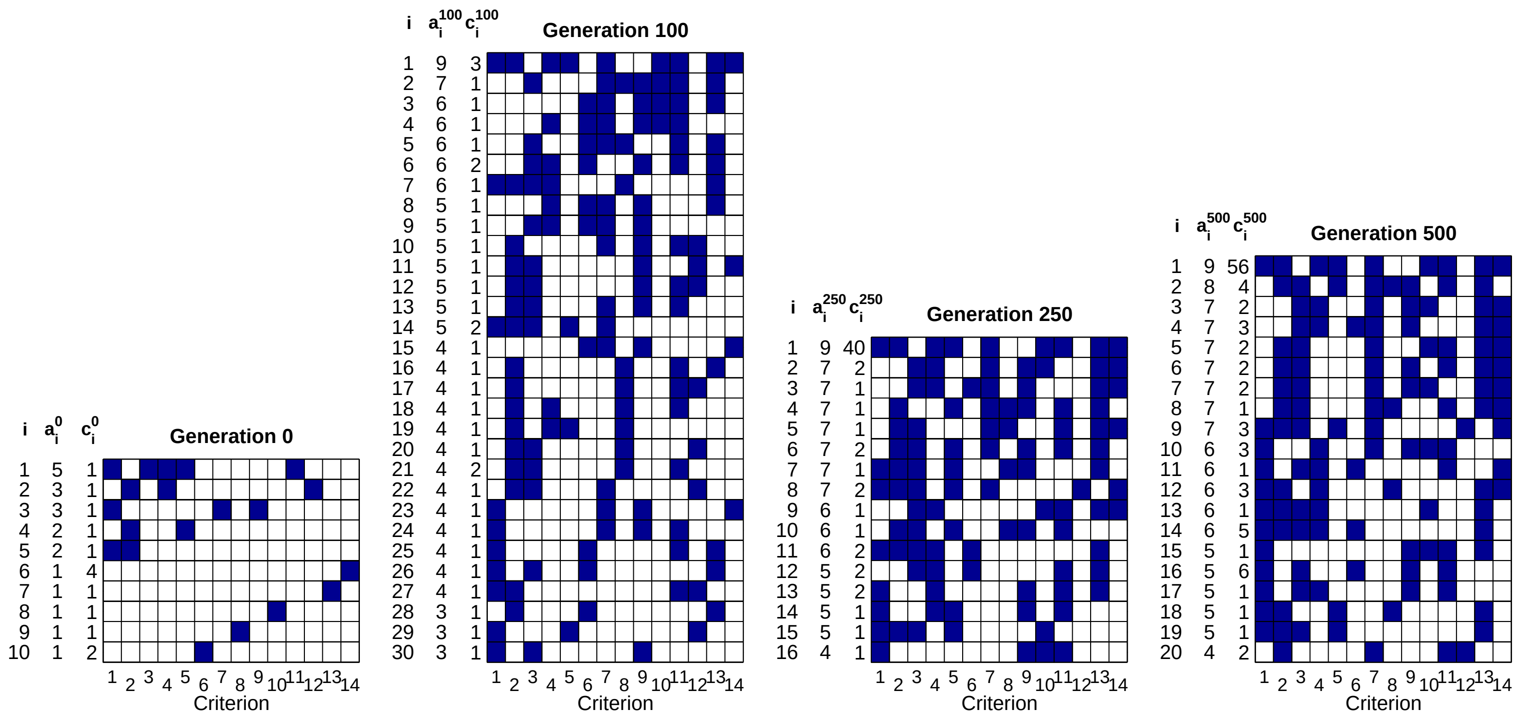


Figure 7.6: Pareto frontiers $\{P_i^g | i = 1, 2, \ldots\}$ at generation $g = 0$, 100, 250 and 500 for trial 1 by Experiment Two. Population size is 100. Black and white squares represent achieved and unachieved, respectively. Three numbers i, a_i^g, c_i^g presented on the left of each plot are index of Pareto group P_i^g, number of achieved criteria by P_i^g, number of corresponding individuals to P_i^g, respectively. The number of Pareto groups is not very large, but there exists one Pareto group which has a lot of corresponding individuals. The number of elites completely reached the population size before generation 300 (Figure 5.9(b)); after that, the Pareto frontier did not change at all.

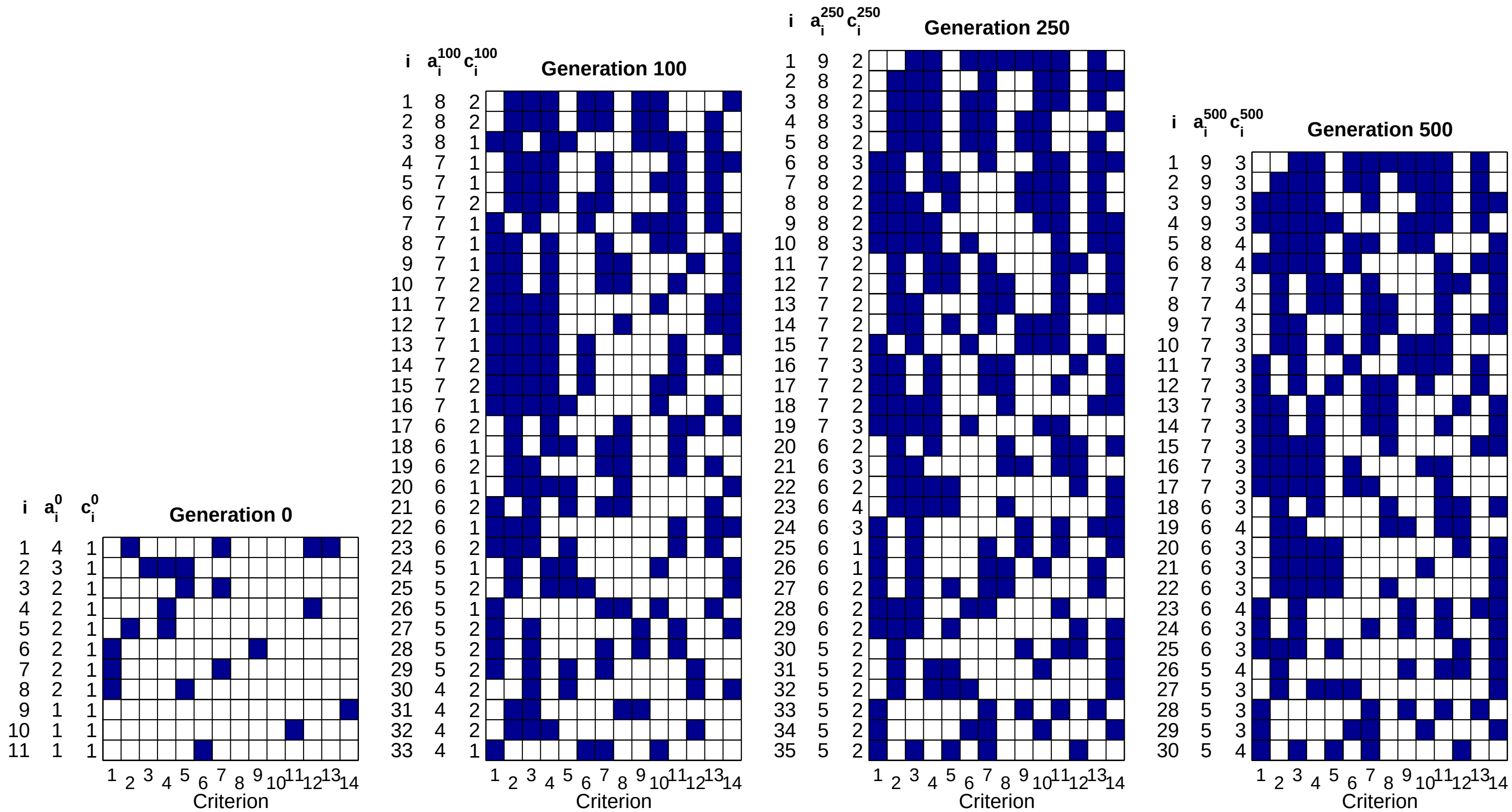


Figure 7.7: Pareto frontiers $\{P_i^g | i = 1, 2, \ldots\}$ at generation $g = 0$, 100, 250 and 500 for trial 1 by Experiment Three. Population size is 100. Black and white squares represent achieved and unachieved, respectively. Three numbers i, a_i^g, c_i^g presented on the left of each plot are index of Pareto group P_i^g, number of achieved criteria by P_i^g, number of corresponding individuals to P_i^g, respectively. Compared to the result by Experiment Two (Figure 7.6), much more variety of Pareto groups are obtained. However, after the number of elites reached the maximum, only one Pareto group joined to or disappeared from the Pareto frontier.

Chapter 8

IMPLEMENTING DYNAMIC CROSSOVER/MUTATION RATE WITH LIMITED DURATION ELITISM

8.1 Introduction

In this chapter, I introduce use of a dynamic crossover/mutation rate. The rate I experimented with previously was linear in relation to the generation number. As the generation number increased, the crossover probability linearly decreased from 2/3 to 0, the mutation probability linearly increased from 1/3 to 1. However, I considered that the search may become more effective if the probability is dynamic, depending on the obtained Pareto frontier.

8.2 Experiment Four: Simple Dynamic Rate

In Experiment Four, first, I tried a simple dynamic crossover/mutation rate with the elitism of Experiment Three (described in Section 6.3). If a parent has been already selected as an elite, it is inefficient to mutate it with small step size Δ defined in Section 2.4. Since mutation changes some parameter values by Δ, the created offspring stays numerically close to the parent in search space if Δ is small. Binary error measure is used for my problem, so it is efficient to apply crossover to the parent so that the search becomes broad; crossover usually makes an offspring jump far from its parent in search space. If a parent was not selected as an elite, it takes mutation. In Experiment Four, I used the same step size Δ for mutation as Experiment Three.

I tried ten times with population size 100 and generation number 500. The changes of the number of non-dominated individuals and elites are shown in Figure 8.1(a) and 8.1(b), respectively. The trials behave similarly, but differently from the trials in the experiment

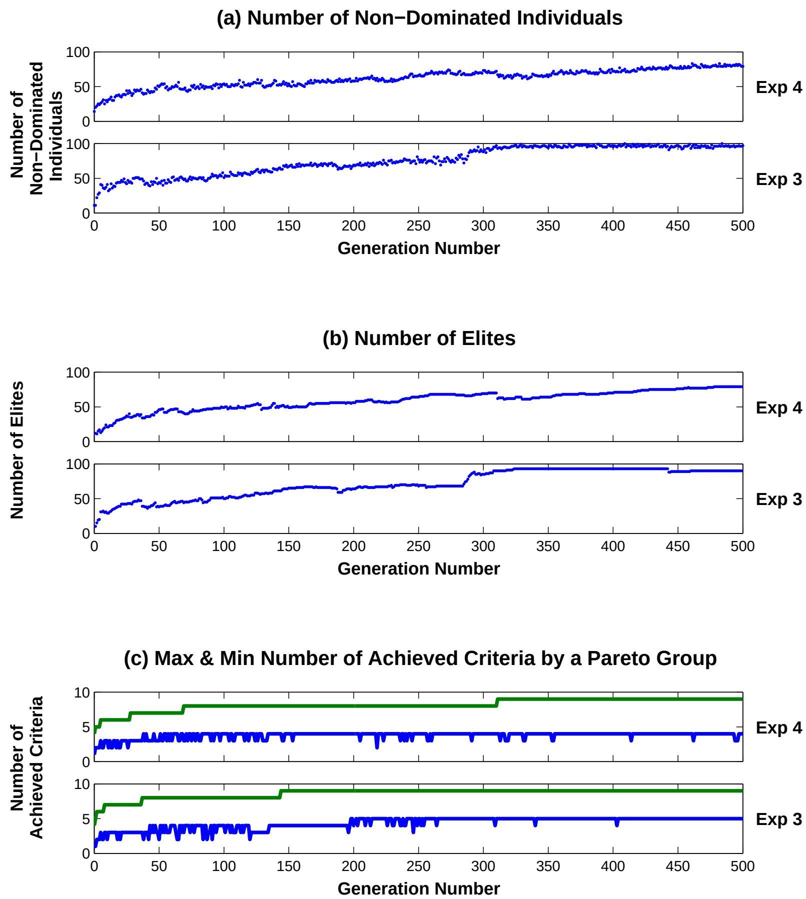


Figure 8.1: Results of a trial with the elitism as specified in Chapter 7 and the simple dynamic crossover/mutation (Experiment Four) and linear crossover/mutation (Trial 1 of Experiment Three in Figure 7.3). Population size is 100. (a) Differently from the result of Experiment Three (bottom), the number of non-dominated individuals of Experiment Four (top) does not reach the population size 100. (b) The number of elites behaves about the same way as the number of non-dominated individuals in Figure (a). (c) The maximum number of achieved criteria increases after search for some generations, but the minimum does not change much, which is similar to that of Experiment Three.

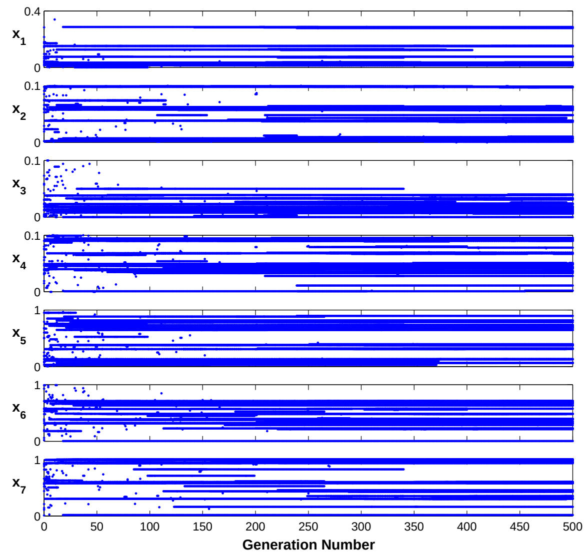


Figure 8.2: Parameter values of all non-dominated individuals found, by generation, for a trial with a simple dynamic crossover/mutation rate (Experiment Four). Population size is 100. As with the linear crossover/mutation probability (Experiment Three, Figure 7.4), the range of each parameter value does not change much from beginning to end. The number of individuals shown in the graph is less than those with the linear crossover/mutation probability because the number of non-dominated individuals is less than that with linear crossover/mutation probability.

with linear crossover/mutation probability (Experiment Three); either the numbers of non-dominated individuals or elites did not reach 100, the population size, in generations. Since the number of elites is more than half of the size at late generations, many parents are expected to already have been selected as elites. This means that many parents had a high probability of crossover.

As it was, the maximum number of achieved criteria increased when generation number became larger, and the minimum did not change much (Figure 8.1(c)). These phenomena are similar to those with the linear crossover/mutation probability (Experiment Three).

Seven parameter values are plotted in Figure 8.2. As with the linear probability in Figure 7.4, each parameter value took about the same value almost from the beginning to the end.

However, the distribution of each parameter value was not as wide as that in Figure 7.4. This is consistent to the fact that the number of non-dominated individuals was less (Figure 7.3(a)).

Figure 8.3 shows that the change of the Pareto frontier is also similar to that with linear probability (Figure 7.7); I obtained a large number of Pareto groups. Therefore, this simple dynamic crossover/mutation rate also brought many different Pareto groups.

Either the numbers of non-dominated individuals or elites did not reach the population size with Experiment Four (Figures 8.1(a) and 8.1(b)); many parents were already selected as elites, so most of them were crossed over to produce offspring which were dominated. After the number of elites stopped increasing, i.e., around generation the Pareto frontier did not changed much. Since the number of parents was too small due to the large number of elites, individuals with new Pareto group were not produced.

8.3 Limited Duration Elitism

Under Experiment Four with the dynamic crossover/mutation rate, once a non-dominated individual dominates one at the previous generation, it can survive as an elite until it is dominated. We saw in Section 7.2 that this allowed too many individuals to become elites. Therefore, as I mentioned, preserving those elites until they are dominated may be too strong a condition. In order to see if the results are affected by the length of duration, I set the maximum generation for which each Pareto group can be kept in external pool Q_g for Pareto groups.

I tried the maximum generation $g_{\mathrm{pr}} = 5, 20, 50, 100$ and 250; if a non-dominated individual is selected as an elite to be preserved, it will be preserved for at most g_{pr} generations. The number of non-dominated individuals is shown in Figure 8.4. As we can expect, the variance of the number of non-dominated individuals becomes smaller, especially as value of g_{pr} is

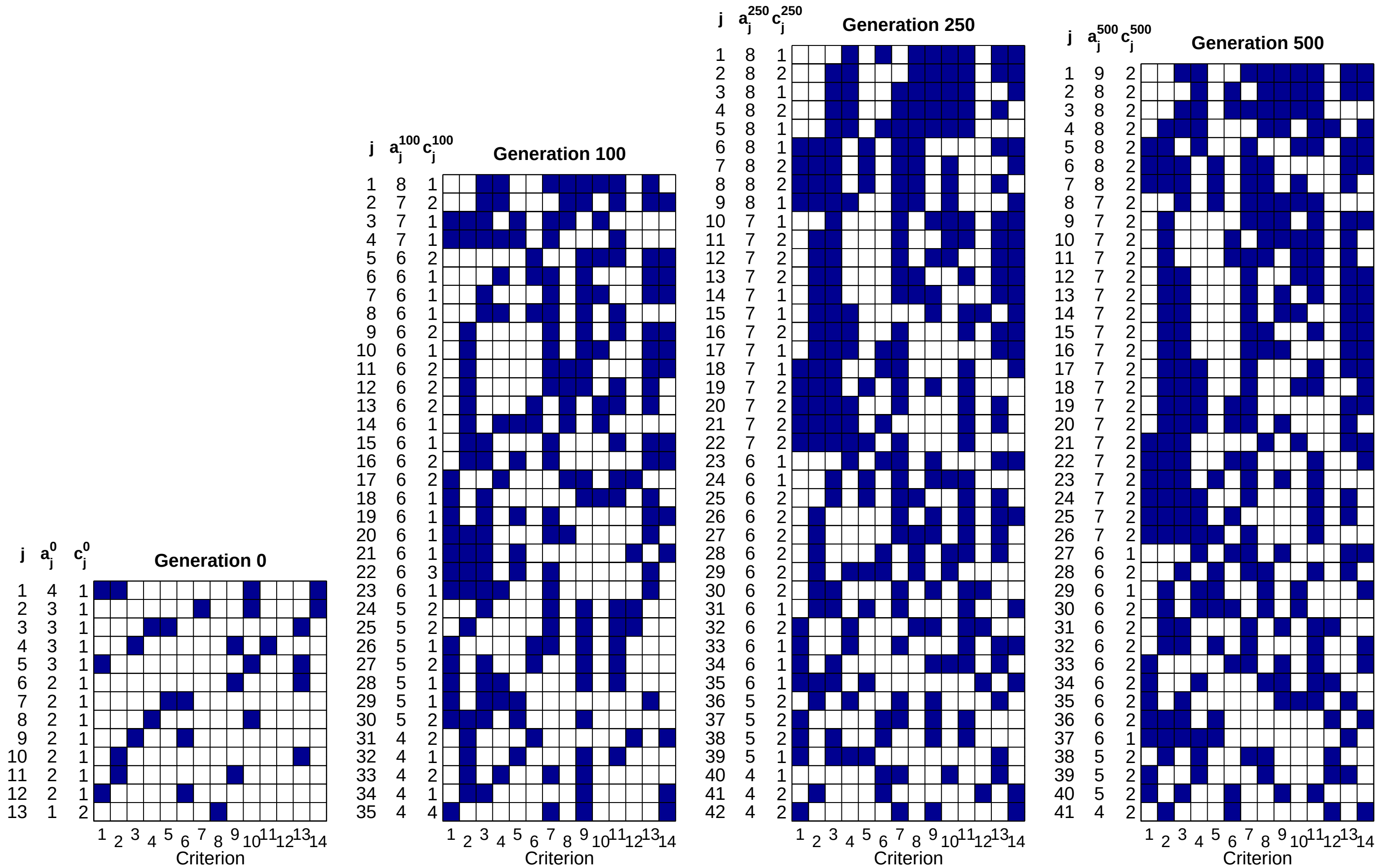


Figure 8.3: Pareto frontiers at generation 0, 100, 250 and 500 by Experiment Four. Population size is 100. Each plot corresponds to Pareto frontier $\{P_i^g | i = 1, 2, \ldots\}$ for generation g =0, 100, 250, 500. Three numbers i, a_i^g, c_i^g presented on the left of each plot are index of Pareto group P_i^g, number of achieved criteria by P_i^g, number of corresponding individuals to P_i^g, respectively, for generation g = 0, 100, 250, 500. Black and white squares represent achieved and unachieved, respectively. As to the Experiment Three, the Pareto frontier consists of many Pareto groups (Figure 7.7).

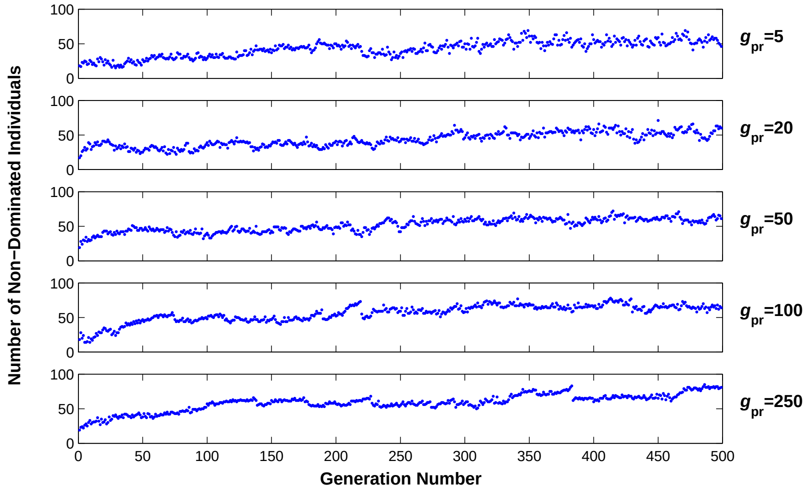


Figure 8.4: The number of non-dominated individuals with $g_{\text{pr}} = 5, 20, 50, 100$ and 250, where g_{pr} is the maximum generation for which each elite can be preserved. The variance becomes smaller, especially at late generations, as the value of g_{pr} gets large.

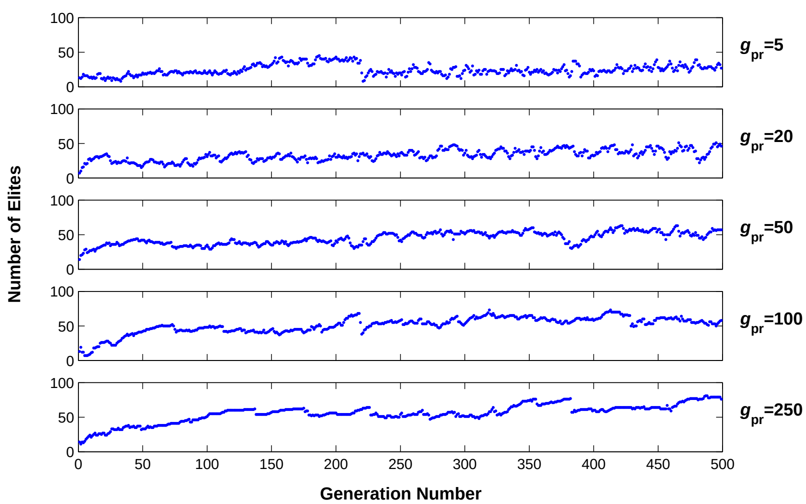


Figure 8.5: The number of elites with $g_{\text{pr}} = 5, 20, 50, 100$ and 250, where g_{pr} is the maximum generation for which each elite can be preserved. As for the number of non-dominated individuals (Figure 8.4), it stagnates as the values of g_{pr} increases.

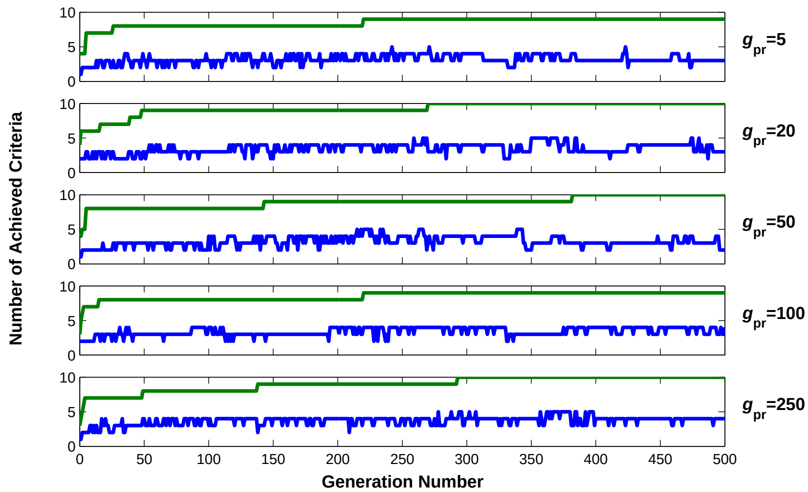


Figure 8.6: The maximum and minimum numbers of achieved criteria with $g_{\mathrm{pr}} = 5, 20, 50, 100$ and 250, where g_{pr} is the maximum generation for which each elite can be preserved. The maximum generation g_{pr} does not affect either maximum or minimum number of achieved criteria.

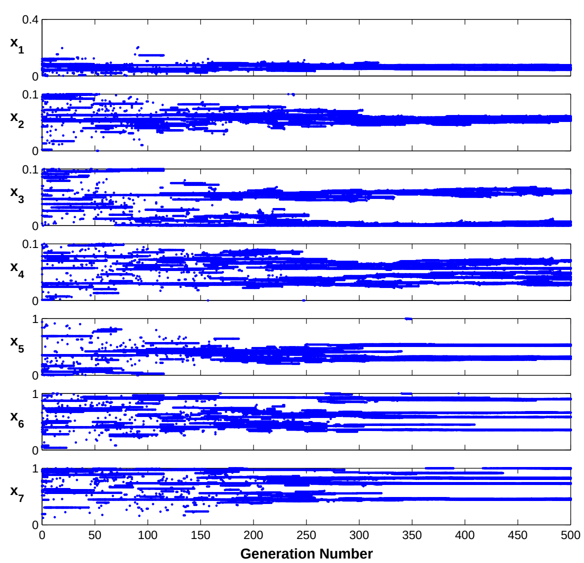


Figure 8.7: Parameter values of all non-dominated individuals found, by generation, with $g_{\mathrm{pr}} = 20$, where g_{pr} is the maximum generation for which each elite can be preserved in the external set Q_g. The values of x_2, x_3 and x_4 do not stabilized.

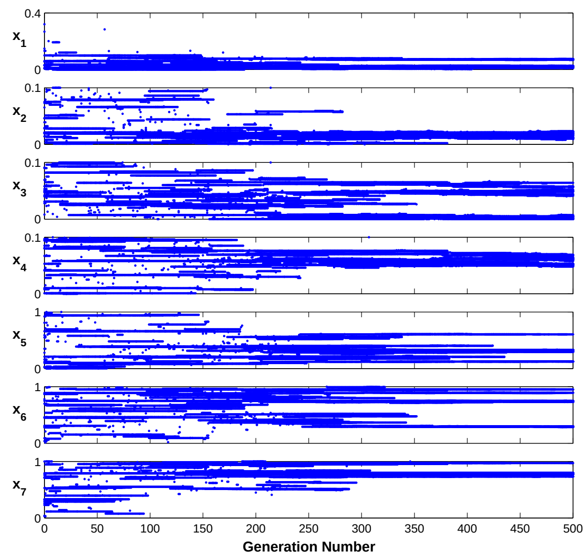


Figure 8.8: Parameter values of all non-dominated individuals found, by generation, with $g_{\text{pr}} = 50$. All values are stable through the search.

increased at late generations.

The number of elites behaves the same way as the number of non-dominated individuals (Figure 8.4 and 8.5). On the other hand, maximum and minimum number of achieved criteria do not seem to be affected by the maximum generation g_{pr} (Figure 8.6). For both Experiment Three and Four, the maximum number of achieved criteria increased when generation number became larger, and the minimum did not change much (Figures 7.3(c) and 8.1(c)). Thus, it is not surprising that g_{pr} does not affect to the results.

The seven parameter values of all non-dominated individuals with $g_{\text{pr}} = 20$ and $g_{\text{pr}} = 50$ are plotted in Figure 8.7 and Figure 8.8, respectively. we saw that the values for x_2, x_3 and x_4, which fluctuated in Experiment One (Section 5.1.1), again fluctuated for $g_{\text{pr}} = 20$, but not for $g_{\text{pr}} = 50$. The results are consistent with those of the number of non-dominated individuals (Figure 8.4) and elites (Figure 8.5) because they also became stable as g_{pr} got large. This also means that the search gets inactive or slow as g_{pr} becomes large.

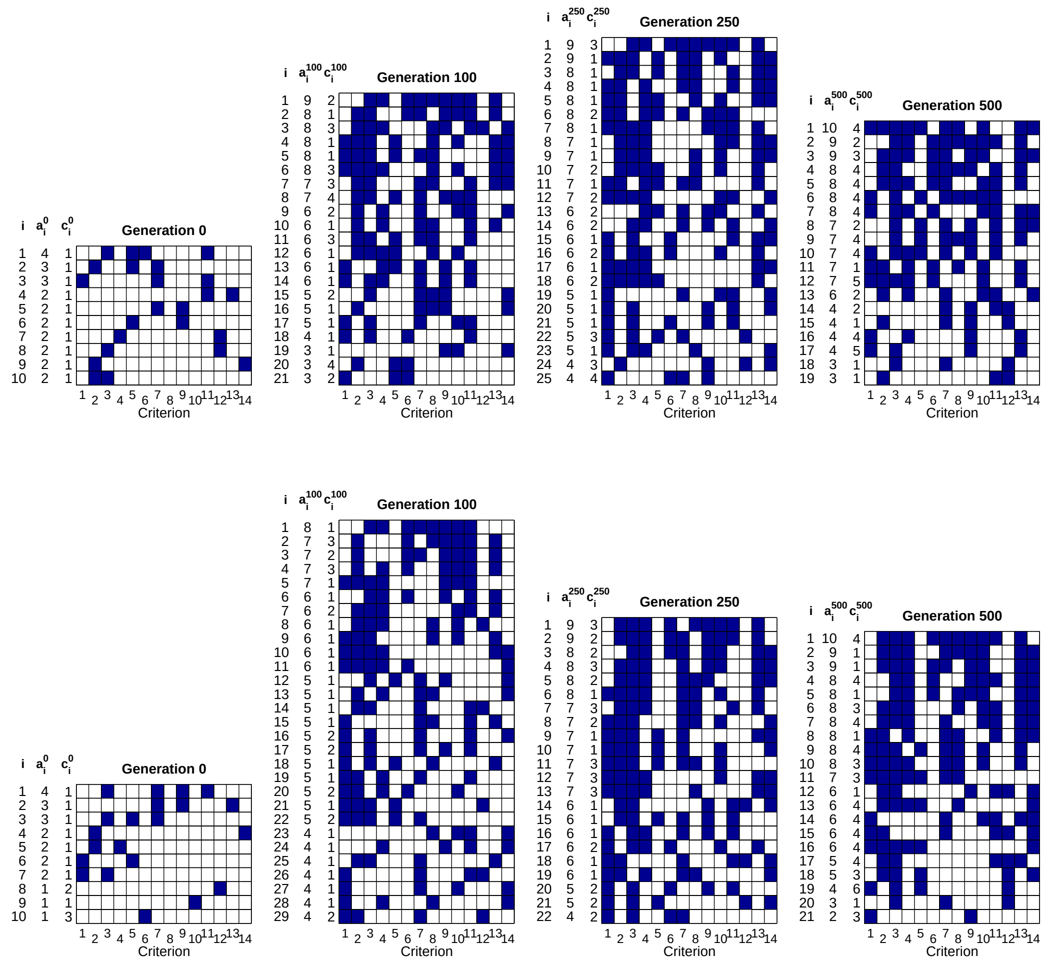


Figure 8.9: Pareto frontiers at generation 0, 100, 250 and 500 with $g_{\rm pr} = 20$ (top) and $g_{\rm pr} = 50$ (bottom). Population size is 100. Each plot corresponds to Pareto frontier $\{P_i^g | i = 1, 2, \cdots\}$ for generation $g =$0, 100, 250, 500. Three numbers i, a_i^g, c_i^g presented on the left of each plot are index of Pareto group P_i^g, number of achieved criteria by P_i^g, number of corresponding individuals to P_i^g, respectively, for generation $g = 0$, 100, 250, 500. Black and white squares represent achieved and unachieved, respectively.

The Pareto frontiers at generation 0, 100, 250 and 500 with $g_{\mathrm{pr}} = 20$ and $g_{\mathrm{pr}} = 50$ are plotted in Figure 8.9(a) and 8.9(b), respectively. The number of Pareto groups for $g_{\mathrm{pr}} = 20$ is about the same as that for $g_{\mathrm{pr}} = 50$.

I examined how the search is affected with different values of g_{pr}. We saw that the range of the numbers of the achieved criteria are not affected by g_{pr} and the Pareto frontiers are not so different between g_{pr} =20 and 50. However, while some of the parameters fluctuated with g_{pr} =20, any of them did not change their values much at late generations with g_{pr} =50.

Chapter 9

RESULTS OF ADDING A NEW CRITERION

I showed the results of several modifications of Pareto_Evolve to make search efficient to find the best fit to the data with many achieved criteria in the previous chapters. However, the error of the simulated data, i.e., the ratio of the residual sum squares (RSS) to the total sum of squares of hourly measured data, did not become smaller than 15%. This could be because of an inappropriate choice of objective functions or parameter search ranges. Actually, each of the simulated data sets with the smallest error (the ratio of the RSS to the total sum of squares of hourly measured data) were due to an individual with a Pareto group achieving only four criteria. Thus the simulation results did not capture the measured data well. This may be because the set of the fourteen criteria attempted to make the simulated data fit the measured data at only the fourteen points but not the other points. To fix this problem, I added one criterion to make the simulated data stay close also at the other points. As a new function, I used the RSS. This chapter shows the results of adding this criterion.

9.1 Search Criterion Using the Residual Sum of Squares

In order to see if adding the RSS as a criteria would help simulation results fit the measured data for the entire period, I added one more criterion. The objective function I used for the new criterion was the one used for the single-objective methods. I selected the RSS as the fifteenth criterion because I considered that it would help the simulated data stay close to the measured data not at the fourteen points but also at the other points.

The population size and generation number remained same, 100 and 500, respectively. For the parameter search ranges, I again set the ranges as shown in Table 3.2. For the first

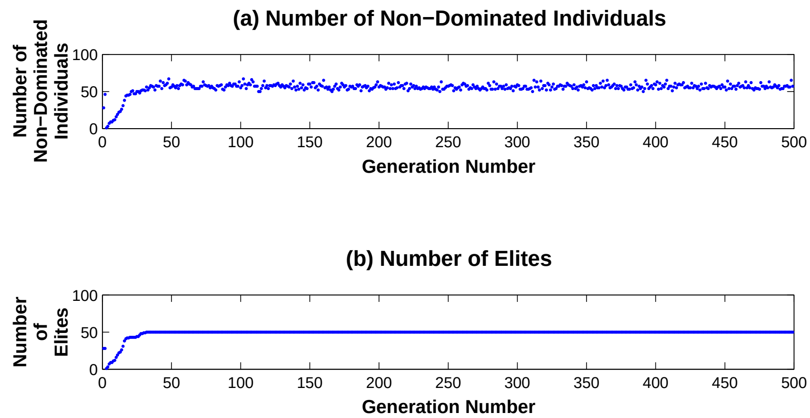


Figure 9.1: The numbers of (a) non-dominated individuals and (b) elites of a trial with three criteria, i.e., criterion 7, 8 and the RSS. Population size is 100. Both of them reached about half of the population size, which means most of the non-dominated individuals are copied as elites from the previous generations.

fourteen objective functions, I used the same criteria as the previous experiments; differences between measured and simulated growth data at the time when the measured growth data take their daily maximums and minimums. In Section 8.3, we saw that the parameters did not change their values much at late generations with $g_{\mathrm{pr}} = 50$, fifty generations for the duration of elites; thus I set $g_{\mathrm{pr}} = 50$. Since the single-objective methods Section 3.3 returned the value than 80 as the objective function value, I set the objective target range at $[0, 90]$ for the RSS.

Although I ran Pareto_Evolve for ten times with the target range $[0, 90]$ of the last criterion (RSS), it was never achieved. Thus, I relaxed its target range; increasing the value of the upper bound. I had to increase the value of the upper bound from 90 to 200 so that results from some of trials constantly achieved the criterion. I considered this happened because the first fourteen criteria were too restrictive for the search to make the last criterion to be achieved with a narrow target range or the model had some deficiency. To see if this was true, I reduced the number of criteria. I chose two criteria among the fourteen criteria with the RSS. Those two are the seventh and eighth i.e., the differences at the minimum and

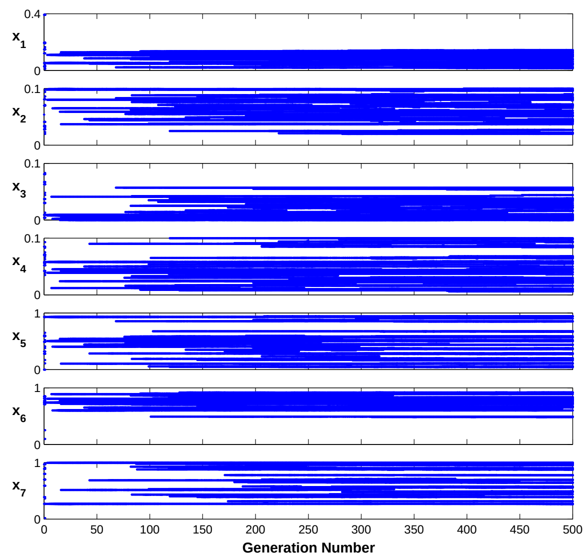


Figure 9.2: Parameter values of all non-dominated individuals found, by generation for a trial by search with the three criteria; the differences between the measured and simulated data at the minimum and maximum growth on day 181 and the RSS.

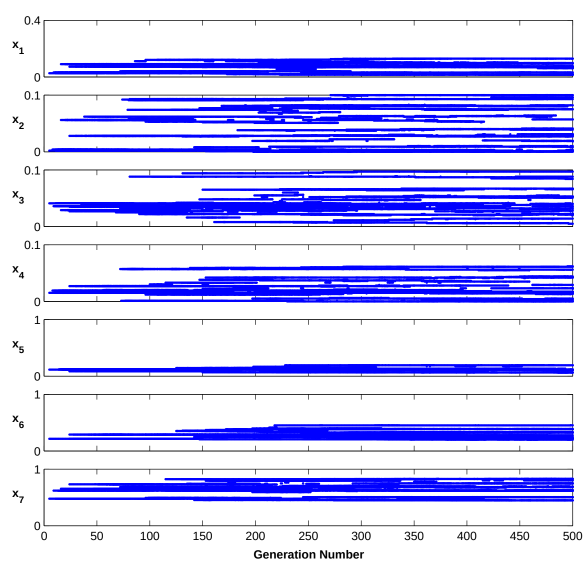


Figure 9.3: Parameter values of all individuals corresponding to Pareto group 001 found, by generation for a trial by search with the three criteria. Values of parameters x_2, x_3 and x_4 are in wider ranges of the search ranges through the entire search.

maximum growth on day 181. I selected this pair because the contraction period on day 181, where the point of the seventh criterion locates, was not captured well.

I set the objective range at $[0, 90]$ for the RSS this time. Although I decreased the number of criteria, the criterion for the RSS was hardly achieved with this target range. I tried twenty trials, but the criterion for the RSS was achieved only in one trial. When it was achieved, either of the other two criteria were not achieved; thus, the resulting Pareto group was 001. This implies that small differences between measured and simulated data at the minimum and maximum growth on day 181 and the small RSS conflict. There were only two criteria other than the RSS, but either of them were not achieved when the RSS was achieved. This means that the simulation result could not take a close value for either of the two points if the RSS was small or the simulation result captured the measured data pretty well for other periods. Therefore, I considered that the model has some problem but not the selection of criteria.

The number of non-dominated individuals and elites with the three criteria are plotted in Figures 9.1(a) and 9.1(b). Since the number of elites reached only half of the population size, the other half of the population were produced by breeding. However, the number of non-dominated individuals did not increase, and parameter values of non-dominated individuals did not change much (Figure 9.2); thus, most of the non-dominated individuals were copied as elites, and non-dominated individuals with small values of the RSS were unable to be produced by breeding. Therefore, the search stagnated.

Figure 9.3 shows the parameter values of all individuals corresponding to Pareto group 001. We can see that the parameterizations of x_2, x_3 and x_4, whose values fluctuated for Experiment One, were in wide ranges of the search ranges through the entire search. These ranges were search widely, but the RSS was achieved with no other criteria. Thus, I considered that the parameter search ranges for those three parameters were too narrow. Or the model itself has a problem. That's why those values cannot be fixed.

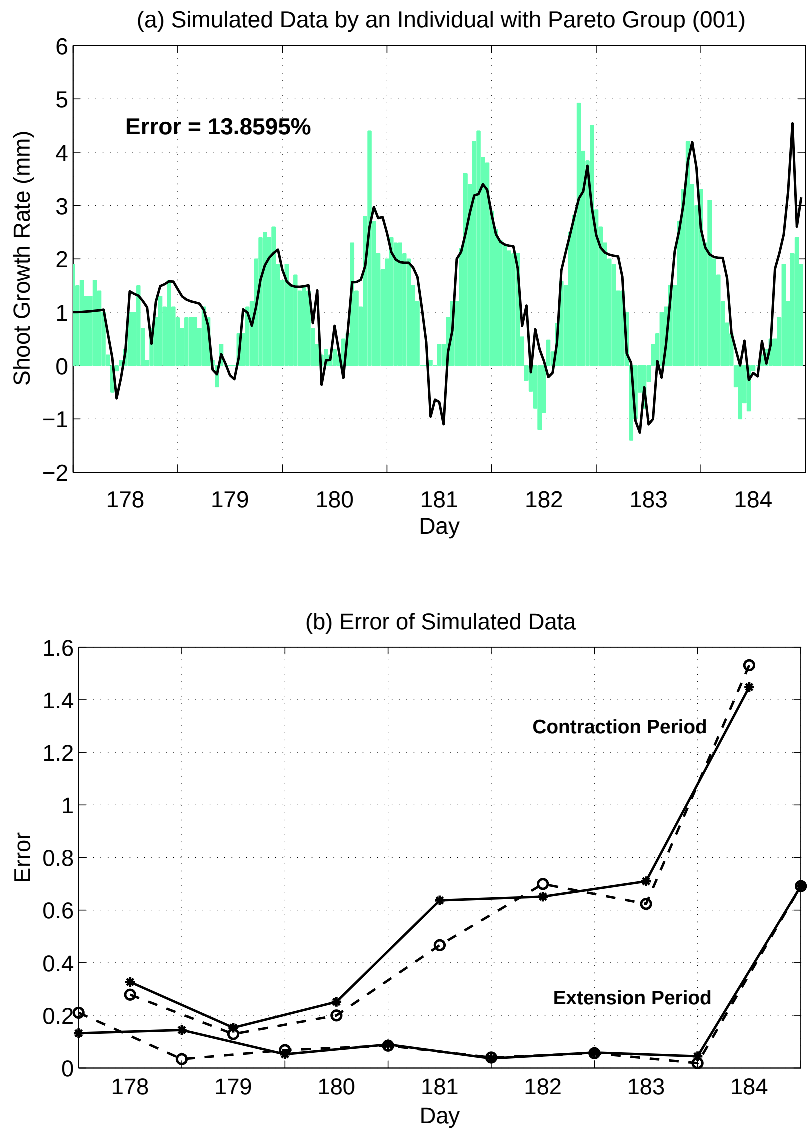


Figure 9.4: Results with the three criteria; two criteria on day 181 and the RSS. (a) The simulated data for the period of seven days, from Julian day 178 through 184 (solid line), and the measured data (bar). Population size is 100, and generation number is 500. Simulated data is by the individual giving the smallest error among all twenty trials. (b) The ratio of the RSS between the measured and simulated data to the total sum of squares of the measured data for day 178-184 (solid line) and the ratio between the measured and simulated data by the Nelder-Mead simplex method (dashed line, Figure 3.3(b)). Errors for contraction periods (6:00-18:00) and extension periods (18:00-6:00+) are plotted separately. As for the result of Experiment Three, the fit on the contraction periods was not as good as that by the simplex methods (Figure 3.3(b)).

Table 9.1: Parameter search ranges and step sizes for a search to see if there is any problem about parameter ranges of x_2, x_3 and x_4. Old values are in parentheses.

Parameter	Lower Bound	Upper B.	(old)	Step Size	(old)
x_1	0.0	0.4		0.004	(0.001)
x_2	0.0	0.2	(0.1)	0.002	(0.001)
x_3	0.0	0.2	(0.1)	0.002	(0.001)
x_4	0.0	0.2	(0.1)	0.002	(0.001)
x_5	0.0	1.0		0.01	(0.001)
x_6	0.0	1.0		0.01	(0.001)
x_7	0.0	1.0		0.01	(0.001)

The simulated data with the smallest error with the three criteria is plotted in Figure 9.4(a). Since I set the maximum allowed values for the RSS small enough, the fitting was as good as that by single-objective methods. However, Figure 9.4(b) shows that the fit on the contraction periods was not as good as that by the simplex methods (Figure 3.3(b)).

9.2 *Relaxed Parameter Search Ranges*

From the results in the previous section, we saw that the parameters x_2, x_3 and x_4 took a wider variety of values than the other four parameters through the entire search. In order to see if this happened because of the parameter search ranges, I relaxed the search ranges of these three parameters. I also made a step size for each parameter larger to allow a more global search. The new search ranges and the step sizes are shown in Table 9.2.

I used the original fourteen objective functions as search criteria. Although I executed Pareto_Evolve ten times, the simulated data did not capture the measured data well; the error of the simulated data was at least 20% for each trial.

Since relaxing the search ranges of x_2, x_3 and x_4 did not help the error get smaller, I tried searching with fixed values of x_1, x_5, x_6 and x_7. The resulting parameterizations of the simulated annealing method (Section 3.3.3) were used for the four fixed parameter values:

$$x_1 = 0.070972, \quad x_5 = 0.106354, \quad x_6 = 0.270681, \quad x_7 = 0.566628.$$

I returned the search ranges of the other three parameters to the original ones (Table 3.2). I again tried ten times, for all trials, and some Pareto groups with parameterizations gave errors of less than 15%. However, the maximum number of achieved criteria was only five for all ten trials. This implies that it is impossible to make the error small with many achieved criteria. Therefore, the fourteen criteria I used do not help provide good fits although the three parameter search ranges are relaxed or the other four parameter values are fixed.

Chapter 10

CONCLUSION FROM THE RESULTS

10.1 Defects of the Model

From the results of the previous chapters, I found that the model might have some problems. To investigate this more carefully, I examined the result of the trial presented in Section 8.3 with $g_{\mathrm{pr}} = 50$, fifty generations for the duration of elites. Table 10.1 shows how many Pareto groups achieved each combination of pairs of criteria at the last generation (generation 500). From this table, we see that criterion 6 was never achieved by any Pareto member when criterion 1 was achieved. Also, when criterion 5, 6 or 12 was achieved, there were not many Pareto groups, i.e., other criteria were difficult to achieve with criteria 5, 6 and 12. I examined the results of all of the ten trials when those three criteria were achieved (Tables 10.2, 10.3, and 10.4). Tables 10.2 and 10.3 show that some criteria were achieved by more than five Pareto groups with criterion 5 or 6. For example, there are more than five Pareto groups achieving criteria 1 and 5 at the same time in Trial 4 in Table 10.2. However, we see that if either criterion 5 or 6 was achieved, then the other criteria were rarely achieved. This means that criteria 5 and 6 were difficult to achieve at the same time, but not separately. On the other hand, Table 10.4 shows that criterion 12 was difficult to achieve in general. In ten trials, if criterion 12 was achieved, then either criterion 6 or 10 were never achieved, and criteria 4 and 8 were rarely achieved.

Criterion 12 is for the extension period from day 183 through day 184. Although we saw that it was difficult to achieve, Figures 3.3(b), 7.5(b), and 9.4(b) show that the error at this extension period was not high. Also, the criterion was achieved when the results with small errors were obtained (Figures 4.4(b), 5.11, and 7.5(a)). Therefore, when an error was small, the fit at the extension period from day 183 through day 184 was good. However, as

Table 10.1: The numbers of Pareto groups that achieved each combination of pairs of criteria at the last generation (generation 500). For example, number six at the left top corner shows that there exists six Pareto groups that achieve criteria 1 and 2 at the same time.

Criteria	2	3	4	5	6	7	8	9	10	11	12	13	14
Criteria 1	6	5	4	4	0	6	4	1	3	3	2	4	4
2		13	10	5	2	9	8	4	7	7	4	9	10
3			11	6	4	8	8	6	9	6	5	10	10
4				3	4	7	8	5	8	5	0	9	9
5					0	2	3	0	1	0	2	1	3
6						2	2	4	4	2	0	4	3
7							6	3	7	5	2	8	6
8								3	6	3	0	7	5
9									5	3	1	5	5
10										5	0	10	7
11											3	7	7
12												1	3
13													8

we saw above, criterion 12 was difficult to achieve in general, From this result, I considered that the model is deficient.

To investigate how the Pareto frontier changes without criterion 12, I dropped the criterion and executed Pareto_Evolve with thirteen criteria ten times. Although there was one less criterion, the average number of achieved criteria by a non-dominated individual became larger in general (Table 10.5 and Figure 10.1). However, even though it was dropped, as before, criteria 5 and 6 were rarely achieved simultaneously (Tables 10.2 and 10.3); thus the difficulty in criteria 5 and 6 was different from that of criterion 12. After dropping criterion 12, the average numbers without just criterion 5 or 6 or 5 and 6 became larger than that without just criterion 12, but those average numbers were not much different from

Table 10.2: The numbers of Pareto groups for all ten trials that achieved the other criteria at the last generation (generation 500) when criterion 5 was achieved. For example, in trial 1, there are four Pareto groups that achieve criterion 1 when criterion 5 is achieved.

Criteria	1	2	3	4	6	7	8	9	10	11	12	13	14
Trial 1	4	5	6	3	0	2	3	0	1	0	2	1	3
2	6	6	4	3	0	4	3	2	0	6	3	2	4
3	3	3	3	2	0	2	2	0	2	0	1	2	1
4	7	8	8	3	0	7	4	2	4	5	1	6	4
5	5	7	7	2	0	7	2	4	3	5	3	3	2
6	3	2	2	1	0	2	1	0	0	0	1	2	0
7	0	1	2	0	0	2	1	0	0	2	1	1	1
8	5	3	6	2	2	3	3	1	1	3	1	4	3
9	3	4	5	1	0	3	2	1	2	2	1	1	1
10	4	6	5	1	0	5	3	4	2	4	2	5	2

Table 10.3: The numbers of Pareto groups for all ten trials that achieved the other criteria at the last generation (generation 500) when criterion 6 was achieved.

Criteria	1	2	3	4	5	7	8	9	10	11	12	13	14
Trial 1	0	2	4	4	0	2	2	4	4	2	0	4	3
2	2	2	3	4	0	1	2	1	3	2	0	4	2
3	2	4	4	5	0	2	3	3	4	1	0	5	3
4	2	3	7	7	0	4	2	4	5	3	0	6	5
5	1	1	1	0	0	1	0	1	0	0	0	2	1
6	3	3	6	5	0	3	4	4	6	6	0	6	6
7	5	3	4	3	0	3	1	2	2	4	0	4	2
8	8	4	10	10	2	3	6	4	8	6	0	10	7
9	4	4	6	7	0	2	3	4	5	4	0	5	6
10	3	2	6	4	0	3	1	4	2	3	0	6	5

Table 10.4: The numbers of Pareto groups for all ten trials that achieved the other criteria at the last generation (generation 500) when criterion 12 was achieved.

Criteria	1	2	3	4	5	6	7	8	9	10	11	13	14
Trial 1	2	4	5	0	2	0	2	0	1	0	3	1	3
2	4	3	3	0	3	0	2	0	0	0	4	0	4
3	1	0	2	0	1	0	1	0	0	0	0	0	0
4	1	3	3	1	1	0	1	0	0	0	1	1	2
5	1	4	2	0	3	0	3	0	1	0	2	0	1
6	2	4	5	2	1	0	1	1	1	0	3	1	2
7	0	0	1	0	1	0	1	0	0	0	1	0	0
8	1	2	1	0	1	0	2	0	1	0	3	1	1
9	1	1	2	0	1	0	1	0	0	0	0	0	0
10	0	2	1	0	2	0	2	0	1	0	2	0	1

Table 10.5: The number of non-dominated individuals (top) and the average number of achieved criteria by a non-dominated individual for ten trials at the last generation (generation 500). For example, when criteria 5, 6, and 12 are removed and Pareto_Evolve was executed with eleven criteria, in trial 1, fifty-two non-dominated individuals were obtained and 8.06 criteria were achieved by those individuals in average.

Trial	1	2	3	4	5	6	7	8	9	10
All fourteen criteria	63	62	60	60	61	65	83	64	56	55
	6.49	6.65	6.63	6.95	6.11	6.86	6.82	6.61	7.11	6.40
Without criterion 12	47	44	50	48	43	37	51	51	41	57
	6.81	7.45	6.62	7.35	7.53	7.24	7.00	6.59	7.10	5.98
Without criteria 5 or 12	46	45	45	36	54	39	48	46	39	56
	7.28	7.11	7.00	7.94	6.87	8.26	7.42	6.61	7.85	7.02
Without criteria 6 or 12	44	41	40	34	39	39	40	33	39	45
	7.70	8.00	7.22	8.32	8.41	8.15	7.33	7.73	7.41	7.58
Without criteria 5, 6 or 12	52	40	51	46	46	50	38	52	37	41
	8.06	6.85	7.53	7.63	8.26	7.16	7.79	7.96	6.41	7.15

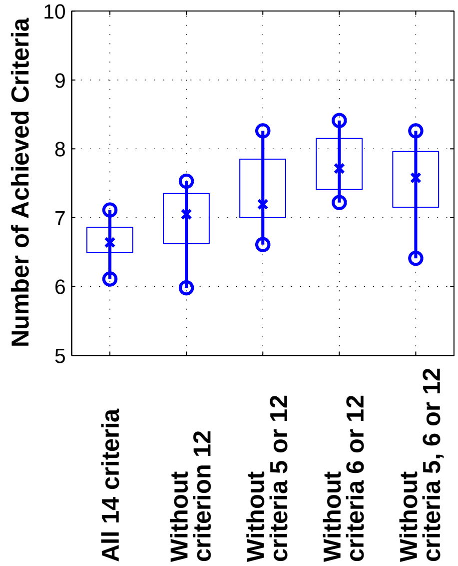


Figure 10.1: Box and whisker plots of the average numbers of achieved criteria by a non-dominated individual for ten trials at the last generation (generation 500). Each box represents a range between the first and third quartiles. Symbol "o"'s shows the smallest and largest values, and "x"'s represent the median.

each others (Table 10.5 and Figure 10.1). Although the three criteria that had difficulty in achievement were dropped, not all criteria were achieved by a Pareto group.

10.2 Modification of the Model

Since the modifications of the search method did not bring simulation results with smaller values of the RSS than those by the single-objective methods, I widened the search ranges of some parameters and changed the search criteria. From the results, I concluded that the model was deficient. In this section, I show how I think that the model should be modified.

First, in order to see how sensitive the solar radiation parts are to the model introduced in Section 3.2, I removed the third or fourth or both the third and fourth terms, which are the total solar radiation for two and three 24-hour periods before the hours being considered.

However, I did not obtain different results from before by this procedure. I also added the total solar radiation for four 24-hour periods before, and found that the first few criteria were sensitive, but that the value of RSS did not become smaller. Therefore, I concluded that the model is not very sensitive to the solar radiation terms.

As I showed in Figure 7.5(b), the model seems to be biased so that it achieves extension periods with grater accuracy than contraction periods. Thus, I next changed the last term of the model, which is for water deficit, because it is applied separately for the extension and contraction periods. Instead of the total sum of growth on the previous day, I tried the total sum of growth for the previous cycle of contraction and extension periods, that is, for 24 hours from 6:00 in the morning on the previous day. However, this did not make the model less biased so that it achieves extension periods with greater accuracy than contraction periods.

Although the last term is separated for extension and contraction periods, they have the same form and each of the parameters has the same range. Since the shoot extends differently for extension and contraction periods, it is necessary to consider the difference between physical reactions of extension and contraction periods to construct a less biased model.

Figure 10.2 shows the water deficit D_t and the difference of water deficit $\Delta_t D$ calculated with the parameterizations of the smallest error (the ratio of the RSS to the total sum of squares of hourly measured data) of the search introduced in Section 8.3 with duration of fifty generations. From Figure 3.1, we see that the magnitudes of growth S_t in extension periods are generally larger than that in contraction periods. The last term of the model determines the amount of addition ($\Delta_t D < 0$, i.e., expansion periods) and subtraction ($\Delta_t D \geq 0$, i.e., contraction periods). Thus, the magnitude of last term for contraction periods has to be smaller than that for extension periods in general. As we see in Figure 10.2(b), the magnitude of $\Delta_t D$ in extension periods are generally larger than those in contraction periods; thus first, I changed the parameter search ranges for x_5 and x_6 so that magnitude of the last term for extension periods could be larger than that for contraction periods. I

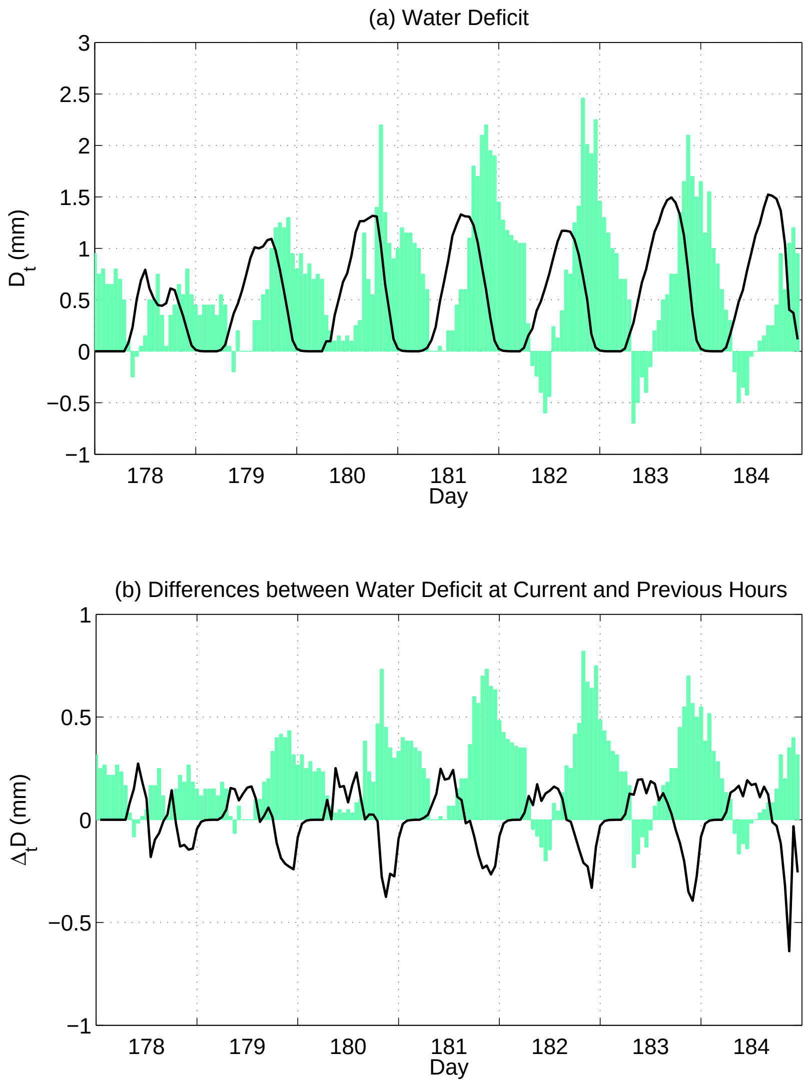


Figure 10.2: (a) Water deficit D_t and (b) difference of water deficit $\Delta_t D$ calculated with the parameterizations of the smallest error of the search introduced in Section 8.3 with duration of fifty generations.

tried two difference cases: $0 \leq x_5 \leq 1$, $0 \leq x_6 \leq 0.5$ and $0.5 \leq x_5 \leq 1$, $0 \leq x_6 \leq 0.5$. I ran each trial several times. For the first case, the largest errors (the ratio of the RSS to the total sum of squares of hourly measured data) were 50% or more, and for the second case, many errors were more than 100%. Fits did not only become better but the bias against the extension also was not fixed.

Next, to make the magnitude of the water deficit term for extension periods larger than that for contraction periods, I squared the sum $\sum_{k=1}^{24} S_k^*$ for contraction periods. The results searched by the same parameter search ranges as in Table 3.2 with one of the smallest errors are shown in Figure 10.3. The smallest errors are much larger than those with experiments introduced in the previous chapters, but one of the errors for contraction periods (day 180) became smaller than that for the previous extension period. Also, I obtained a Pareto group with eleven achieved criteria where the maximum number of achieved criteria was ten before.

Since these simple modifications did not substantially reduce the RSS, I changed the model to represent the physical process of contraction more effectively. Contraction depends on net growth made in the previous cycle, i.e., this defines the amount of newly made tissue that is hardened by cell wall thickening and so may contract when water deficit increases. On the other hand, the extension due to reduction in water deficit should operate on the contraction that has just occurred. Therefore, I changed the last term of the model,

$$- \begin{cases} x_5 \cdot \Delta_t D \cdot \sum_{k=1}^{24} S_k^* & (\Delta_t D < 0) \\ \\ x_6 \cdot \Delta_t D \cdot \sum_{k=1}^{24} S_k^* & (\Delta_t D \geq 0), \end{cases}$$

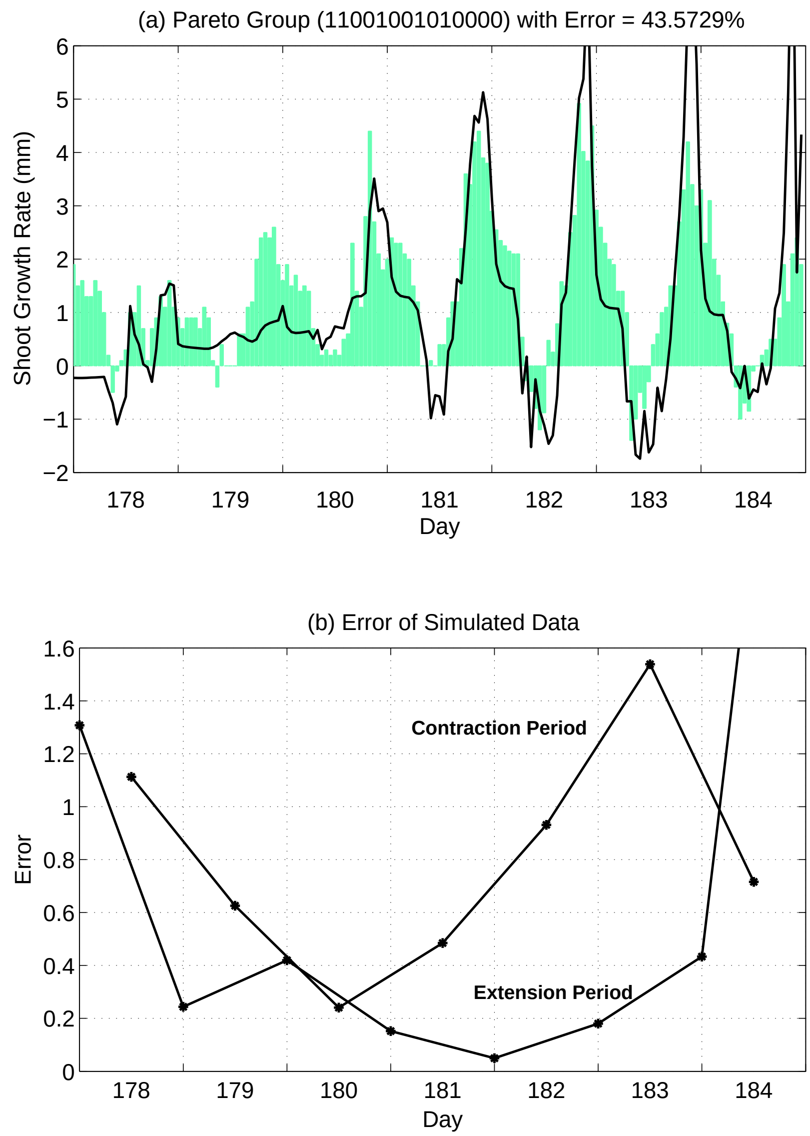
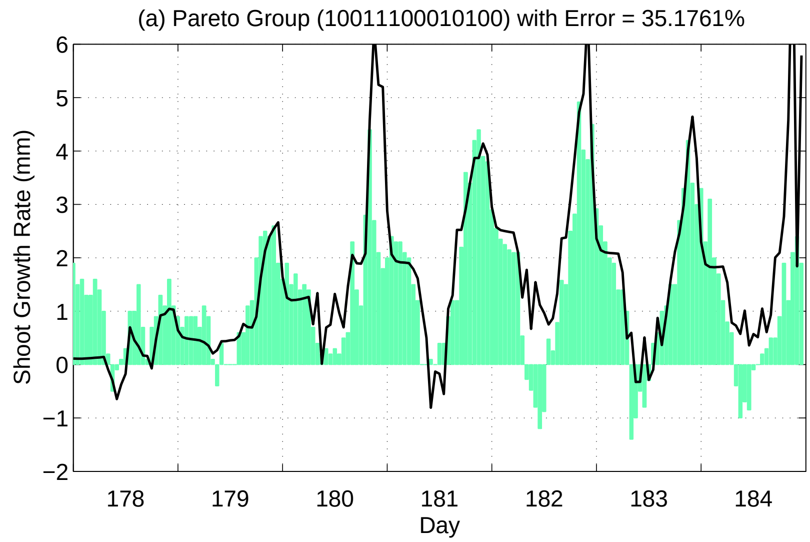


Figure 10.3: The sum in the last term of the model for contraction periods is squared. The simulated data is by the parameterization with one of the smallest error among all ten trials. (a) The simulated data for the period of seven days, from Julian day 178 through 184 (solid line), and the measured data (bar). Population size is 100, generation number is 500, and duration is 50 generations. Simulation overestimates the measured data. (b) The ratio of the RSS between the measured and simulated data to the total sum of squares of the measured data for day 178-184. Errors for contraction periods (6:00-18:00) and extension periods (18:00-6:00+) are plotted separately. Extension periods on the first and last days are 0:00-6:00 and 18:00-0:00, respectively. Some errors in contraction periods are smaller than those in extension periods.



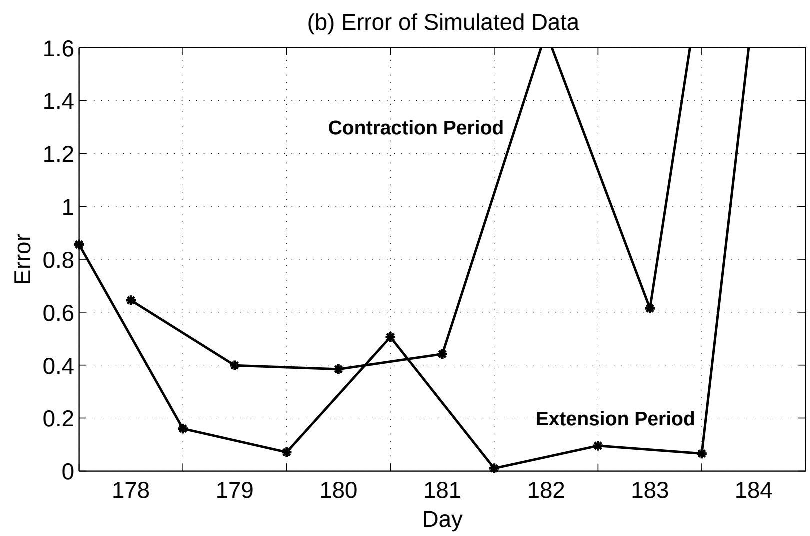


Figure 10.4: The sums in the last term of the model for both expansion and contraction periods are modified. (a) The simulated data for the period of seven days, from Julian day 178 through 184 (solid line), and the measured data (bar). Population size is 100, generation number is 500, and duration is 50 generations. (b) The ratio of the RSS between the measured and simulated data to the total sum of squares of the measured data for day 178-184. Errors for contraction periods (6:00-18:00) and extension periods (18:00-6:00+) are plotted separately. Extension periods on the first and last days are 0:00-6:00 and 18:00-0:00, respectively. Some errors in contraction periods are smaller than those in extension periods.

to

$$-\begin{cases} x_5 \cdot \Delta_t D \cdot \left(\sum\limits_{\Delta D^* > 0} S_k^* - \gamma_{t-1} \right) & (\Delta_t D < 0) \\[2em] x_6 \cdot \Delta_t D \cdot \left(\sum\limits_{k=7}^{24} S_k^{**} + \sum\limits_{k=1}^{6} S_k^* \right) & (\Delta_t D \geq 0). \end{cases}$$

The term $\sum\limits_{\Delta D^* > 0} S_k^*$ is the sum of growth for the period right before the current time when the increment of water deficit is positive and $\gamma_t = \sum\limits_{\Delta D^* > 0} S_k^* - \gamma_{t-1}$. On the other hand, for $\Delta_t D > 0$, the sum of growth from 6:00 on the previous day to 6:00 on the current day is used, where S_k^{**} and S_k^* are growth rates for the previous and current days, respectively.

I ran Pareto_Evolve for ten times with population size 100, generation number 500 and duration 50 generations. The smallest error is about 25% among the ten trials. One of simulation result is plotted in Figure 10.4(a). Although the RSS is larger than the smallest one calculated with the previous model, the model results are less biased against expansion periods (Figures 7.5(b) and 10.4(b)). In constructing a model of a biological process, it may be just as important to avoid bias as it is to minimize the RSS and multi-objective assessment can inform us about this bias.

10.3 Conclusion

Using a multi-objective method, I tried to improve the results of fitting with single-objective methods and I provided an analysis of for which criteria the model had difficulty with. I thus attempted to analyze where the model was deficient. In general, for a multi-objective optimization problem, it is hard to achieve all criteria simultaneously. I had this difficulty in my problem, too. After some analysis, I assumed that it was from a deficiency of the model.

First, I introduced elitism. With elitism, some individuals were selected as elites depending on their assessment vectors, and they were copied to the population of the next generation. For the first three experiments (Chapter 5-7), the resulting fits were not as good as that of single-objective methods. Also, the number of elites reached the population size before the generation reached its maximum, which means that the search stopped at that point.

After I modified elitism, not too many elites were copied to the next generation, so the number of produced offspring became larger than those of the previous experiments. However, the Pareto frontier did not change much at each generation, and also the resulting fit with the smallest error, i.e., the ratio of the residual sum of squares (RSS) to the the total sum of squares of hourly measured data, did not become better than that found using single-objective methods. This was possibly because fitting to the measured data was considered only on the fourteen points, and the set of criteria I used does not guarantee that the simulated data fit the measured data on the other points. Thus, I added the RSS as a new criterion (Section 9.1). Since this new criterion with a narrow objective target range was not achieved with the original fourteen criteria, I decreased the number of criteria; two from the original fourteen criteria and the RSS. Execution with the three criteria brought a smaller error than those of the previous experiments, but not smaller than those of single-objective methods. The number of non-dominated individuals were kept at only half of the population size, and almost all of them were just copied as elites from the previous generations. Thus, the other half of the population were produced by breeding, but they did not become non-dominated for late generations. Also, if the RSS was achieved, the other two criteria were not achieved, which means a small value of the RSS conflicted with fits at the selected two points. There were only two criteria other than the RSS, but either of them were not achieved when the RSS was achieved. Thus, the simulation result could not take a close value for either of the two points if the RSS was small or the simulation result captured the measured data pretty well for other periods. Therefore, I considered that the model has some problem but not the selection of criteria.

I noticed that the parameterizations of x_2, x_3 and x_4 had wide ranges in the experiment with three criteria. To investigate effects of the search ranges of those three parameters, I relaxed the search ranges of these parameters and executed Pareto_Evolve with the original fourteen criteria. (Section 9.2). Since errors did not become as small as those of single-objective methods, I tried the search with some fixed parameter values. Parameters x_1, x_5, x_6 and x_7, whose values did not fluctuate, were selected to fix. The resulting errors were nearly as small as those of single-objective methods, and those small errors were obtained for each of the ten trials. However, there was a problem because the maximum number of achieved criteria was only five for all of the ten trials. Again, having a small value of the RSS conflicts with achieving many criteria.

The smallest error I obtained by both single and multi-objective methods was more than 13%, and it did not become smaller even though I modified some search methods or changed the set of criteria or parameter search ranges. Therefore, I considered that the error does not become smaller with the current model introduced in Section 3.2. Thus, I reconsidered the model. The modified model did not return results with small errors, but it was less biased against extension periods. After modification of the model, the bias of the model became smaller; that is, accuracy of achievement at contraction periods was improved. However, the error did not become small. Therefore, I concluded that more biological information about contraction and expansion is needed.

One possibility to construct a better model is selection of different type of criteria. I used a set of simple assessment criteria with the revised model. However, if assessment criteria themselves are used to define the model, defect of the model may be reduced. For example, all of sums in the model, i.e., the average temperature, total solar radiation and total growth, are based on a twenty-four hour period, but length of the period for those sums could be defined by assessment criteria.

The single-objective methods returned the result with the smallest errors. The multi-objective method did not give a better fit, but from analysis of its results, the model deficiencies were found; that is, I used the multi-objective method to find what the problem of the model was.

BIBLIOGRAPHY

[1] Bäck, T., F. Hoffmeister, H-P Schwefel. A survey of evolution strategies. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann, 1991.

[2] Bäck, T., U. Hammel, H-P. Schwefel. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3–17, 1997.

[3] Baker, J. E. Adaptive selection methods for genetic algorithms. In *Proceedings of an International Conference on Genetic Algorithms and their Application*, pages 101–111. Lawrence Erlbaum Associates, 1985.

[4] Banzhaf, W., P. Nordin, R. E. Keller F. D. Francone. *Genetic Programming: An Introduction: On the Automatic Evolution of Computer Programs and Its Applications.* Morgan Kaufmann, 1997.

[5] Beasley, D., D. R. Bull, R. R. Martin. An overview of genetic algorithms: Part 1, fundamentals. *University Computing*, 5:56–69, 1993.

[6] Box, G. E. P., G. M. Jenkins. *Time Series Analysis; Forecasting and Control.* Holden-Day, 1970.

[7] Das, R., M. Mitchell, J. P. Crutchfield. A genetic algorithm discovers particle-based computation in cellular automata. In *Parallel Problem Solving From Nature III*, pages 344–353. Springer-Verlag, 1994.

[8] De Jong, K. A. *An Analysis of the Behavior of a Class of Genetic Adaptive Systems.* PhD thesis, University of Michigan, 1975.

[9] Deans, J. D. Fluctuations in the soil environment and find root growth in a young Sitka spruce plantation. *Plant and Soil*, 52:195–208, 1979.

[10] Deb, K., S. Agrawal, A. Pratap, T. Meyarivan. A first elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858. Springer, 2000.

[11] Fogel, D. B. An evolutionary approach to the traveling salesman problem. *Biological Cybernetics*, 60:139–144, 1988.

[12] Fogel, D. B. Applying evolutionary programming to selected control problems. *Computers and Mathematics with Applications*, 27:89–104, 1994.

[13] Fogel, D. B. *Evolutionary Computation.* IEEE Press, 2nd edition, 2000.

[14] Fogel, L. J. Autonomous automata. *Industrial Research*, 4:14–19, 1962.

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

[16] Ford, E. D., J. D. Deans, R. Milne. Shoot extension in *Picea sitchensis* I. seasonal variation within a forest canopy. *Annals of Botany*, 60:531–542, 1987.

[17] Ford, E. D., R. Milne, J. D. Deans. Shoot extension in *Picea sitchensis* II. analysis of weather influences on daily growth rate. *Annals of Botany*, 60:543–552, 1987.

[18] Goldberg, D. E., B. Korb, K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5):493–530, 1989.

[19] Goldberg, D. E., J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms and their Application*, pages 41–49. Lawrence Erlbaum Associates, 1987.

[20] Grefenstette, J. J. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man and Cybernetics*, 16(1):122–128, 1986.

[21] Holland, J. H. Outline for a logical theory of adaptive systems: An introductory analysis with applications to biology, control, and artificial intelligence. *Journal of the Association for Computing Machinery*, 9(3):297–314, 1962.

[22] Holland, J. H. *Adaptation in Natural and Artificial Systems.* MIT Press, 1992.

[23] Horn, J., N. Nafpliotis. Multiobjective optimization using the niched Pareto genetic algorithm. IlliGAL Report 93005, Illinois Genetic Algorithm Laboratory (IlliGAL), University of Illinois at Urbana-Champaign, July 1993.

[24] Keeney, R. L., H. Raiffa. *Decisions with Multiple objectives: Preferences and Value Tradeoffs.* John Wiley & Sons, 1976.

[25] Kirkpatrick, S., C. D. Gelatt, M. P. Vecchi. Optimization by Simulated Annealing. *Science*, 220:671–680, 1983.

[26] Koza, J. R. *Genetic Programming: on the Programming of Computers by Means of Natural Selection.* MIT Press, 1992.

[27] Kursawe, F. A variant of evolution strategies for vector optimization. In *Parallel Problem Solving From Nature*, pages 193–197. Springer-Verlag, 1991.

[28] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolutionary Programs.* Springer-Verlag, 1996.

[29] Milne, R., E. D. Ford, J. E. Deans. Time lags in the water relations of Sitka spruce. *Forest Ecology and Management*, 5:1–25, 1983.

[30] Milne, R., J. D. Deans, E. D. Ford, P. G. Jarvis, J. Leverenz, D. Whitehead. A comparison of two methods of estimating transpiration rates from a Sitka spruce plantation. *Boundary Layer Meteorology*, 32:155–175, 1985.

[31] Milne, R., S. K. Smith, E. D. Ford. An automatic system for measuring shoot length in Sitka spruce and other plant species. *Applied Ecology*, 14:523–529, 1977.

[32] Moilanen, A. Simulated evolutionary optimization and local search: Introduction and application to tree search. *Cladistics*, 17:512–525, 2001.

[33] Nelder, J. A., R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[34] Peña-Reyes, A., M. Sipper. Evolutionary computation in medicine: An overview. *Artificial Intelligence in Medicine*, 19:1–23, 2000.

[35] Powell, M. J. D. A direct search optimization method that models the objective and constraint functions by linear interpolation. In *Advances in Optimization and Numerical Analysis: Proceedings of the Sixth Workshop on Optimization and Numerical Analysis, Oaxaca, Mexico*, pages 51–67. Kluwer Academic Publishers, 1994.

[36] Press, W. H., B. P. Flannery, S. A. Teukolsky, W. T. Vetterling. *Numerical Recipes in C: the Art of Scientific Computing.* Cambridge University Press, 2nd, revised edition, 1997.

[37] Reynolds, J. H. *Multi-Criteria Assessment of Ecology Process Models Using Pareto Optimization.* PhD thesis, University of Washington, 1997.

[38] Reynolds, J. H., E. D. Ford. Multi-criteria assessment of ecological process models. *Ecology*, 80(2):538–553, 1999.

[39] Rudolph, G. Evolutionary search under partially ordered fitness sets. In *Proceedings of the International NAISO Congress on Information Science Innovations (ISI 2001)*, pages 818–822. ICSC Academic Press, 2001.

[40] SAS Institute. *SAS/IML User's Guide, Version 8*, 1999.

[41] Schoenauer M., Z. Michalewicz. Evolutionary computation. *Control Cybernetics*, 26(3):307–338, 1997.

[42] Srinivas N., K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.

[43] Syswerda, G. Uniform crossover in genetic algorithms. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 2–9. Morgan Kaufmann, 1989.

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

[45] Zitzler, E., L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 4(3):257–271, 1999.

Appendix A

MANUAL OF PARETO_EVOLVE

I now explain how to use the Evolutionary Algorithm (EA) software Pareto_Evolve. Pareto_Evolve is written in C language, and we used Microsoft Visual C++ to develop and implement it.

The user need two sets of codes to run Pareto_Evolve. One is `pareto_run`, which is to execute the EA and consists of four source files (`PARETO_main.c`, `PARETO_update.c`, `PARETO_breed.c` and `PARETO_misc.c`) and two header files (`pareto_evolve.h` and `pareto_userconst.h`). The other is `criteria`, which is for objective functions, i.e., the functions to be optimized in `pareto_run`. The user needs to create this set of codes. In Section A.1, I will show how criteria should be constructed.

As introduced in Section 2.4, Pareto_Evolve works as follows.

1. Generation is set at 0, and `pareto_run` randomly creates initial population; those created parameterizations are written to input file input.txt.

2. The data in input.txt are read by `pareto_run`, and then `criteria` is called to calculate their objective function values. With the input data, those calculated values are written to output file crit.out. Referring to the calculated objective function values, `pareto_run` assign a fitness value to each individual. Then it selects parents depending on the fitness values and applies crossover or mutation operators to these parents to produce offspring. Finally, the parameterizations of the offspring are written to input.txt. This is the end of the current generation.

3. The processes in a generation introduced above are repeated by `pareto_run` until the

generation number reaches to the maximum allowed number or the objective function values used in the optimization are satisfied.

A.1 The Set of Codes for Objective Functions

The set `criteria` of codes is to define the objective functions to optimize in `pareto_run` using a multi-objective EA. The following three procedures need to be included in the source files:

1. Read in the parameter values from input file `input.txt`; in the file, the parameter names are written at the first line, and the parameterizations start from the second line; each line has only one parameterization;
2. Using the read-in parameter values, calculate the objective function values;
3. Write the parameter names and criterion names at the first line in output file `crit.out`; from the second line, print the parameterization and the calculated objective function values for each individual per line.

In Appendix B, I introduce an example of `criteria`. It is for the objective functions of the ecological process model used in this thesis. In this example, `shoot_growth_crit.c` is the source file of `criteria`, and there are three header files. Two are ones used in `pareto_run`, `pareto_evolve.h` and `pareto_userconst.h`. One is `data.h`, which I created for the data used in `shoot_growth_crit.c`.

A.2 The Set of Codes for Pareto_Evolve

A search by the multi-objective EA is executed in `pareto_run`. As I mentioned above, `pareto_run` consists of four source files (`PARETO_main.c`, `PARETO_update.c`, `PARETO_breed.c` and `PARETO_misc.c`) and two header files (`pareto_evolve.h` and `pareto_userconst.h`). File `PARETO_main.c` is for essential processes in Pareto_Evolve. File `PARETO_update.c` is to assign fitness, and `PARETO_breed.c` is for selection, crossover and mutation. File `PARETO_misc.c` is for subroutines called in more than one files or `PARETO_main.c`. The constants to be determined by the user are defined in `pareto_userconst.h`. File `pareto_evolve.h` is for other fixed constants, all structures and some simple functions. Those codes are introduced in Appendix B.

Before the user starts search, it is necessary to determine several things in `PARETO_misc.c` and `pareto_userconst.h`. Next, I explain what the user needs to determine before starting the search and how those codes work.

A.2.1 Things to Be Set before the Search

Before starting search, the user needs to define several things in `pareto_userconst.h` and `PARETO_misc.c`.

In `pareto_userconst.h`, all of the following seven constants need to be defined by the user:

BINARY	If error measures are binary, BINARY = 1, and 0 if continuous;
NUMPARAMS	Number of parameters;
NUMCRITERIA	Number of objective functions;
POP_SIZE	Size of population;
GEN_NUM	Number of the maximum allowed generation;
MAX_DURATION	Number of the maximum allowed duration when elites can be copied;

GEN_TO_PRINT	Period of the generation number when the user wishes to see search results; e.g., if 100, then at generation 100, 200, ..., the objective function values are written to output file `crit.out100`, `crit.out200`, ..., respectively.

The following is a part of my example file `pareto_userconst.h` where those seven variables are defined.

```
*************************************************************************
#define BINARY 1   /* =1 if error measures are binary, =0 if continuous */

#define NUMPARAMS 7      /* number of parameters */
#define NUMCRITERIA 14   /* number of criteria used in assessment */

#define POP_SIZE 100   /* number of individuals per generation */
#define GEN_NUM 500    /* maximum allowed number of generations for
                          a search */

#define MAX_DURATION 50   /* maximum duration for elites */

#define GEN_TO_PRINT 500   /* how often results to be written out */
*************************************************************************
```

In `PARETO_misc.c`, the following three vectors need to be defined by the user:

pnames	Name of parameters; this consists of the number of NUMPARAMS elements;
psearch	Parameter search ranges; this consists of the number of PARAMS vectors, and each vector has three elements; the first, second, third elements for each parameter represent the minimum and maximum of the search range and its step size;
assessinfo	Objective target ranges or values; this consists of the number of NUMCRITERIA vectors, and each vector has two elements; if BINARY $= 1$, then the first and second elements represent the minimum and maximum of the range for an achieved criterion; if BINARY $= 0$, then the first element represents the target value to be desired to achieve, and the second element is discarded.

The following is a part of my example file `PARETO_misc.c`, where those three vectors are defined.

```
*************************************************************************
   /* Enter each parameter's name used in printing out the search list.
NAMES are LIMITED to 10 characters!! */

   char pnames[NUMPARAMS][11] = {"x1","x2","x3","x4","x5","x6","x7"};

   double psearch[NUMPARAMS][3] = {

   /* Min, Max, Min. Step Size for each parameter; each has 20 possible
values, so 20^5 combinations */
```

```
    {0.0, 0.4, 0.001},   /* search for param 1 */
    {0.0, 0.1, 0.001},   /* search for param 2 */
    {0.0, 0.1, 0.001},   /* search for param 3 */
    {0.0, 0.1, 0.001},   /* search for param 4 */
    {0.0, 1.0, 0.001},   /* search for param 5 */
    {0.0, 1.0, 0.001},   /* search for param 6 */
    {0.0, 1.0, 0.001}    /* search for param 7 */
  };


  double assessinfo[NUMCRITERIA][2] = {

  /* If BINARY = 1, then min, max Acceptable range for each criteria;
e.g., {5, 32} */

  /* If BINARY = 0, then target value, trash for each criteria;
e.g., {18,0} */

    {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21},
    {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21},
    {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21}, {-0.21,0.21}
  };
*************************************************************************
```

After setting those variables and vectors, the user can start search. I now explain how each routine in `pareto_run` works.

A.2.2 Main Function

File PARETO_main.c consists of the main function main, which calls functions evaluation and breed for essential processes of the EAs. In main, first, generation is set at 0, and function initial_input is called from PARETO_misc.c to create initial population. Then, the iteration process starts. The main routine in PARETO_update.c, evaluation is called to assign fitness to each population member. Next, the main routine in PARETO_breed.c, breed is called to generate offspring. Generation number is increased by 1, and evaluation and breed are kept calling until generation number reaches to the maximum allowed number or the objective function values used in the optimization are satisfied.

A.2.3 Functions for Selection

File PARETO_update.c is for routine evaluation, which is to assign fitness, and subroutines called only in evaluation. In evaluation, first, function read_input is called to read in parameter values and objective function values for the first individual in the current population. Then criteria_assess assesses its objective function values; if BINARY=1, 1 is assigned when an objective value is in its objective target range and 0 is assigned when it is outside of the range, and if BINARY=0, the absolute distance from the objective target value is calculated. Then, the parameter values, objective function values and assessment vectors of the first individual are added to the list.

Subroutine read_input and crit_assess are called for other individuals in the current population, but from the second individual, the individual is compared to the individuals in the list. If it is not dominated by the compared individual in the list, i.e., is codominant with or dominates the one in the list, it is added to the list before the compared individual. This is done by subroutine ordered_add_to_current_gen.

After all individuals are added to the lexicographically ordered list, assign_fitness assigns them fitness as defined in Section 2.4, and elites are selected by elitism as defined in

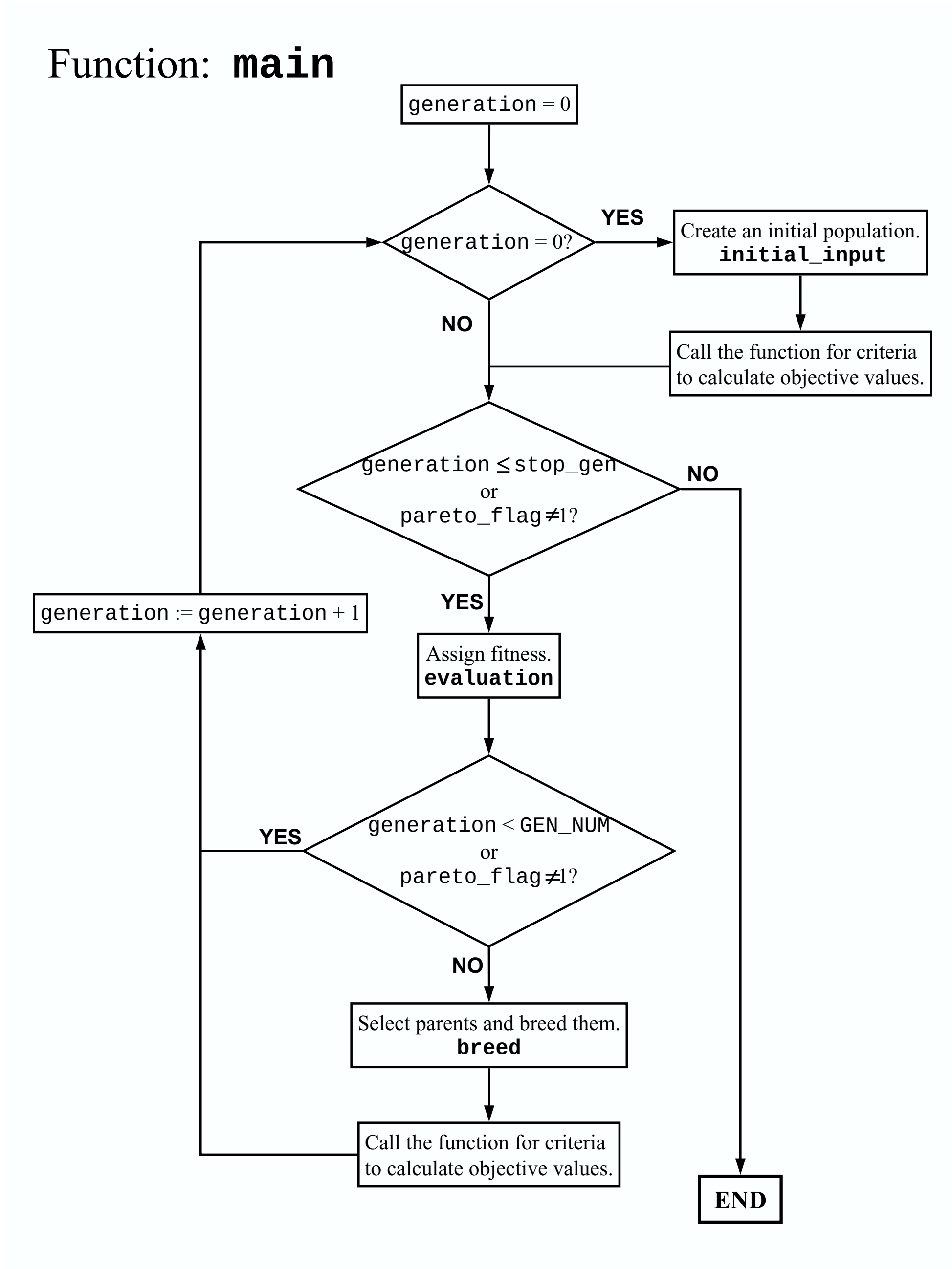


Figure A.1: Main flow of function main. The boldfaced word at the end of each box represents the function name. Constant stop_gen is the number of generation desired to stop the search, and GEN_NUM is the maximum allowed number of generation. If all of the criteria are achieved, pareto_flag is 1; otherwise 0. Function initial_input, evaluation and breed are defined in `PARETO_misc.c`, `PARETO_update.c` and `PARETO_breed.c`, respectively.

Section 8.3.

A.2.4 Functions for Breeding

File `PARETO_breed.c` is for routine breed, which is for selection, crossover and mutation, and subroutines called only in breed. First, parents are selected by par_selector, referring to the fitness values. A roulette introduced in Section 2.2.2 is created; then a random number at $[0, 1]$ is picked and the corresponding individual to the values on the roulette is chosen as a parent. These procedures are repeated for the number of parents the user needs, that is, POP_SIZE$-$ number of elites. Which parents will be crossed over or mutated is determined dynamically as I explained Section 8.3, and that is done by operator_selector.

Crossover and mutation is executed by crossover and mutator, respectively. Routine offspring_output defined in `PARETO_misc.c` writes the parameterizations of the produced offspring and elites to input file `input.txt`.

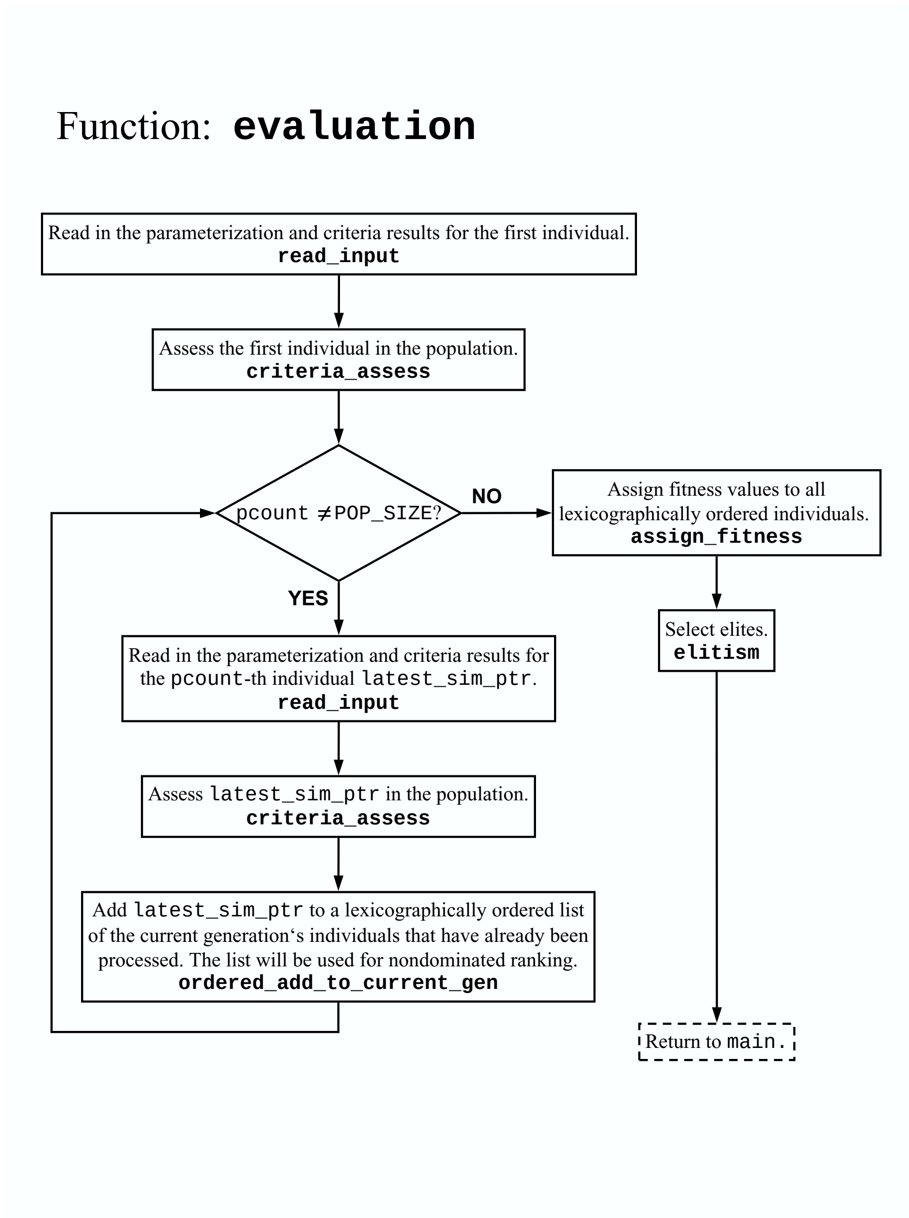


Figure A.2: Main flow of function evaluation called in function main (Figure A.1). The bold-faced word at the end of each box represents the function name. Variable pcount is to count the number of individuals that have been read in to order the population lexicographically. Individuals are ordered lexicographically ordered to assign fitness.

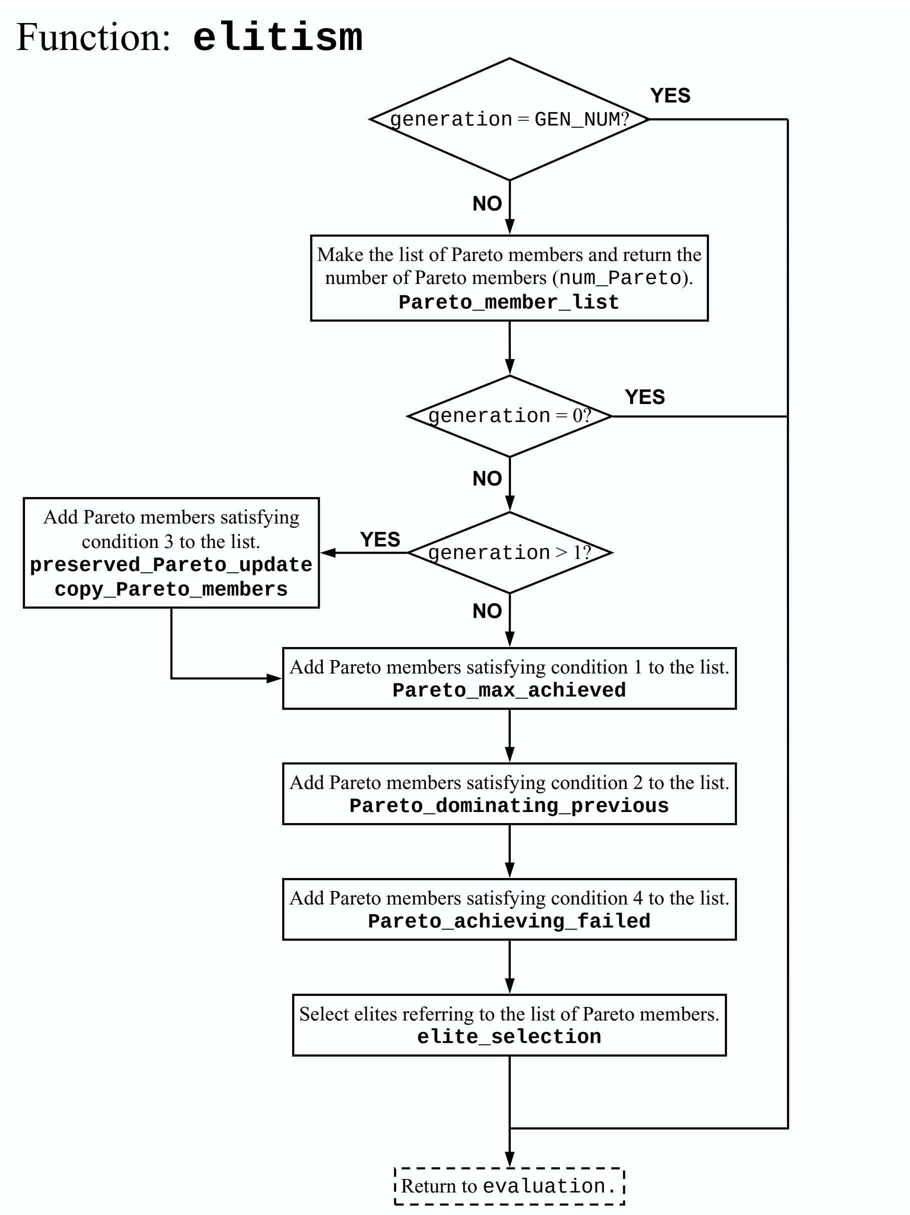


Figure A.3: Main flow of function elitism called in function evaluation (Figure A.2). Constant GEN_NUM is the maximum allowed number of generation.

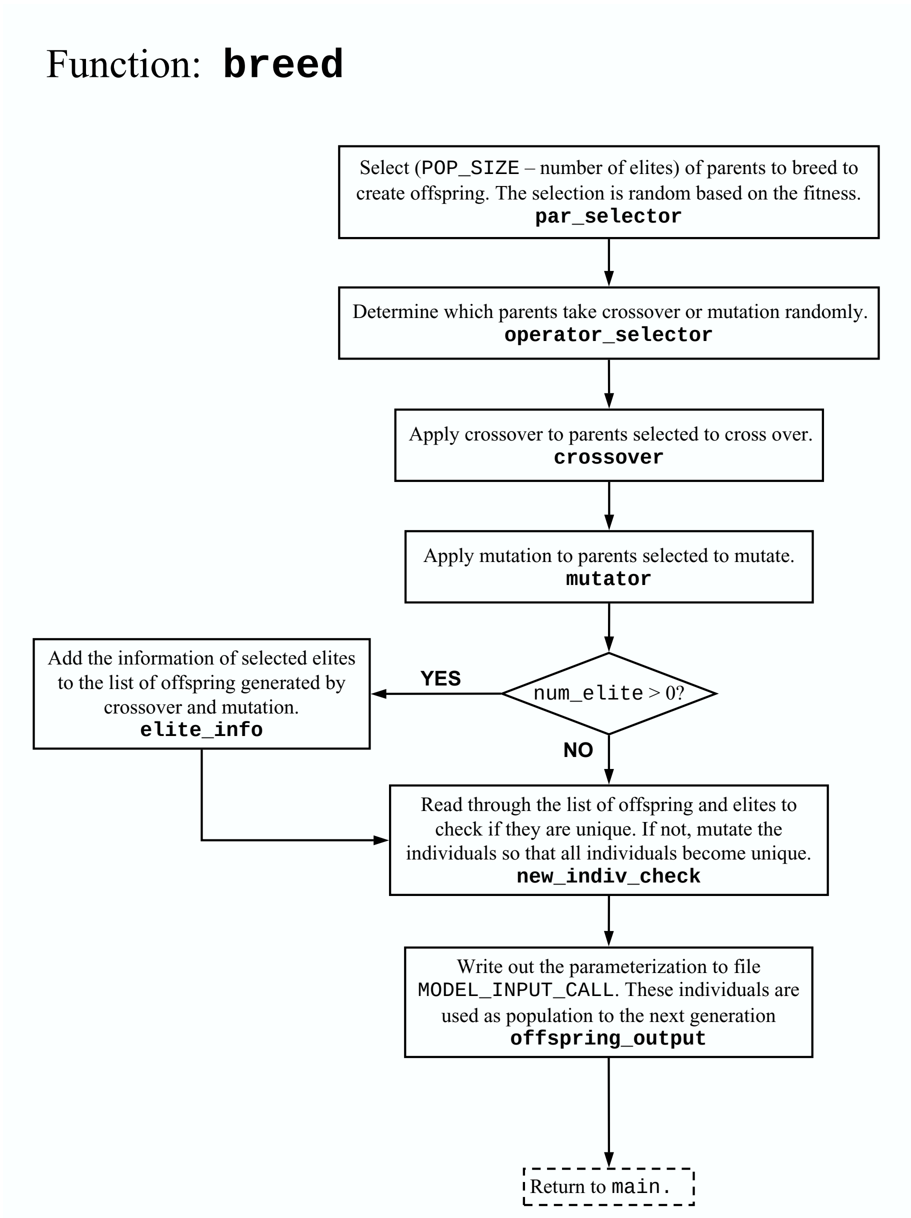


Figure A.4: Main flow of function breed called in function main (Figure A.1). Constant POP_NUM is the number of population. Whenever an elite is selected, num_elite is increased by one.