Automotive Engine Calibration with Experiment-Based Evolutionary Multi-objective Optimization

Hirotaka Kaji

Doctoral Thesis

Kyoto University
Graduate School of Informatics
Kyoto, Japan

August 2008

Abstract

The aim of this thesis is establishment of an overall framework of a novel control parameter optimization of automotive engine. Today, control parameters of an automotive engine have to be adjusted adequately and simultaneously to achieve plural criteria such as environmental emissions, fuel-consumption and engine torque. This process is called 'engine calibration'. Because many electronic control devices have been adopted for engine to satisfy these objectives, the complexity of engine calibration is increasing year to year. Recent progress in automatic control and instrumentation provides a smart environment called Hardware In the Loop Simulation (HILS) for engine calibration. In addition, Response Surface Methodology (RSM) based on statistical model is currently employed as the optimization method. Nevertheless, this approach is complicated by adequate model selection, precise model construction, and close model validation to confirm the precision of the model output.

To cope with these problems, we noticed experiment-based optimization via HILS environment based on Multi-objective Evolutionary Algorithms (MOEAs), that is expected to be a powerful optimization framework for real world problems such as engineering design, as another automatic calibration approach. In experiment-based optimization, the parameters of a real system are optimized directly by optimization techniques in real time through experimentation. In this thesis, this approach is called Experiment-Based Evolutionary Multi-objective Optimization (EBEMO) and it is proposed as a novel automatic engine calibration technique. This approach can release us from burdens of model selection, construction, and validation. When using this technique, calibration can be done immediately after specifications have been changed after optimization. Hence, EBEMO promises to be an effective approach to automatic engine calibration. However, since conventional MOEAs face several difficulties, it is not easy to apply it to real engines.

On the one hand, deterioration factors of the search performance of MOEAs in real environments have to be considered. For example, the observation noise of sensors included in output interferes with convergence of MOEAs. In addition, transient response by parameter switching also has similar harmful effects. Moreover, the periodicity of control inputs increase the complexity of the problems. On the other hand, the search time of MOEAs in real environments has to reduce because MOEAs require a tremendous number of evaluations. While we can obtain many measurements with

HILS, severe limitations in the number of fitness evaluations still exist because the real experiments need real-time evaluations. Therefore, it is difficult to obtain a set of Pareto optimal solutions in practical time with conventional MOEAs. Additionally, plural MOPs defined by plural operating conditions of map-based controllers has to be optimized.

In this thesis, to overcome the difficulties and to make EBEMO using the HILS environment feasible, five techniques are proposed. Each technique is developed through problem formulation, and their effectiveness are confirmed via numerical and real engine experiments. First, observation noise handling technique for MOEAs is considered. Because observation noise deteriorates the search ability of MOEAs, a memory-based fitness estimation method to exclude observation noise is introduced. Then, a crossover operator for periodic functions is proposed. Periodicity exists in engineering problems and leads to harmful effects on the performance of evolutionary algorithms. Moreover, the influence of transient response caused by parameter switching for dynamical systems is considered. In order to solve this problem, a solver of traveling salesman problems is used to determine the evaluation order of individuals. In addition, Pre-selection as acceleration method of MOEAs is proposed. In this technique, the generated offspring are pre-evaluated in the approximation model made by the search history, and then the promising offspring are evaluated in a real environment. Finally, parameterization of multi-objective optimization problems is considered. In engine calibration for maps, optimal control parameters have to be obtained at each operating condition such as engine speed and torque. This problem can be formulated in a form that needs to solve all of the plural multi-objective optimization problems defined by plural conditional variables. To solve this problem effectively, an interpolative initialization method is proposed. Through the real engine experiments, it was confirmed that EBEMO can achieve a practical search accuracy and time by using proposed techniques.

In conclusion, the contribution of EBEMO for engine calibration is discussed. Additionally, the directions for future work are outlined.

Acknowledgments

I would first like to express my deep gratitude and appreciation to my supervisor Professor Hajime Kita of the Academic Center for Computing and Media Studies (ACCMS) at Kyoto University for his guidance, advice, and encouragement throughout the course of this work.

I would also like to thank my advisors Professor Osamu Katai and Professor Tetsuro Sakai of the Graduate School of Informatics at Kyoto University for their excellent advice and suggestions.

I am the most grateful to Associate Professor Tetsutaro Uehara and Assistant Professor Mikihiko Mori of ACCMS for their support and useful discussions. I am also the most grateful to Assistant Professor Kokolo Ikeda of ACCMS for his deep discussions on evolutionary computations. It is thanks to all members of the Kita Laboratory of Graduate School of Informatics that I had such an exciting and productive laboratory experience.

I would like to thank Senior General Manager Masahito Suzuki, General Manager Kazuki Takahashi, and Group Leader Masashi Yamaguchi of the Research and Development Operation at Yamaha Motor Co., Ltd. for their understanding and heartful support. Special gratitude is also extended to Manager Shigeho Sakoda and engine calibration experts of Research and Development Operation for their support of engine experiments and very useful discussions on engine calibration.

Finally, I wish to express my sincere gratitude to my family, Ayako, Minami, and Koichiro who provide their continued support and love to me.

List of Publications

Journals

1. H. Kaji, "Proposal of a Multi-objective Genetic Algorithm for Noisy Fitness Functions," *Transactions of the Institute of System, Control and Information Engineers*, Vol. 18, No. 12, pp. 423–432, 2005. (in Japanese)

2. H. Kaji, and H. Kita, "A Crossover Operator of Genetic Algorithms for Periodic Function Optimization," *Transactions of the Society of Instrument and Control Engineers*, Vol. 43, No. 4, pp. 323–330, 2007. (in Japanese)

3. H. Kaji, and H. Kita, "Acceleration of Experiment-Based Evolutionary Multi-objective Optimization using Fitness Estimation," *IEEJ Transactions on Electronics, Information and Systems*, Vol. 128-C, No. 3, pp. 388–398, 2008. (in Japanese)

4. H. Kaji, and H. Kita, "Individual Evaluation Scheduling for Experiment-Based Evolutionary Multi-objective Optimization," *IEEJ Transactions on Electronics, Information and Systems*, Vol. 128-C, No. 6, pp. 986–996, 2008. (in Japanese)

International Conference Proceedings

1. H. Kaji, and H. Kita, "Individual Evaluation Scheduling for Experiment-Based Evolutionary Multi-objective Optimization," *4th International Conference on Evolutionary Multi-Criterion Optimization 2007 (EMO 2007)*, LNCS 4403, pp. 645–659, Springer, 2007.

2. H. Kaji, and H. Kita, "Acceleration of Experiment-Based Evolutionary Multi-objective Optimization using Fitness Estimation," *4th International Conference on Evolutionary Multi-Criterion Optimization 2007 (EMO 2007)*, LNCS 4403, pp. 818–831, Springer, 2007.

3. H. Kaji, and H. Kita, "Acceleration of Experiment-Based Evolutionary Multi-objective Optimization of Internal-Combustion Engine Controllers using Fitness

Estimation," *Proceedings of 2007 IEEE Congress on Evolutionary Computation (CEC 2007)*, pp. 1777–1784, 2007.

4. H. Kaji, K. Ikeda, and H. Kita, "Acceleration of Parametric Multi-objective Optimization by an Initialization Technique for Multi-objective Evolutionary Algorithms," *Proceedings of 2008 IEEE Congress on Evolutionary Computation (CEC 2008)*, pp. 2291–2297, 2008.

Domestic Conference Proceedings

1. H. Kaji, "Multi-objective Optimization of Noisy Fitness Function by Multi-objective Genetic Algorithm with Fitness Estimation Method," *Proceedings of the 66th National Convention of IPSJ*, 2L–5 (CD-ROM), 2004. (in Japanese)

2. H. Kaji, "Proposal of a Multi-objective Genetic Algorithm for Noisy Fitness Functions," *Proceedings of SICE Symposium on Systems and Information 2004*, pp. 79–84, 2004.

3. H. Kaji, and H. Kita, "A Crossover Method of Genetic Algorithms for Periodic Function Optimization," *Proceedings of 32nd SICE Symposium on Intelligent Systems*, pp. 157–162, 2005. (in Japanese)

4. H. Kaji, and H. Kita, "A Study of Evaluation Reduction Method of Multi-objective Genetic Algorithms," *Proceedings of EvoTech '06–1*, pp. 1–4, 2006. (in Japanese)

5. H. Kaji, and H. Kita, "Application of an Acceleration Method to Experiment-Based Evolutionary Multi-objective Optimization of Real Internal-Combustion Engine," *Proceedings of 34th SICE Symposium on Intelligent Systems*, pp. 155–160, 2007. (in Japanese)

6. H. Kaji, and H. Kita, "Acceleration of Experiment-Based Evolutionary Multi-objective Optimization of Internal-Combustion Engine Controllers using Fitness Estimation," *Proceedings of 2007 JSAE Annual Congress (Autumn)*, No. 118–07, 2007. (in Japanese)

7. H. Kaji, K. Ikeda, and H. Kita, "Proposal of Multi-objective Evolutionary Algorithms for Parametric Multi-objective Optimization Problems," *Proceedings of Evolutionary Computation Symposium 2007*, pp. 51–54, 2007. (in Japanese)

8. H. Kaji, K. Ikeda, and H. Kita, "An Initialization Method of Multi-objective Evolutionary Algorithms for Parametric Multi-objective Optimization Problems," *Proceedings of 20th SICE Symposium on Decentralized Autonomous Systems*, pp. 121–126, 2008. (in Japanese)

Others

1. H. Kaji, and H. Kita, "Experiment-Based Evolutionary Multi-objective Optimization of Internal Combustion Engines," *Yamaha Technical Review*, Vol. 43, pp. 90–99, 2007. (in Japanese)

2. K. Ikeda, H. Kaji, and H. Kita, "Engineering Application of Multi-objective Evolutionary Computations," *Kyoto University ICT Innovation 2008*, Poster, 2008. (in Japanese)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Over 100 years ago, two German engineers, G. Daimler and K. Benz, developed spark ignition gasoline engine that had almost same characteristic of modern engines. In this era, although steam engines were the most popular power source and electric vehicles had already been developed, they were gradually replaced by gasoline engines. The performance of gasoline engine had improved year by year and the main stage of car development moved from Germany to the United States. After legislation of Clean Air Act of 1970 (Muskie Act), severe environmental emission regulations were introduced to solve the air pollution problem, especially in the United States. To maximize the performance of automotive engines and to satisfy the regulations, mechanical control systems such as conventional carburetors and distributors were replaced by electronic control devices. Automotive manufacturers developed many clean engine technologies such as the fuel-injector, and catalytic converter. In response to ever stricter environmental emission regulations in North America, Europe, and Japan, electronic control systems significantly impact automotive development due to the need to enhance performance and comply with the restrictions.

Today, design of control systems for modern automotive gasoline and diesel engines is one of the most important engine development processes. Control management is implemented in onboard computers called Electronic Control Units (ECUs). The primary purpose of ECUs is to lower environmental exhaust emissions (carbon monoxide (CO), hydrocarbons (HC), nitrogen oxides (NO_x)) and minimize fuel-consumption thereby addressing environmental and global warming issues. On the one hand the environment is a main concern of manufacturers and consumers, on the other hand the maximization of engine torque and drivability are attractive features for consumers. To achieve these countering requirements, many electronic control devices such as fuel-injectors, variable valve timing, are mounted to current engines, and the parameters of the ECUs have to be designed adequately. Hence, these evaluation criteria have to be balanced simultaneously and precisely.

Since the 1980s, application of control theory and modeling have increased in the development of automotive controllers. For instance, closed-loop air-fuel ratio control for three-way catalytic converter (TWC) is one of promising applications that had been introduced in TWC. Yet, because internal-combustion engines have non-linearity and complexity, a feedforward control called map-based control is still the main approach used in the development of engine control systems. In the map-based control, electronic devices are controlled by stored parameters that correspond to operating points on the maps of an ECU. Control parameters on the maps are determined by pre-experimentation on each operating point. Each point is defined by input variables such as engine speed and load torque in the engine control system [15, 39, 40, 63]. This process is called 'engine calibration'. Since aforementioned evaluation criteria have trade-offs, engine calibration is multi-criterion evaluation, which in turn makes it very time-consuming process in the engine development. To make the process of engine calibration more efficient, automatic design based on multi-objective optimization [13, 96] is needed to replace the conventional manual calibration.

Simulation-based optimization using a precise theoretical model is one candidate for automatic design [45, 64, 103]. This approach enables us to calibrate control parameters offline by using 'virtual engines' in place of real engines. Computational Fluid Dynamics (CFD) is used to represent the virtual engines. This approach has been realized because of recent progress in computer performance and parallelization technique such as PC cluster [108].

Response Surface Methodology (RSM) [74], a statistical offline optimization technique, is currently employed for engine calibrations. Control parameters are calibrated based on simple statistical models called response surface through few experiments designed by Design of Experiments (DoE) [23, 41].

Currently, huge progress has been made in automatic control and instrumentation. Real-time simulation provides a smart environment called Hardware In the Loop Simulation (HILS) for engine calibration through real experiments. This engine HILS environment is composed of a real engine and an engine test-bench that simulates vehicle-running conditions using a computer controlled ultra-low inertia dynamometer. Significant advancement and progress has been made in the past few years allowing RSM combined with an engine HILS bench to become the main stream in automatic engine calibrations [88, 89].

1.2 Approach

Many numerical optimization methods such as gradient methods can be applied to the theoretical and statistical engine models and can obtain optimal control parameters offline. However, significant effort and time is needed to construct precise engine models.

Despite the fact that in theoretical models parts of flow dynamics, thermodynamics and mechanical dynamics of engine are described, unknown parameters exist that have to be estimated through real experiments. Moreover, to construct a physical and chemical model of an internal-combustion engine in detail requires a lot of effort, and is not cost effective. Additionally, the engine model based on CFDs requires substantial simulation time in order to solve partial differential equations.

Whereas, in statistical models, although few measurement data that construct approximation model are determined by DoE, a lot of measurement data are needed to verify the precision of the model. Additionally, model selection is a critical and time-consuming process that influences the optimization result. Furthermore, these engine models must be re-constructed because engine specifications are frequently changed in the early stage of development.

To cope with the aforementioned problems, we notice online optimization, that is, experiment-based optimization via HILS environment as another automatic calibration approach overcomes the shortcomings of using statistical models and DoE. In this thesis, the term 'experiment-based optimization' is used to mean "the parameters of a real system are optimized directly by optimization techniques in real time through experiments". This approach can release us from burdens of adequate model selection, precise model construction, and close model validation. When using this technique calibration can be done immediately after specifications have been changed after optimization.

We paid special attention to Multi-objective Evolutionary Algorithms (MOEAs) that is a multi-objective optimization method based on Evolutionary Algorithms (EAs) as an experiment-based optimization method. MOEAs are optimization methods that make the best use of the features of multipoint search in EAs, and can obtain the set of Pareto optimal solutions, the rational solutions of multi-objective optimization problems, at one time [14, 18]. Multi-objective optimization based on MOEAs is called Evolutionary Multi-objective Optimization (EMO). Currently, MOEAs are widely applied to real world problems through computer simulations. We call this approach Experiment-Based Evolutionary Multi-objective Optimization (EBEMO) in this thesis, and we are sure it is a promising field of applications.

Although EAs were originally suitable for experiment-based optimization [87, 97], this no longer holds true because automotive engine calibration through experiments includes several problems. Since optimization through experiment is severely limited by the condition of limited evaluation time and fluctuation of observation, we have to develop methodologies that overcome these problems for EBEMO. Two of these are called 'expensive evaluation cost problem' and 'uncertainty environment problem' respectively, and are challenging and attractive topics in recent research into EAs. Moreover, the periodicity of control inputs and the plural operating conditions of map-based controllers have to be considered from a viewpoint of engine controls. Therefore, in order to construct a practical automatic engine calibration technique based on EBEMO,

these difficulties have to be solved one at a time.

1.3 Outline of Thesis

Through this thesis, the fundamental methodology of EBEMO for automotive engine calibration as a novel overall approach will be established. This thesis is composed by the following chapters after this introductory chapter.

Chapter 2 first explains the details of automotive engine control and its calibration techniques. A description of the foundation of map-based control that is widely used for internal combustion engine control in automotive industries is given. For calibration of the maps, the concept of HILS, a key technology of experiment-based optimization, is also introduced.

Chapter 3 introduces the basic scheme of EMO. Mathematical definitions of Pareto optimal solution and the foundation of MOEAs are denoted briefly. Moreover, the aims and the difficulties of EBEMO are discussed.

In Chapter 4, an observation noise handling technique for MOEAs is discussed. In experiment-based optimization, observation noise added to sensor data is a problem in precise optimization of real world applications. To solve this problem, a memory-based fitness estimation method is used to estimate true fitness value. The effectiveness of the proposed method is confirmed through numerical experiments and a real engine experiment.

In Chapter 5, a periodicity of objective function that exists in engineering problems such as mechanical systems is discussed. Periodic landscape of objective function leads to harmful effects on performance of evolutionary algorithms, because local optimal solutions are generated due to the periodicity. A crossover operator for periodic functions is proposed in order to overcome this problem. It is demonstrated that the proposed operator has good performance to not only on periodic functions but also on non-periodic functions in bounded search spaces.

In Chapter 6, an influence of transient response that is caused by parameter switching for dynamical systems during optimization is considered. To cope with this problem, Individual Evaluation Scheduling (IET) is proposed. In this technique, a solver of traveling salesman problems is used to determine the evaluation order of individuals. Formal numerical experiment and experiment using a real engine are executed to demonstrate the effectiveness of IET.

In Chapter 7, a reduction of the number of evaluations is considered. It is an intriguing research topic in evolutionary computation with simulation-based optimizations that have expensive calculation costs such as CFDs as main applications. Preselection is proposed as an acceleration method of MOEAs for experiment-based optimization. In this technique, the generated offspring are evaluated in advance with the approximation model made from the search history, and then only offspring promising

in the previous evaluation are applied to a real environment. It is confirmed that the proposed method can effectively reduce the number of evaluation in both numerical experiments and real engine experiments.

In Chapter 8, a parameterization of multi-objective optimization problems is considered. In engine calibration for maps, optimal control parameters have to be obtained at each operating condition such as engine speed. First, a new problem class called Parametric Multi-objective Optimization Problems (PMOPs) is introduced to handle this problem. This is a problem of solving all of plural multi-objective optimization problems defined by plural conditional variables. Next, interpolative initialization method is proposed to solve PMOPs effectively. This method demonstrates good performance when applied to a simple numerical test function.

In Chapter 9, the over all conclusions of this thesis are presented. Additionally, future works and directions of EBEMO are discussed.

Chapter 2

Automotive Engine Calibration

2.1 Outline

In this chapter, first fundamentals of automotive engine control are briefly introduced. After that, an outline of automatic engine calibration is described. Currently, two types of techniques are used in automotive industries instead of manual calibration. One is theoretical simulation-based optimization that does not use real engines, the other is response surface methodology that constructs empirical models via engine testing. Additionally, Hardware In the Loop Simulation (HILS) environment that is a newer automatic measurement environment is introduced. Recent progress in automatic engine calibration is based on RSM with an engine HILS environment.

2.2 Automotive Engine Control

In automotive engine control, feedback and feedforward controllers are used to satisfy the performances and restrictions. For example, feedback control of air-fuel ratio for a three-way catalytic converter is used to meet with environmental legislation. A feedforward control called 'map-based' or 'lookup-table' is widely used for engine control systems to handle nonlinearity and complexity of internal-combustion engines. As shown in Fig. 2.1, electronic devices such as fuel injectors and spark ignitions are controlled to desired values calculated by maps. For example, injector map outputs injection timing and injector energizing time to control amount of fuel injected. The fuel injector driver energizes the injectors to inject the desired amount of fuel precisely at the optimal timing.

In a map-based control, the maps implemented in ECUs are represented by a grid of engine operating conditions defined by input variables such as engine speed and load torque. These desired values are stored as parameters that correspond to operating points on the grid, and calculated by linear interpolation of these values for the input variables that changes continuously. Figure 2.2 shows a conceptual diagram of map calculation. A two-dimensional map, that is, two-inputs one-output function is usually

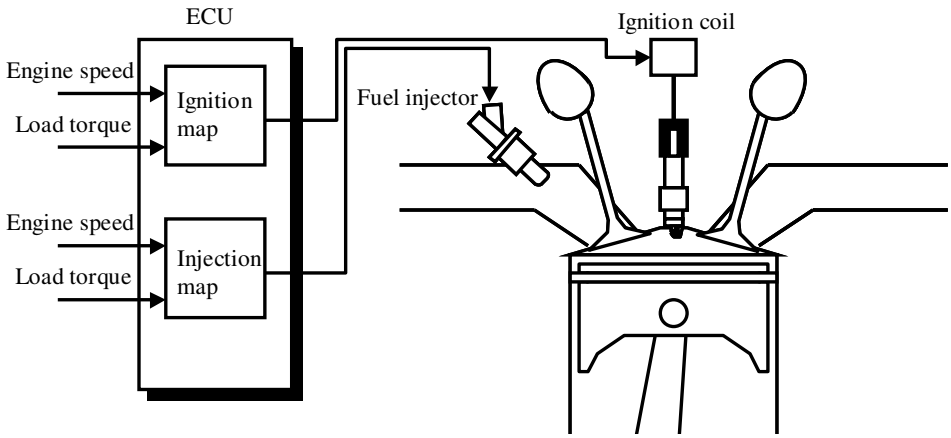


Figure 2.1: Engine control system.

used. A map that has $n \times m$ grids is represented by $l = n \times m$ tuples $(u_1^i, u_2^j, x_{\mathrm{d}}^{(i,j)})$, where $x_{\mathrm{d}}^{(i,j)}$ is the output value that corresponds the grid (u_1^i, u_2^j), $i = 1, \ldots, n$, $j = 1, \ldots, m$. The representative operating points $u_1^1, \ldots, u_1^n$ and $u_2^1, \ldots, u_2^m$ are called 'label'. The desired ignition timing $\tilde{x}_{\mathrm{d}}$ at the current operating condition (engine speed, torque) $= (\tilde{u}_1, \tilde{u}_2)$ is calculated by following linear interpolation [5, 6, 78]:

$$\begin{aligned}\tilde{x}_{\mathrm{d}} := (\tilde{u}_2 - u_2^j)\left((\tilde{u}_1 - u_1^i)x_{\mathrm{d}}^{(i,j)} + (u_1^{i+1} - \tilde{u}_1)x_{\mathrm{d}}^{(i+1,j)}\right) \\ + (u_2^{j+1} - \tilde{u}_2)\left((\tilde{u}_1 - u_1^i)x_{\mathrm{d}}^{(i,j+1)} + (u_1^{i+1} - \tilde{u}_1)x_{\mathrm{d}}^{(i+1,j+1)}\right),\end{aligned}$$

where $u_1^i \leq \tilde{u}_1 < u_1^{i+1}$ and $u_2^j \leq \tilde{u}_2 < u_2^{j+1}$ are satisfied. Therefore, to develop adequate maps, we have to find optimal control parameters by experimentation on each operating point, and integrate them as maps. This optimization process is called 'engine calibration' [15, 39, 40, 63].

2.3 Automatic Engine Calibration

The basic framework of automotive engine calibration systems that combined a calibration computer with ECUs have been already developed in-1980s. For example, Watanabe and Tümer have proposed an automotive engine calibration system for spark ignition timing to satisfy low environmental emissions and low fuel-consumption [109]. Because a trade-off relationship exists between these evaluation criteria, engine calibration becomes a multi-criterion evaluation. The number of control parameters of previous engine calibrations was significantly smaller than current ECUs as shown in Watanabe's study, and the manual calibration using full-factorial testing or engineer's heuristic law was standard. However, the number of the devices and the complexity

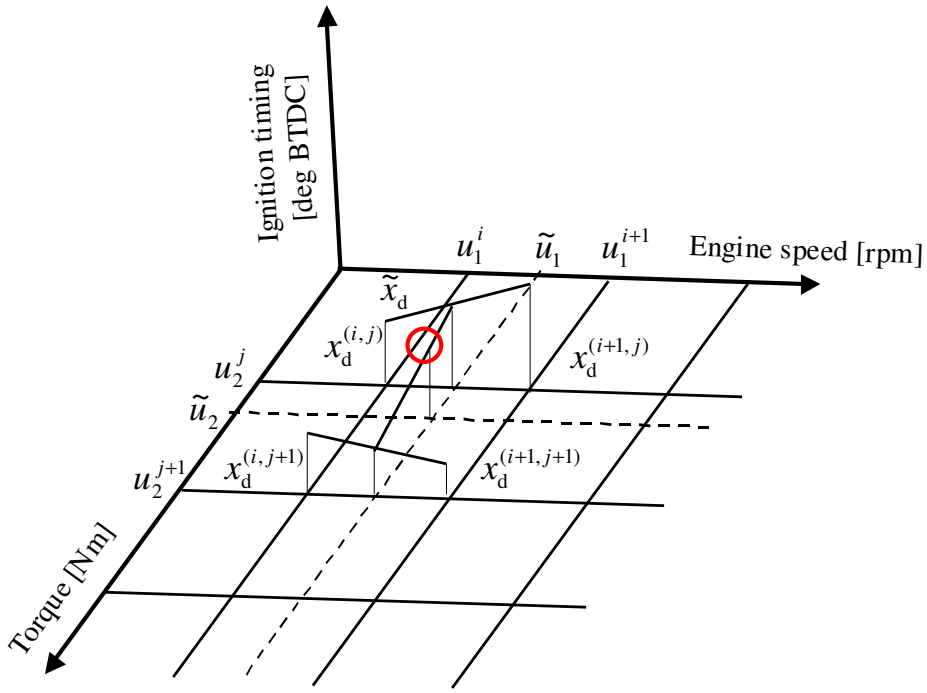


Figure 2.2: Calculation of map.

of the systems gradually increased, and engine calibration has become a very time-consuming process in the engine development. In order to accomplish this process more efficiently, automatic design based on multi-objective optimization is needed to replace the conventional manual calibration.

2.3.1 Simulation-Based Optimization

Precise theoretical simulation-based optimization is one candidate of automatic design. Instead of real engines, 'virtual engines' represented by computer code such as Computational Fluid Dynamics (CFD) is used for optimization. Recent progress in computer performance and parallelization techniques such as PC cluster have been realized as a practical approach. Moreover, various optimization techniques such as gradient methods and genetic algorithms can be applied to theoretical models. For instance, Hiroyasu et al. have applied multi-objective evolutionary algorithms to phenomenological diesel engine models called HIDECS [45]. Their objectives are minimization of fuel-consumption, NO_x, and Soot (Carbon particulate matter), and decision variables are parameters of multiple injection strategy. Similarly, Kim et al. have applied micro-GA for a diesel engine simulation using KIVA-3V code [64]. Srinivasan and Tanner also have applied a gradient method for same type of simulation [103]. They have optimized multiple injection strategy.

This approach enables the calibration of control parameters offline without real

engines. It is an effective approach because optimization can be executed even when no engine is available. However, the following drawbacks exist in this approach:

- To construct a physical and chemical model of an internal-combustion engine in detail requires a lot of effort, and it is not cost effective.
- Unknown parameters exist and eventually have to be estimated via the real engine experiments. Otherwise, the optimization results might only be used as the information about qualitative tendencies.
- The fluid and thermal dynamics of the engine model need enormous simulation time due to need to solve complex numerical calculation such as partial differential equations.

2.3.2 Response Surface Methodology

Response Surface Methodology (RSM) is an offline optimization technique based on Design of Experiments (DoE) and statistical approximation models that can reduce the large evaluation burden of engineering designs [74]. Approximation function of relationship between decision variables and objective function value is called 'response surface'. In RSM, DoE methods such as Latin Hypercube Sampling (LHS), Center Composite Design (CCD) and D-Optimal Design are employed to determine evaluated solutions in decision variable space. Additionally, polynomials, artificial neural networks and radial basis functions are usually used to construct response surfaces.

In automotive industries, the term 'Design of Experiments' currently means the entire process including design planning, the modeling and the optimization, although the original the term meant only design planning. Moreover, this calibration process is also called 'Model-Based Calibration'. In this thesis, the term 'Response Surface Methodology' is used with the same meaning as the aforementioned terms.

The typical engine calibration process by using RSM shown in Fig. 2.3 is as follows:

1. Set an operating point of the engine determined by an engine speed and a load torque.
2. Determine a set of evaluation control parameters based on DoE.
3. Measure the engine output using the control parameters.
4. Build the response surfaces of the engine output from the measured data using approximation modeling techniques.
5. Optimize the response surfaces offline by the control parameters, e.g., use generic mathematical (multi-objective) optimization method.

6. Verify the performance of the optimum control parameters in the real engine, and then return to Step 1) if necessary.

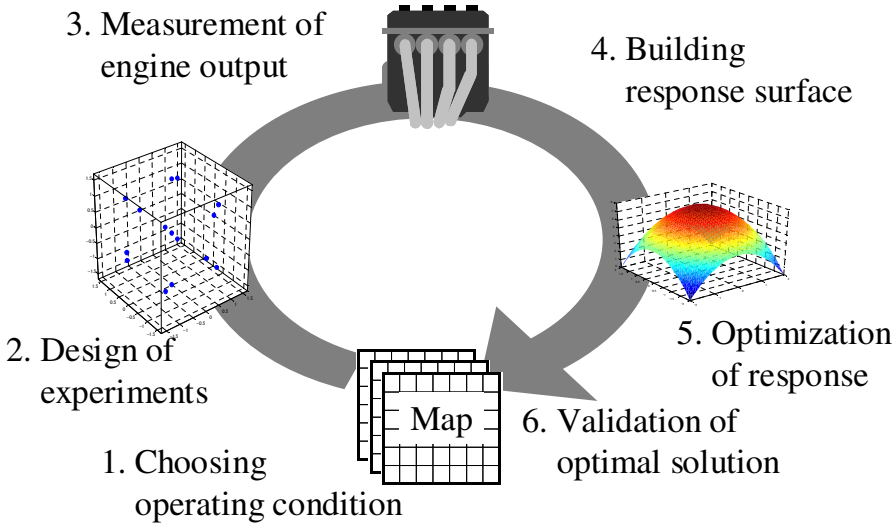


Figure 2.3: The procedure of response surface methodology.

Alonso et al., for example, have applied genetic algorithms for diesel engine models constructed by artificial neural networks [1]. This type of application can be seen in many previous works [23, 41, 88, 89]. They have optimized air mass, Exhaust Gas Recirculation (EGR), injection pressure, and multi injection strategy to minimize the weighted sum function of fuel consumption and exhaust emissions.

RSM can reduce the number of tests, and improve the efficiency of calibration process. However, the following drawbacks exist in this method:

- Re-evaluation of the parameters and re-construction of the response surfaces is necessary when the specifications of the tested engine are changed.
- The model selection and the design of experiments must be re-examined when new electronic control devices and their parameters are added.
- There is a possibility for the occurrence of a large modeling error. The influence on the calibration cannot be neglected when the complex characteristics of an engine are approximated by straightforward functions such as a second-order polynomial.

2.3.3 Hardware In the Loop Simulation

In recent automotive development, we can use a smart environment called Hardware In the Loop Simulation (HILS) for automated measurement. HILS is a technique

for accurately simulating a whole system by synchronizing a simulator with a real machine to evaluate the performance of a large-scale system. Although the term 'HILS' usually means an ECU verification process with real time engine/vehicle simulators in automotive control system developments, this term is used in a wider meaning in this thesis.

Figures 2.4 depicts an engine test-bench as a HILS environment. The engine test-bench shown in Fig. 2.5 consists of an ultra-low inertia dynamometer and a dynamo controlling computer having I/O interfaces. The ultra-low inertia dynamometer shown in Fig. 2.6 is connected with the crankshaft of the engine and controls load torque in real time. The dynamo controlling computer implements transmission and vehicle models to evaluate the engine in a condition almost identical to that of real cars. Moreover, instruments such as an exhaust gas analyzer, a fuel flow meter and a combustion analyzer are connected to measure the engine performances.

A calibration PC connects with the ECU which controls electronic devices. The control parameters on the maps can be changed automatically by the PC's software or manually by the operator. The calibration PC monitors the outputs from the engine test-bench, the instruments, and the ECU. Therefore, the calibration PC can handle this environment including a real machine in a manner similar to a simulation, and can use mathematical optimization methods. The HILS environment that introduced

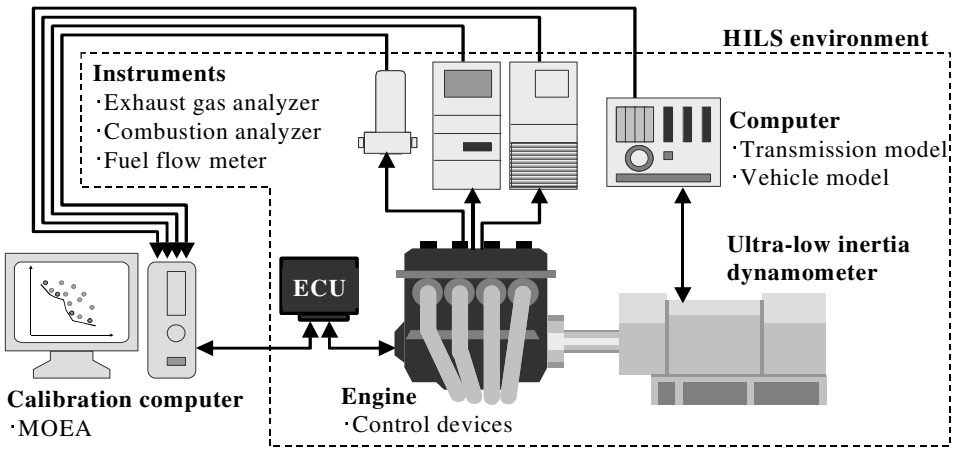


Figure 2.4: Block diagram of the engine HILS environment and the calibration PC.

RSM as optimization method is used as an automatic engine calibration system, and this approach has made significant progress and advancement over the past few years. However, we do not feel that RSM sufficiently takes full advantage HILS ability to take a lot of samples automatically, because RSM is usually adopted for the problems that cannot take many samples. We believe that a new methodology that brings out the best performance in HILS is necessary.

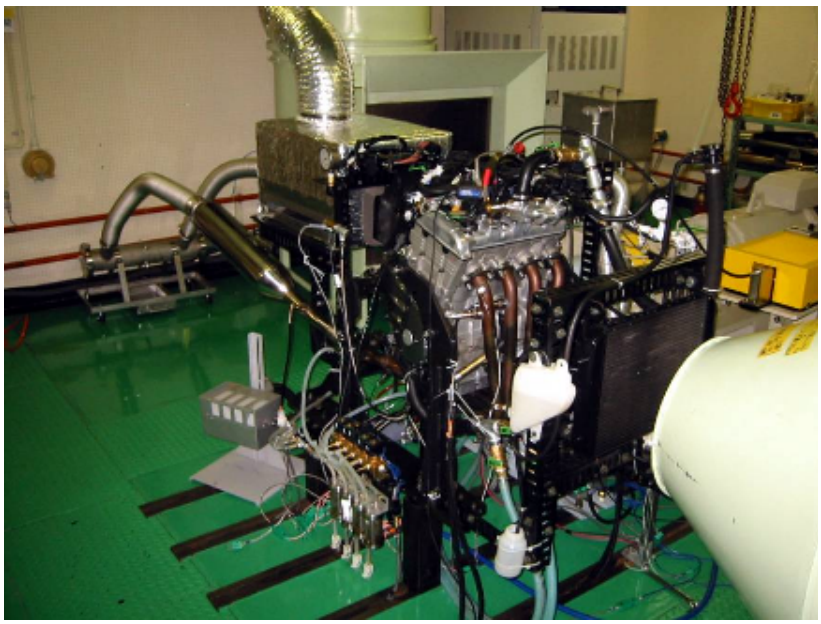


Figure 2.5: The engine test-bench.

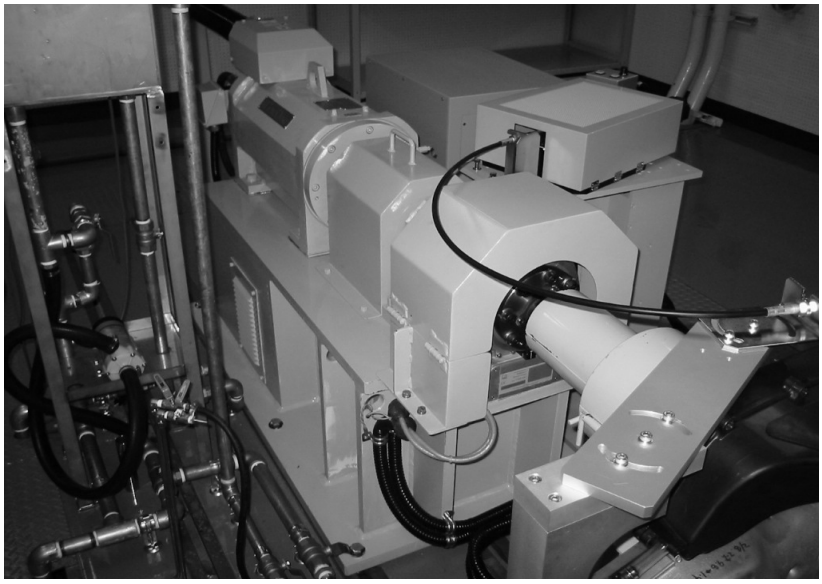


Figure 2.6: The ultra-low inertia dynamometer.

2.4 Summary

In this chapter, the brief introduction of automotive engine control and engine calibration was described. Although automotive engine calibration is one of the most important development processes, current engine calibration is too complex to solve manually because of many control parameters and performance requirements. Currently, we can choose two methods, theoretical simulation-based optimization and Response Surface

Methodology (RSM). In fact, recent automatic engine calibration systems widely have adopted RSM combined with Hardware In the Loop Simulation (HILS) environments. The drawbacks of theoretical simulation and RSM and necessity of new methodology for HILS environment were emphasized.

Chapter 3

Evolutionary Multi-objective Optimization

3.1 Outline

When we consider what the best internal-combustion engine is, the engine should be evaluated from various viewpoints; maximum horse power, maximum torque, minimum amount of exhaust emissions, fuel-consumption, response, sound, noise, and so on. Let us consider the engine torque and the fuel-consumption as two engine performances. An engine under these criteria should achieve large torque and low fuel-consumption simultaneously. However, to obtain large torque, engine needs more fuel. Whereas to achieve low fuel-consumption, engine needs to suppress torque. Hence, the offsetting performances should be treated as multidimensional variables, rather than single scalar ones, in the optimization process.

In this chapter, first, definition of Multi-objective Optimization Problems (MOPs) and Pareto optimal solution as a theory of handling such multiple evaluation criteria are introduced. Next, a brief introduction of Multi-objective Evolutionary Algorithms (MOEAs) is explained. Moreover, the motivation and the difficulties of Experiment-Based Evolutionary Multi-objective Optimization (EBEMO) in applying optimization through experiment such as HILS are discussed.

3.2 Multi-objective Optimization

3.2.1 Problem Formulation

Multi-objective Optimization Problem (MOP) is one of the problem class that has plural objectives. In this chapter, minimization problems are considered. Without lost of generality, every maximization problems can be transformed

$$\max_{\boldsymbol{x}} f(\boldsymbol{x})$$

into minimization problem

$$\min_{\boldsymbol{x}} -f(\boldsymbol{x}),$$

where f is an objective function, and $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_n]^{\mathrm{T}}$ is n-dimensional decision variable vector. In the following, decision variable vector is called 'solution'.

An aim of MOPs is to optimize objective functions by cooperating with each other as much as possible. MOPs are generally defined as the search for $\boldsymbol{x}$ to minimize competing m-objective functions

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x})\ f_2(\boldsymbol{x})\ \cdots\ f_m(\boldsymbol{x})]^{\mathrm{T}}$$

under p inequation constraints

$$\boldsymbol{g}(\boldsymbol{x}) := [g_1(\boldsymbol{x})\ g_2(\boldsymbol{x})\ \cdots\ g_p(\boldsymbol{x})]^{\mathrm{T}} \leq \boldsymbol{0}.$$

Because the objective functions compete with each other, the unique optimal solution for all the objective functions may not be obtained. Hence, Pareto optimal solution is introduced as a solution concept in place of the unique optimal solution. The concept of Pareto optimal solution, rational solutions of MOPs, was introduced by Vilfredo Pareto who was an economist at the turn of the 19th century and into the 20th century. Pareto optimal solution is defined by dominance comparison of solution in MOPs. The definition of dominance comparison is shown below:

Definition 3.1. (Pareto Dominance)

Let $\boldsymbol{x}^1, \boldsymbol{x}^2 \in X$, where $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_n]^{\mathrm{T}}$. $\boldsymbol{x}^1$ is said to dominate $\boldsymbol{x}^2$ if and only if $\boldsymbol{f}(\boldsymbol{x}^1)$ is partially less than $\boldsymbol{f}(\boldsymbol{x}^2)$, i. e. ,

$$\forall i \in \{1, 2, \ldots, m\},\ f_i(\boldsymbol{x}^1) \leq f_i(\boldsymbol{x}^2)\ \wedge\ \exists i \in \{1, 2, \ldots, m\},\ f_i(\boldsymbol{x}^1) < f_i(\boldsymbol{x}^2).$$

If $\boldsymbol{x}^1$ dominates $\boldsymbol{x}^2$, $\boldsymbol{x}^1$ is a more suitable solution than $\boldsymbol{x}^2$.

Next, the definition of Pareto optimal solution based on the dominance comparison is shown below:

Definition 3.2. (Pareto Optimal Solution)

Let a solution $\boldsymbol{x}^* \in X$. $\boldsymbol{x}^*$ is said to be a Pareto optimal solution, if there exists no other feasible $\boldsymbol{x} \in X$ that dominates $\boldsymbol{x}^*$.

Pareto optimal solution is also called 'non-dominated solution'.

Weakly Pareto optimal solution that is relaxation concept of Pareto optimal solution is introduced as follows:

Definition 3.3. (Weakly Pareto Optimal Solution)

Let a solution $\boldsymbol{x}^* \in X$. $\boldsymbol{x}^*$ is said to be a weakly Pareto optimal solution, if there exists no $\boldsymbol{x} \in X$ such that $\forall i \in \{1, 2, \ldots, m\},\ f_i(\boldsymbol{x}) < f_i(\boldsymbol{x}^*)$.

The purpose of MOPs is searching the set of Pareto optimal solutions, that is, Pareto optimal set described below:

Definition 3.4. (Pareto Optimal Set)

For a given MOP, the Pareto optimal set X^* is defined as:

$$X^* := \{\boldsymbol{x} \in X |\ \nexists \boldsymbol{x}^* \in X,\ \boldsymbol{x}^* \textit{ dominates } \boldsymbol{x}\}$$

Figure 3.1 shows an example of Pareto optimal set of a two-objective optimization problem. The left hand side of figure indicates objective function space and the right hand side of figure depicts decision variable space. In the decision variable space, Pareto optimal set is shown by solid line. The curve composed by the Pareto optimal set in the objective space is generally called 'Pareto frontier'. Similarly, the weakly Pareto optimal set in the objective space is called 'weakly Pareto frontier'.

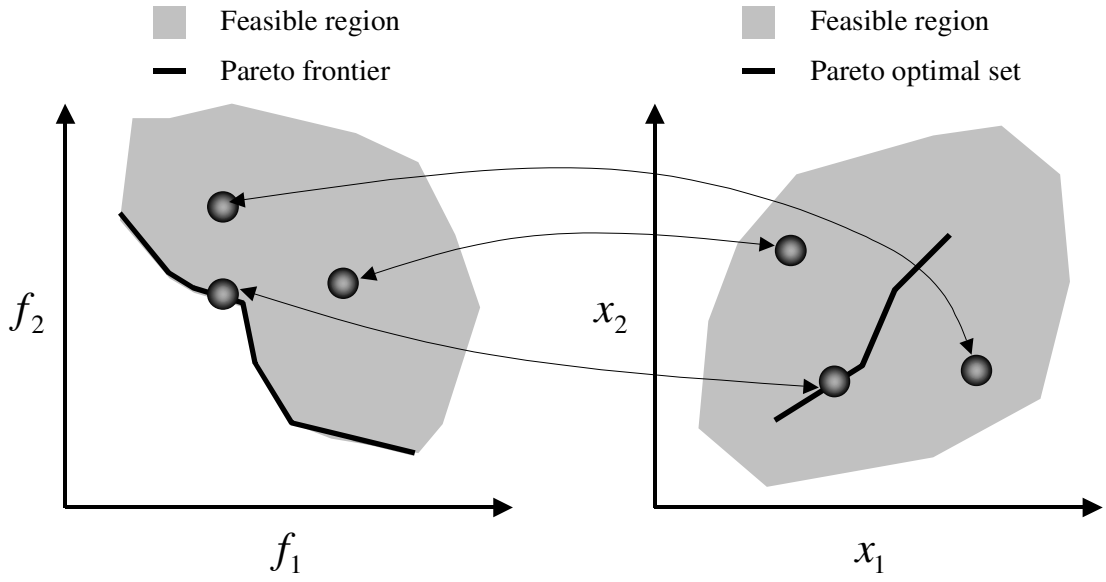


Figure 3.1: The conceptual diagram of Pareto optimal set.

3.2.2 Scalarization Method

Scalarization is the most common strategy to obtain Pareto optimal solutions. In scalarization methods, MOPs are characterized to Single-objective Optimization Prob-

lems (SOPs) and are solved by appropriate single-objective optimization methods. Among the many ways of scalarization methods, two of the methods are explained in the following:

Weighted Sum Method

In weighted sum method, objective function vector $\boldsymbol{f}(\boldsymbol{x})$ is transformed into a scalar objective function $F(\boldsymbol{x})$ as follow:

$$\min_{\boldsymbol{x}} F(\boldsymbol{f}(\boldsymbol{x}))$$
$$F(\boldsymbol{f}(\boldsymbol{x})) = \sum_{i=1}^{m} w_i f_i(\boldsymbol{x}),\ w_i \geq 0,\ \sum_{i=1}^{m} w_i = 1,$$

where w_i is weight of i th objective function. Figure 3.2 shows a conceptual diagram of weighted sum method. This method can find various Pareto optimal solutions by sweep of w_i. However, this method have a drawback, that is, the solutions in concave Pareto frontier are not obtained.

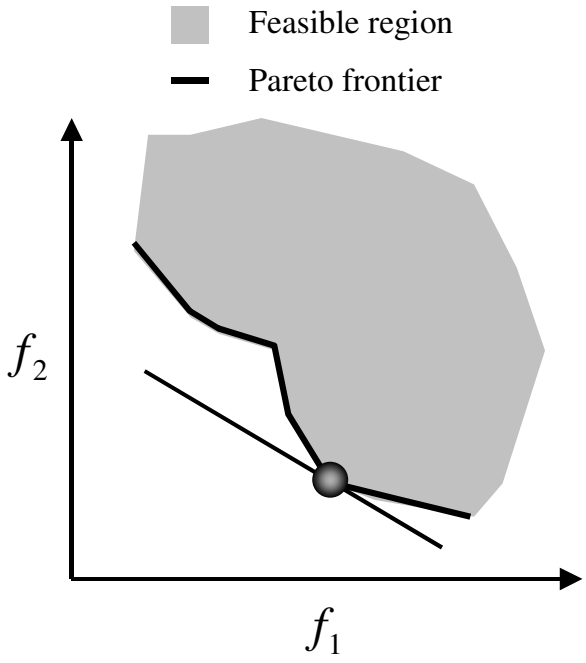


Figure 3.2: Weighted sum method.

ε-Constraint Method

In this method, a primary objective function f_k is selected and then other functions are converted to constraints that have some upper bounds ε. This method is also called

constraint transformation method. The problem is formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}} f_k(\boldsymbol{x}) \\ &\text{subject to } f_i(\boldsymbol{x}) \leq \varepsilon_i,\ i = 1, \ldots, m, i \neq k, \end{aligned}$$

where $\boldsymbol{\varepsilon} = [\varepsilon_1 \ \cdots \ \varepsilon_{k-1} \ \varepsilon_{k+1} \ \cdots \ \varepsilon_m]^{\mathrm{T}}$. This method can find various Pareto optimal solutions by sweep of ε_i even concave Pareto frontier. The conceptual diagram is shown Fig. 3.3.

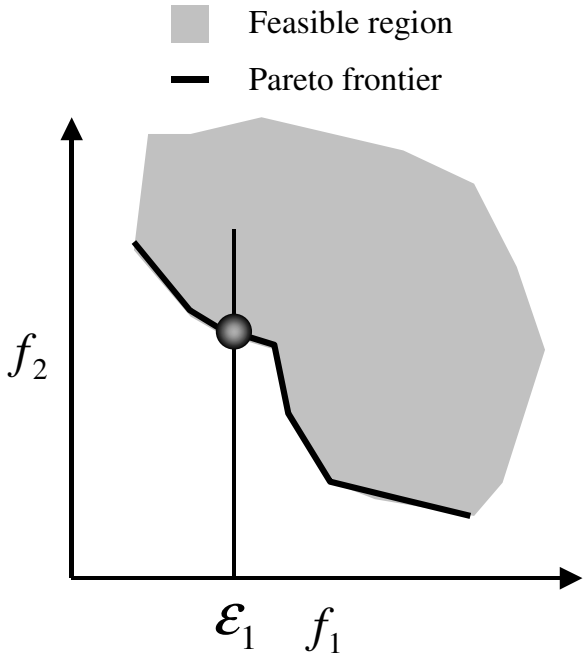


Figure 3.3: ε-constraint method.

These methods are easy to understand and implement, and are widely used. However, these methods obtain only one Pareto optimal solution via one optimization trial.

Pareto optimal set, that is relative solutions for MOPs, has many alternative solutions. Hence, Decision Maker (DM) finally selects the most suitable solution called 'preference solution' from Pareto optimal set after the multi-objective optimization. This process is called Multi-Criterion Decision Making (MCDM). If DM has a prior knowledge such as the priority order, importance (weight), upper bounds, and other trade-off information, DM can determine appropriate a weight vector or constraint vector. Hence, DM might obtain a preference solution via few iterations. If the DM lacks prior knowledge, trial-and-error will be required to obtain the preference solution.

3.3 Evolutionary Multi-objective Optimization

Over past decade, Evolutionary Multi-objective Optimization (EMO) has been paid attention as a practical multi-objective optimization method, because of dramatic progress of the computer performance, and have been applied to many real world problems (See detailed survey by Coello Coello et al. [14]). A general requirement of MOPs is to obtain the whole set of Pareto optimal solutions because plural Pareto optimal solutions exist. Multi-point search, that is one of the features of EAs, is suitable for this requirement. The purpose of MOEAs is to search the Pareto approximation set of the Pareto optimal set as precisely as possible since MOEAs are a stochastic search.

In this section, first, basic framework of Evolutionary Algorithms (EAs) is introduced. EAs are basis methodologies of MOEAs that are extended algorithms suitable for solving MOPs. Next, brief introduction of MOEAs is described. Customarily in EAs, decision variable vector is called an 'individual', the set of individual is called a 'population', objective function is called a 'fitness function', the objective function value is called 'fitness' and so on. In this thesis, these notations adhere to the customs when describing EA operations.

3.3.1 Evolutionary Algorithms

EAs are a generic name of some natural-inspired optimization techniques, and are in the class of stochastic and direct optimization algorithms without gradient information. The origin of EAs was the research activity of three groups established independently in-1960s, they were, Genetic Algorithms (GA) [47, 37], Evolution Strategy (ES) [87, 97], Evolution Programming (EP).

The typical procedure of EAs is described as follow:

1. Initialization: Generate an initial population $P(0)$ randomly, and set the number of generations $t = 0$.

2. Evaluation: Calculate the value $f(\boldsymbol{x})$ of the objective function for individuals in the population $P(t)$.

3. Selection: Select the individuals as parents from the $P(t)$.

4. Crossover: Generate offspring for the population of the next generation with the crossover operation pre-specified probability.

5. Mutation: Add random perturbation to the offspring with the pre-specified mutation probability.

6. Termination: If a termination condition is satisfied, finish this procedure; otherwise, set $t := t + 1$ and return to Step 2).

3.3.2 Multi-objective Evolutionary Algorithms

Multi-objective Evolutionary Algorithms (MOEAs) are optimization methods that make the best use of the features of multipoint search of EAs, and can obtain Pareto approximation set at the same time. Nowadays, MOEAs are widely applied to real world problems via computer simulations such as automotive engineering design [71], optical lens design [81], machine learning [60] and robust design [21]. Various MOEAs techniques, for example, MOGA [34], PESA [16], NSGA-II [19], and SPEA2 [116] have been proposed.

In the following, NSGA-II is introduced as a representative MOEA. In this thesis, the term 'non-dominated individual' means an individual that is not dominated by other individuals in the population. Moreover, the population of final generation is called 'Pareto approximation set', and the population distribution in the objective function space is called 'approximation Pareto frontier'.

NSGA-II

Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II) proposed by Deb et al. [19] is well known as an excellent implementation of MOEA, although various MOEAs have been proposed. NSGA-II is able to obtain high level of accuracy and a wide spread Pareto approximation set with high accuracy and converge by following features.

Feature 1: High validity individual ranking method based on dominance comparison.

Feature 2: Sorting algorithm based on crowding measure to judge the density of population and to maintain diversity.

Feature 3: Elitism to preferentially leave non-dominated individual for the next generation.

Figure 3.4 shows the conceptual diagram of NSGA-II. The algorithm of NSGA-II is as follows:

1. Select parents from population $P(t)$ by binary tournament selection based on the rank and the crowding distance (crowded tournament selection, see Appendix B. 3).

2. Generate offspring population $Q(t)$ by applying crossover and mutation operators to the selected parents.

3. Calculate the fitness of population $R(t) = P(t) \cup Q(t)$.

4. Calculate the rank of $R(t)$ by non-dominated sorting algorithm (see Appendix B. 1) and sort $R(t)$ in ascending order by the rank as a key.

5. Calculate the crowding distance of $R(t)$ of each rank (see Appendix B. 2) and sort the individuals that have same rank in ascending order by the crowding distance as a key.

6. Select high-ranked $|P|$ individuals from $R(t)$ as next generation.

7. If an termination condition is satisfied, finish this procedure; otherwise, return to Step 1).

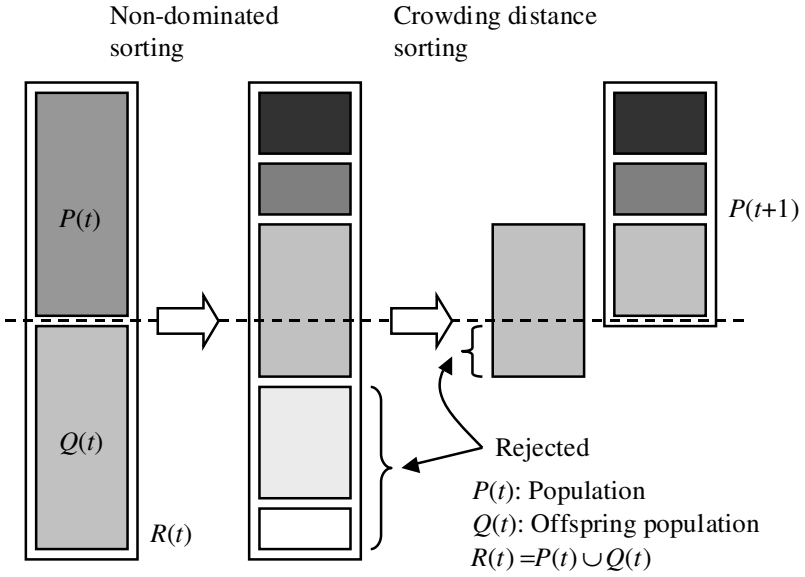


Figure 3.4: Elitist Non-dominated Sorting Genetic Algorithm (NSGA-II).

Details of non-dominated sorting, crowding distance and crowded tournament selection are described in Appendix B.

3.4 Experiment-Based Evolutionary Multi-objective Optimization

3.4.1 Aims of EBEMO

Evolution Strategy (ES), one of EA method proposed by Rechenberg and Schwefel in-1960s, was originally developed for experiment-based optimization [87, 97]. In this thesis, the term 'experiment-based optimization' is used to mean "the system parameters of a real system are optimized directly by optimization techniques in real time through experiments".

In real experiments, observation values usually include measurement error by sensor noise, transient response and so on. Hence, gradient-based optimization methods such as steepest decent method and quasi Newton method are difficult to apply

to experiment-based optimization because the numerical differential calculation using fluctuated observation values misleads the search direction. However, direct search methods such as local search, simulated annealing do not use gradient information. Therefore, they can adequately avoid the influence of fluctuated observation values. In particular, because EAs are multi-point searches that have a set of search information, they have robustness for measurement error in the sense that they can optimize mean fitness of population. Therefore, experiment-based optimization by using EAs is a suitable approach for real world problems.

Various engineering applications of experiment-based optimization using EAs have been studied. For example, Kim and Lee have applied ES to a time delay controller for the position control system of electro-hydraulic servo [65]. Shi et al. have optimized the covariance matrices of Extended Kalman Filter for an induction motor speed estimation by GA [98]. Neri et al. have applied their original GA that considers observation noise, for gain and coefficient optimization of an AC servo motor controller [77]. Moreover, Kamihira et al. have proposed Bio-Control Architecture (BCA) for adaptive vehicle controller [61]. Although most previous research has used single-objective EAs, Büche et al. have applied NT-SPEA that has a noise tolerant scheme to parameter tuning of gas-turbine controllers [10].

Therefore, it is a natural and realistic approach to apply MOEAs directly to HILS environments introduced in Chapter 2. Both of improvement in computer performance and progress of automatic measurement environments such as HILS make for easy installation of MOEAs. This approach is called Experiment-Based Evolutionary Multi-objective Optimization (EBEMO) in this thesis. In EBEMO for engine calibration, we do not need to build theoretical or statistical engine models needed in RSM, instead we are able to use the optimization results directly. Therefore, some processes such as design of experiments, model selection, and verification can be omitted. Moreover, the automated data measurement using the HILS lightens the burden of the parameter adjustment. As a result, since a certain number of the trial-and-error evaluations by MOEAs can be allowed, the EBEMO using HILS environments can be a smarter calibration technique.

3.4.2 Difficulties of EBEMO

Because EAs are robust optimization techniques, online optimization applied naive application of existing MOEAs to experiment-based optimization may be an effective approach for complex real systems that can not be modeled easily. However, to achieve a practical level EBEMO have to solve two main problems in its application.

The first problem is the search performance of MOEAs in real environments. Conventional MOEAs do not sufficiently consider some factors that deteriorate search performance. For example, observation noise of sensors included in output is one such factor. Additionally, especially in engine calibration, the influence of transient response

by parameter switching is a serious problem. Moreover, from a viewpoint of engine control, we also have to consider the periodicity of control inputs.

The second problem is the search time of MOEAs in real environments. Generally speaking, MOEAs require a tremendous number of evaluations. While we can obtain many measurements with HILS, there still exists severe limitations in the number of fitness evaluations because real experiments need real-time evaluations. Hence, it is difficult to obtain a Pareto approximation set in practical time with conventional MOEAs. In addition, plural MOPs corresponding to the operating conditions of maps have to be optimized to integrate map-based controllers. For example, when a map is represented by 10×10 grids, we have to solve 100 MOPs at one engine calibration process.

Therefore, we should develop some innovative techniques to overcome the difficulties that are potentially and essentially included in online optimization of real systems. In the following chapters, five techniques to solve aforementioned difficulties for EBEMO are proposed.

3.5 Summary

In this chapter, the definition of MOPs and the concept of Pareto optimal solution were introduced. In addition, the foundation of conventional multi-objective optimization methods and MOEAs were introduced. Moreover, as a typical MOEA technique, the detail of NSGA-II proposed by Deb et al. was denoted. The concept of EBEMO was also introduced. In particular, the aims of EBEMO approach and its difficulties were emphasized. EBEMO is an integrated problem of various difficulties. The rest of this thesis, these piled difficulties will closely be solved one by one to compose automatic engine calibration methods.

Chapter 4

Handling of Observation Noise

4.1 Outline

Currently, MOEAs are applied to many real-world problems as multi-objective optimization methods. Because Evolutionary Algorithms are a flexible framework, multi-objective optimization for real systems and simulations using random numbers are promising applications. Observation values of these applications have uncertain fluctuation, which is one of its attractive research topics [10, 36, 48, 99, 100].

The aim of multi-objective optimization by MOEAs is acquisition of Pareto approximation set which uniformly distributed in the neighborhood of Pareto frontier. However, search efficiency under constrained evaluations and precision of shape of approximation Pareto frontier fluctuated by uncertainty of observations are also important factors, when real systems are optimized online. Previous studies have focused on techniques to obtain the Pareto approximation set. There has been little researches focused on the search efficiency and the precision of the shape of approximation Pareto frontier.

In the following, the problem with the application of conventional MOEAs to MOPs having observation noise is discussed. Then, Memory-based Fitness Estimation and Distribution-based Selection GA (MFE-DSGA) that is suitable for noisy environments is proposed. Moreover, the effectiveness of the proposed method is confirmed through numerical experiments and a real engine experiment.

4.2 MOEAs for Uncertainty Environment

4.2.1 Background

In this chapter, unconstrained m-objective optimization problems that have uncertainty in observation values, that is, observation noise are considered. The unconstrained multi-objective optimization problem with observation noise is defined as fol-

lows:

$$\min_{\boldsymbol{x}} \langle \boldsymbol{F}(\boldsymbol{x}) \rangle, \quad \boldsymbol{x} \in X \subset \mathbf{R}^m$$

$$\overbrace{\boldsymbol{F}(\boldsymbol{x})}^{\text{Sampled value}} = \overbrace{\boldsymbol{f}(\boldsymbol{x})}^{\text{True fitness}} + \overbrace{\boldsymbol{\delta}}^{\text{Observation noise}},$$

$$\boldsymbol{F}(\boldsymbol{x}) = \begin{bmatrix} F_1(\boldsymbol{x}) \\ \vdots \\ F_n(\boldsymbol{x}) \end{bmatrix}, \boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} f_1(\boldsymbol{x}) \\ \vdots \\ f_n(\boldsymbol{x}) \end{bmatrix}, \boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix},$$

where $\langle\rangle$ represents expectation value over observation noise δ. In other words, this problem defines as a minimization problem of expectation value of sample $\langle \boldsymbol{F}(\boldsymbol{x}) \rangle$. In this thesis, continuous space is assumed as search space of $\boldsymbol{x}$.

Features 1 to 3 introduced by NSGA-II described in Chapter 3 are indispensable elements for MOEAs in problems that do not have uncertainty. However, the following three problems occur in NSGA-II in environments with observation noise [10, 48].

Problem 1: The ranking algorithm becomes unstable. This problem easily leads to the dismissal of good individuals and stagnation of the search.

Problem 2: The distribution of the population becomes biased because the crowding measure is not appropriately allocated. Hence, diversity maintenance does not work well.

Problem 3: Since the individual having good sampled fitness value is treated as an elite and stored for a long term even if its true fitness is bad, the search performance is decreased. In particular, dominated individual might be treated as non-dominated individual by the dominance comparison based on the sampled value. This phenomenon easy occurs in a multi-objective optimization problem that has a weakly Pareto frontier.

Fig. 4.1 shows the conceptual diagram of the phenomenon of which a weakly Pareto individual becomes a 'fake' non-dominated individual. Assume that the observation noise is added for true fitness $\boldsymbol{f}(\boldsymbol{x})$ of weakly Pareto individual $\boldsymbol{x}$ within the range shown in gray circle. If observation noise that improves objective function f_1 is added, $\boldsymbol{x}$ is treated as a non-dominated individual by the dominance comparison based on the sample value $\boldsymbol{F}(\boldsymbol{x})$. This fake non-dominated individual can survive until dominated by other individuals. When the weakly Pareto frontier is widened, a large crowding measure can be allocated to $\boldsymbol{x}$ easily and the possibility selected as parent increases. As a result, it causes the search performance to be decreased.

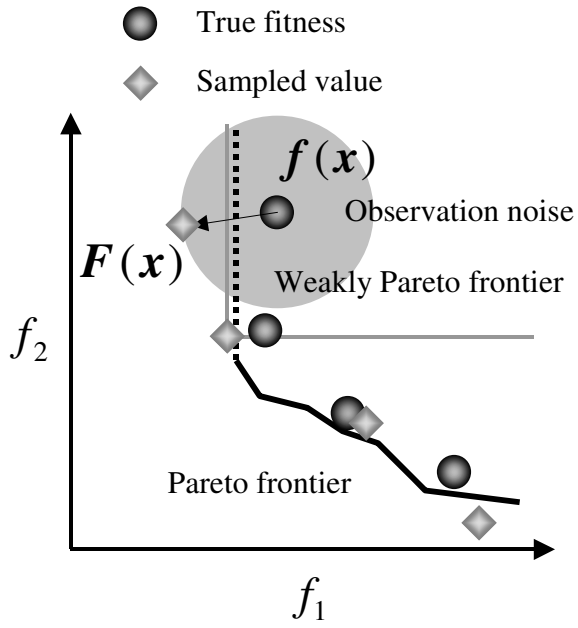


Figure 4.1: A weakly Pareto individual treated as a non-dominated individual.

4.2.2 Related Works

For the aforementioned problems, the following methods have been proposed as MOEAs that are suitable for the uncertainty:

Rank Estimation

Hughes has proposed Multi-Objective Probabilistic Selection EA (MOPSEA) [48]. MOPSEA applies the rank estimation method based on probability distribution for conventional MOEAs.

This approaches can solve problems 1 and 3, because the ranking becomes stable through rank estimation. However, it is necessary to know the standard deviation of the observation noise beforehand. Moreover, problem 2 occurs because the sample value is used for crowding measure calculations.

Fitness Estimation

In this thesis, estimation of true fitness value $f(\boldsymbol{x})$ is expressed as $\tilde{f}(\boldsymbol{x})$. $\tilde{f}(\boldsymbol{x})$ is called estimated value.

Singh et al. have discussed MOEAs that uses the mean value of plural sample values as the estimated value [100]. They have recommended five or more as the number of samples to be used. In addition, they also have proposed a method that uses the mean of sample values of past evaluated same individual stored in search history as an estimated value. This method can solve problems 1 and 2 because the ranking

and crowding measures become stable by using the estimated values. However, the increase in the number of evaluation is a significant problem. When a small number of samples is evaluated, problem 3 cannot be solved because estimation error cannot be reduced sufficiently. Singh has proposed a method combining this method with MOPSEA. However, the problem with the number of evaluations cannot be solved.

Consideration of Influence of Elitism

Singh et al. have investigated non-elitism NSGA-II [99]. They concluded that the performance of this method is better than that of conventional NSGA-II. However, when elitism is not used, it is thought that convergence slows down, and hence making it unsuitable for a real problem.

Büche et al. have proposed NT-SPEA (Noise Tolerant SPEA) and applied it for the experiment-based optimization of gas turbines [10]. In NT-SPEA, non-dominated individuals have their own lifespan and are re-evaluated when the lifespan is completed. This technique can regularly remove outliers, that is, individuals added extreme noise. However, problems 1 and 2 cannot be solved because NT-SPEA uses sampled values.

4.3 Memory-based Fitness Estimation and Distribution-based Selection Genetic Algorithm

In this section, MOEAs that are suitable for a problem having uncertainty of observation value is proposed. This MOEA is called Memory-based Fitness Estimation and Distribution-based Selection Genetic Algorithm (MFE-DSGA). MFE-DSGA utilizes the following three solutions to solve three problems that discussed in the previous section.

Solution 1: To stabilize the search for uncertainty by using estimated value for individual evaluation, a fitness estimation method that does not increase the number of evaluation is introduced.

Solution 2: To maintain the diversity and to improve estimation accuracy, a selection method that pays attention to population distribution is introduced. The parents are selected from a sparse area of population.

Solution 3: To maintain the effectiveness of elitism by excluding the weakly Pareto individuals treated as fake non-dominated individual, an extended dominance comparison is introduced.

A prototype algorithm of MFE-DSGA is shown below. In this algorithm, Euclidean distance is used in both of decision variable space and objective function space. Hence, domain of decision variables and range of objective function have to be normalized beforehand.

1. Evaluate all fitness vector $\boldsymbol{f}(\boldsymbol{x})$ of the initial population $P(0)$, $t = 0$ in a real environment, and preserve individuals and their fitness vector in search history set H.

2. Select one parent from a sparse area of distributed population in the objective function space from the non-dominated population $P_{\text{nd}} \in P(t)$ (details are described in Subsection 4.3.2), and select another parent at random from $P(t)$.

3. Generate a family population $F(t)$ that is composed by the aforementioned parents and offspring generated by crossover, and obtain their sampled value.

4. Calculate the estimated value of population $R(t) = P(t) \cup F(t)$ by using the memory-based fitness estimation method (details are described in Subsection 4.3.1).

5. Calculate the rank of $R(t)$ by an extended dominance comparison (details are described in Subsection 4.3.3). An offspring in F that has the same chromosome of individual in $P(t)$ is ranked the lowest.

6. Calculate the crowding measure of $R(t)$ of each rank, and sort them in ascending order based on the crowding measure, and select high-ranked $|P|$ individuals of $R(t)$ for the next generation.

7. If the termination condition is satisfied, finish this procedure; otherwise, set $t := t + 1$ and return to Step 2).

The aim of Step 2) in MFE-DSGA is to generate offspring around specific parents. When a large size of the offspring population is taken such as in conventional NSGA-II, it is easy to lose the diversity in the early stages of the search. Therefore, to prevent this problem, the size of family $|F|$ is set smaller than $|P|$. In this chapter, UNDX (Unimodal Normal Distribution Crossover) proposed by Ono et al. [80] is used as crossover operator. Moreover, crowding distance of NSGA-II as a crowding measure and Pareto ranking method proposed by Fonseca and Fleming [34] as a ranking method are used. In Fonseca's Pareto ranking method, rank $r(\boldsymbol{x})$ of individual $\boldsymbol{x}$ dominated by p individuals is calculated by a following equation:

$$r(\boldsymbol{x}) = 1 + p. \tag{4.1}$$

Details of aforementioned solutions are described in following subsections.

4.3.1 Fitness Estimation Method for MOEAs

Sano and Kita have proposed Memory-based Fitness Estimation Genetic Algorithm (MFEGA) that have a fitness estimation method [91, 93]. MFEGA is a Real-Coded Genetic Algorithm (RCGA) suitable for single-objective optimization for a fitness function with uncertainty. The algorithm of MFEGA is shown as follows:

1. Install a stochastic model of uncertainty of fitness function before the search.
2. Obtain sampled values of the fitness value of individuals during the search.
3. Preserve individuals and their sampled values as a search history.
4. Estimate the fitness value of individuals using a statistical method based on the stochastic model defined in Step 1) and the search history.

MFEGA stores evaluated individual in the past to the search history and estimates fitness value of focused individuals by using the search history. It has been confirmed that MFEGA is able to optimize noisy test problems more effectively than plural sampling method. Figure fig4-2 shows the uncertainty model that MFEGA adopts.

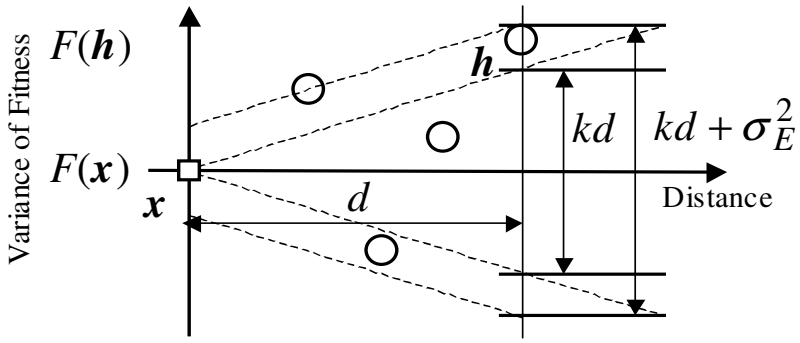


Figure 4.2: A stochastic model of fitness functions.

This model combines two uncertainties. One is change of fitness function represented by normal distribution. The variance of the normal distribution is assumed to be proportional to the distance from the focused individual. The other is observation noise. In Fig. 4.2, the horizontal axis indicates Euclidean distance from the focused individual in the decision variable space, and vertical axis indicates the variance of fitness. Assuming that fitness $f(\boldsymbol{h})$ of individual $\boldsymbol{h}$ seen from focused individual $\boldsymbol{x}$ represented by a square is represented by a normal distribution given by mean $f(\boldsymbol{x})$ that is the true fitness value, and variance kd that is change of fitness function:

$$f(\boldsymbol{h}) \sim N(f(\boldsymbol{x}), kd), \tag{4.2}$$

which is proportionally changed to distance d from $\boldsymbol{x}$, where k is the weight of distance in the decision variable space.

Hence, sampled value $F(\boldsymbol{h})$ is modeled as $f(\boldsymbol{h})$ that is added observation noise represented by normal distribution given by mean 0 and variance σ_E^2:

$$F(\boldsymbol{h}) = f(\boldsymbol{h}) + \delta \sim N(f(\boldsymbol{x}), kd + \sigma_E^2), \tag{4.3}$$

$$\delta \sim N(0, \sigma_E^2). \tag{4.4}$$

From the model shown in Fig. 4.2, estimated value $\tilde{f}(\boldsymbol{x})$ of true fitness $f(\boldsymbol{x})$ is obtained by using the following maximum likelihood estimation method:

$$\tilde{f}(\boldsymbol{x}) = \frac{F(\boldsymbol{x}) + \sum_{l=2}^{H} \frac{1}{k' d_l + 1} F(\boldsymbol{h}_l)}{1 + \sum_{l=2}^{H} \frac{1}{k' d_l + 1}}. \tag{4.5}$$

Individuals in the search history H are represented by $\boldsymbol{h}_l, l = 1, 2, \ldots, H$, and $F(\boldsymbol{h}_l)$ is sampled value at $\boldsymbol{h}_l$. d_l is the Euclidean distance from the focused individual $\boldsymbol{x}$ to $\boldsymbol{h}_l$. Moreover, it is $k' = k/\sigma_E^2$.

In this chapter, k-Fixed MFEGA that uses constant k' is used [91, 93]. The parameter k' is unknown including variance σ_E^2 of observation noise. In experiment-based optimization, σ_E^2 is set taking into consideration prior knowledge such as data taken in preliminary experiments.

Since this fitness estimation method of MFEGA is an independent algorithms from GAs, it can be easily combined with MOEAs. Hence, it is expected that the fitness estimation method can stabilize individual ranking and search effectively.

However, this fitness estimation method of MFEGA is not necessarily suitable for MOEAs because the method adopts the uncertainty model that assumes the use of single-objective GAs. Equation (4.5) calculates estimated value by using stored individuals in search history, and the individuals are weighted by the reciprocal of Euclidean distance. On the one hand, in single-objective GAs, the weights of distant individuals becomes relatively small, because the population converges in the vicinity of the optimal solution in the last stage of the search. On the other hand, the population is widely distributed in the vicinity of Pareto optimal set in MOEAs. If the aforementioned Eq. (4.5) is used for estimation, the boundary of Pareto frontier (both ends of the Pareto curve in cases with two objectives) are allocated comparatively large weights. Hence, the accuracy of the estimated value deteriorates. To cope with this problem, following fitness estimation equation is used for each objective function:

$$\tilde{f}_i(\boldsymbol{x}) = \frac{F_i(\boldsymbol{x}) + \sum_{l=2}^{H} \frac{1}{k_i' {d_l}^3 + 1} F_i(\boldsymbol{h}_l)}{1 + \sum_{l=2}^{H} \frac{1}{k_i' {d_l}^3 + 1}}, \ i = 1, \ldots, n. \tag{4.6}$$

In Eq. 4.6, we adopt cubic dependency on distance d_l instead of the linear one in Eq. 4.5.

Thus the influence received from distant individuals in the search history is suppressed more. Hence, an appropriate estimated value can be expected to be obtained even in the last stage of the search.

4.3.2 Selection Operator using Population Distribution

As discussed in Chapter 3, binary tournament selection and elitism type generation alternation models are widely used in MOEAs such as NSGA-II. In addition, offspring population size $|Q|$ equals to population size $|P|$ in almost all implementations. These two operations lead to the loss of diversity and selection bias of specific parents. In NSGA-II etc., when the number of non-dominated individuals exceeds the population size, diversity maintenance operation for non-dominated individuals are executed. However, it is difficult to make all individuals becomes non-dominated individuals under uncertainty. In this situation, bias of distribution of the sampled values in the search history is caused. As a result, the accuracy of the estimated value worsens in some areas. To maintain the diversity and to improve the accuracy of estimated value all over the Pareto frontier with certainty, a selection operator that pays attention to the population distribution is introduced. This method is called 'distribution-based selection'.

An algorithm for two-objective problems are described as follows:

1. Sort non-dominated population P_{nd} in ascending order by the objective function f_1 as a key.
2. Calculate the Euclidean distance between adjoined individuals in the objective function space.
3. Select one individual at random as a parent from two individuals that comprise the maximum distance.
4. Select one individual at random as the other parent from the population P.

When the fitness vectors of non-dominated population P_{nd} are sorted in ascending order by objective function f_1 as a key, objective function f_2 is sorted in descending order. The distance between adjacent individuals is calculated by using this feature in this algorithm.

In more than three objective problems, non-dominated individuals are distributed on a manifold. In such cases, when objective function vectors are sorted by f_1 as a key, other elements of objective functions are not sorted. Therefore, it is necessary to determine the neighborhood of individuals by some method. In MFE-DSGA, the neighborhood is determined by Delaunay triangulation of non-dominated individuals. Delaunay triangulation is an algorithm of computational geometry [4].

An algorithm for m-objective problems is described as follows:

1. Project non-dominated population P_{nd} in m-dimensional objective space for $m-1$-dimensional hyper-plane.
2. Apply Delaunay triangulation of projected P_{nd}. P_{nd} is divided to $m-1$-dimensional simplex in $m-1$-dimensional hyper-plane.

3. Transpose projected P_{nd} to m-dimensional space by adding the element that is removed.

4. Restore the element lost through projection to triangulated P_{nd}, and return them to m-dimensional space.

5. Calculate the volume of all simplex.

6. Randomly select one individual among n individuals that compose the maximum simplex.

7. Randomly select one individual as the other parent from population P.

The computational complexity of Delaunay triangulation in a two-dimensional plane is $O(|P_{\mathrm{nd}}| \log |P_{\mathrm{nd}}|)$, where $|P_{\mathrm{nd}}|$ is the number of non-dominated individuals. If geometry conversion [4] is used, Delaunay triangulation of m-dimensional space becomes equivalent to the problem of constructing a convex hull in $m+1$ dimensional space. Hence, it is understood that the distribution-based selection algorithm for two-dimensional space is essentially the same as the algorithm for m-dimensional space. If Pareto frontier is continuous, offspring are easily generated around non-dominated individuals in the sparsely populated areas by proposed distribution-based selection. It can be expected to positively search maintaining diversity. For a problem that has discontinuous Pareto frontiers, there is a possibility that non-dominated individuals on the edge of a Pareto frontier could be selected as a parent. This parent weakens the effect that generates the offspring in sparsely populated areas of non-dominated individuals. However, the distribution-based selection is a realistic choice because it is thought that real world problems hardly have discontinuous Pareto frontiers.

4.3.3 α-domination Strategy

In experiment-based optimization, due to observation noise or estimation error, there is a possibility that dominated individual may survive as a non-dominated one. Figure 4.1 shows a situation in which a weakly Pareto individual is treated as a non-dominated individual. Such an individual is called 'fake' non-dominated individual in this chapter.

To solve this problem, we use the α-domination strategy proposed by Ikeda et al. [49] for the ranking applied to the estimated values. The α-domination strategy is defined as an expansion of dominance comparison as follows:

Definition 4.2. (α-domination)

Let $\boldsymbol{x}^1, \boldsymbol{x}^2 \in X$, where $\boldsymbol{x} = [x_1\ x_2\ \ldots\ x_n]^{\mathrm{T}}$. $\boldsymbol{x}^1$ is said to α-dominate $\boldsymbol{x}^2$ if and only if

$$\forall i \in \{1, 2, \ldots, m\},\ g_i(\boldsymbol{x}^1, \boldsymbol{x}^2) \leq 0\ \wedge\ \exists i \in \{1, 2, \ldots, m\},\ g_i(\boldsymbol{x}^1, \boldsymbol{x}^2) < 0,$$

where

$$g_i(\boldsymbol{x}^1, \boldsymbol{x}^2) = f_i(\boldsymbol{x}^1) - f_i(\boldsymbol{x}^2) + \sum_{j \neq i}^{n} \alpha_{ij}(f_j(\boldsymbol{x}^1) - f_j(\boldsymbol{x}^2)).$$

In α-domination strategy, weighted linear sum of all objective functions is used for the definition of dominance comparison. The weights are given by the ratio of each objective function. When an objective function f_i is compared by the α-domination strategy, another objective function f_j is considered in a ratio of α_{ij}. The conceptual diagram of the α-domination strategy is shown in Fig. 4.3. The solid black line indicates a Pareto frontier. The black and gray circle indicate non-dominated and dominated individual, respectively. Normal dominance comparison is execute by the dotted gray line. While the dominance comparison based on α-domination strategy is executed by the solid gray line. For example, an individual that needs to greatly corrupt f_j to improve f_i a little is dominated easily. As a result, fake non-dominated individuals are excluded efficiently.

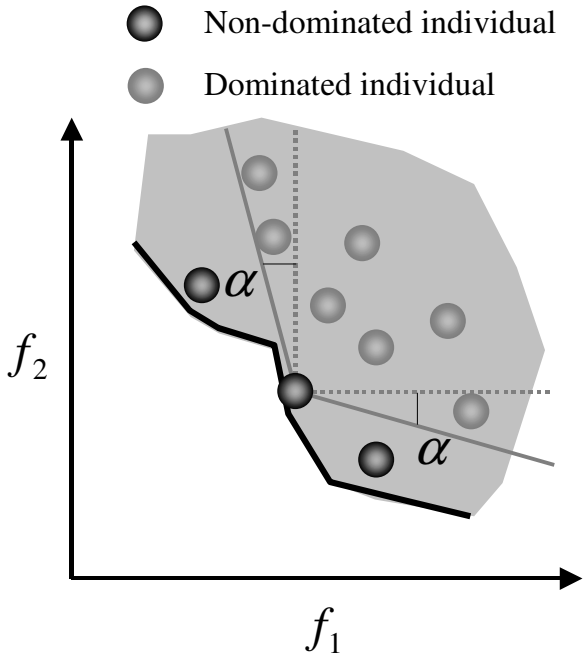


Figure 4.3: α-domination strategy.

4.4 Numerical Experiments

4.4.1 Experiment Settings and Measures

Two-objective optimization problem ZDT1 and ZDT2, and three-objective optimization problem DTLZ2 [20, 114] that are standard test problems of MOEAs are described

as follows:

- ZDT1

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= x_1 \\
f_2(\boldsymbol{x}) &= g(\boldsymbol{x})(1 - \sqrt{f_1/g(\boldsymbol{x})}) \\
g(\boldsymbol{x}) &= 1 + 9 \cdot \sum_{i=2}^{n} \frac{x_i}{n-1} \\
x_i &\in [0,1], \quad i = 1, \ldots, n
\end{aligned}
$$

  This function has a convex Pareto frontier. In this experiment, $n = 2$ is used.

- ZDT2

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= x_1 \\
f_2(\boldsymbol{x}) &= g(\boldsymbol{x})(1 - (f_1/g(\boldsymbol{x}))^2) \\
g(\boldsymbol{x}) &= 1 + 9 \cdot \sum_{i=2}^{n} \frac{x_j}{n-1} \\
x_i &\in [0,1], \quad i = 1, \ldots, n
\end{aligned}
$$

  This function has a concave Pareto frontier. In this experiment, $n = 2$ is used.

- DTLZ2

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= (1 + g(\boldsymbol{x})) \cos(x_1\pi/2) \cos(x_2\pi/2) \\
f_2(\boldsymbol{x}) &= (1 + g(\boldsymbol{x})) \cos(x_1\pi/2) \sin(x_2\pi/2) \\
f_3(\boldsymbol{x}) &= (1 + g(\boldsymbol{x})) \sin(x_1\pi/2) \\
g(\boldsymbol{x}) &= \sum_{i=3}^{n} (x_i - 0.5)^2 \\
x_j &\in [0,1], \quad i = 1, \ldots, n
\end{aligned}
$$

  This function is m-objective optimization problem ($m \geq 3$) having a concave Pareto frontier. In this experiment, $m = 3$ and $n = 3$ are used.

Finally, noisy test problems using numerical experiments based on ZDT1, ZDT2, and DTLZ2 are defined as follows:

$$
F_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \delta_i, \ \delta_i \sim N(0, \sigma_N^2), \ i = 1, \ldots, n,
$$

where δ_i represents observation noise, and $N(0, \sigma_N^2)$ is 0 mean σ_N^2 variance normal distribution random number. These problems are called Noisy ZDT1, Noisy ZDT2 and Noisy DTLZ2, respectively. The number of dimension is $m = 2$ for Noisy ZDT1

and Noisy ZDT2, and $m = 3$ for Noisy DTLZ2. Standard deviation of δ_i is $\sigma_N = 0.1$ for all problems.

In this experiment, performances of MOEAs were compared by the mean values of the plural trials for three evaluation measures described below:

Coverage: This measure is proposed by Hiroyasu et al. [44], and it indicates the ratio of Pareto frontier that is covered by the population. The coverage is defined as

$$C = \frac{1}{m} \sum_{i=1}^{m} \frac{c_i}{c_{\max}}, \tag{4.7}$$

where m is the number of the objective functions, $c_{\max}$ is the number of small areas where a hyper-plane composed of $m-1$ objective functions are evenly divided, and c_i is the number of areas including the true fitness of individuals projected to the hyper-plane. The best value of this measure is 1. In the case of two-objective problems, the population is projected to f_1 and f_2 axises. In the case indicated in the left hand side of Fig. 4.4, the coverage is $C = 1/2 \times (3/5 + 3/5) = 0.6$. Similarly in the case of three-objective problems, the population is projected to three planes $f_1 - f_2$, $f_1 - f_3$, and $f_2 - f_3$. The conceptual diagram is shown in the right hand side of Fig. 4.4.

In this experiment, $c_{\max} = 50$ ($m = 2$) and $c_{\max} = 10$ ($m = 3$) were used. Because MFE-DSGA introduces α-domination strategy, decreased coverage is predicted. Hence, we can verify the influence by using this measure.

Mean Absolute Error: This measure indicates the error of the population for Pareto optimal set X^*. The mean absolute error is defined as the mean value of Euclidean distances from each individual to the nearest solution in X^*. The conceptual diagram is shown in Fig. 4.5. In this experiment, the Pareto optimal sets of ZDT1, ZDT2, and DTLZ2 are given by

$$X^*_{\mathrm{ZDT1}} = X^*_{\mathrm{ZDT2}} = \{x_1 \in [0,1],\ x_2 = \ldots = x_n = 0\},$$
$$X^*_{\mathrm{DTLZ2}} = \{x_1, x_2 \in [0,1],\ x_3 = \ldots = x_n = 0.5\}.$$

Hence, the Mean Absolute Error of the population is calculated by

$$A_{\mathrm{error}} = \begin{cases} \dfrac{1}{|P|n} \displaystyle\sum_{i=1}^{|P|} \sum_{j=2}^{n} |x_j^i| & (\mathrm{ZDT1,2}) \\ \dfrac{1}{|P|n} \displaystyle\sum_{i=1}^{|P|} \sum_{j=3}^{n} |x_j^i - 0.5| & (\mathrm{DTLZ2}) \end{cases}, \tag{4.8}$$

where $|P|$ is the population size of P. The best value of this measure is 0.

Mean Estimation Error: Mean Estimation Error between true fitness and estimated value is defined as follows:

$$E_{\text{error}} = \frac{1}{|P|m} \sum_{i=1}^{|P|} \sum_{j=1}^{m} |f_j(\boldsymbol{x}^i) - \tilde{f}_j(\boldsymbol{x}^i)|. \tag{4.9}$$

Minimum value of the mean estimation error is 0. This measure is used for verification of ascendancy of MFE-DSGA to plural sampling.

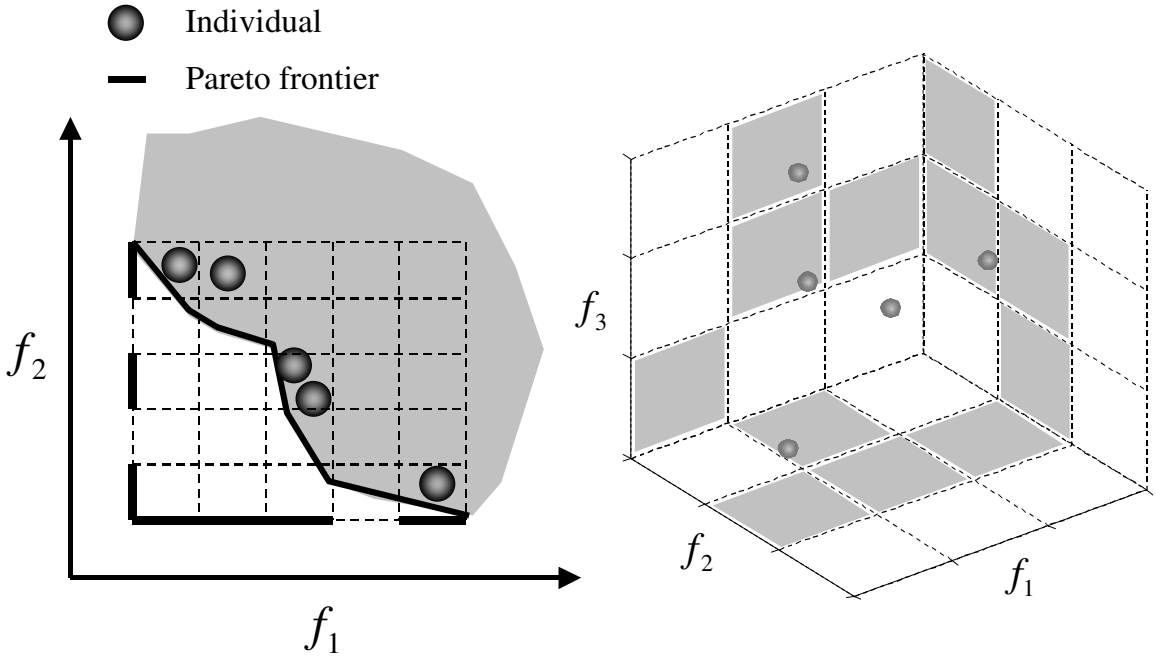


Figure 4.4: Coverage measure (left: two-objective, right: three-objective).

In numerical experiments, following three method was used.

MFE-DSGA: Setting parameters k' and α_{ij} are determined by pre-experiments.

NSGA-II: The sampled value is used as the fitness. Individuals are not re-evaluated in this method.

Sample-5 NSGA-II: The mean value of five times of the sample value is used as the estimated value. Individuals are not re-evaluated in this method.

Additionally, test problems without noise are optimized by NSGA-II to compare the performance under noiseless environments. Experiment settings are shown in Table 4.1. In these numerical experiments, the number of evaluations was considerably smaller than in typical experiments, because real engine experiments were considered. The performances of each method are compared by the mean value of 30 trials of evaluation criteria.

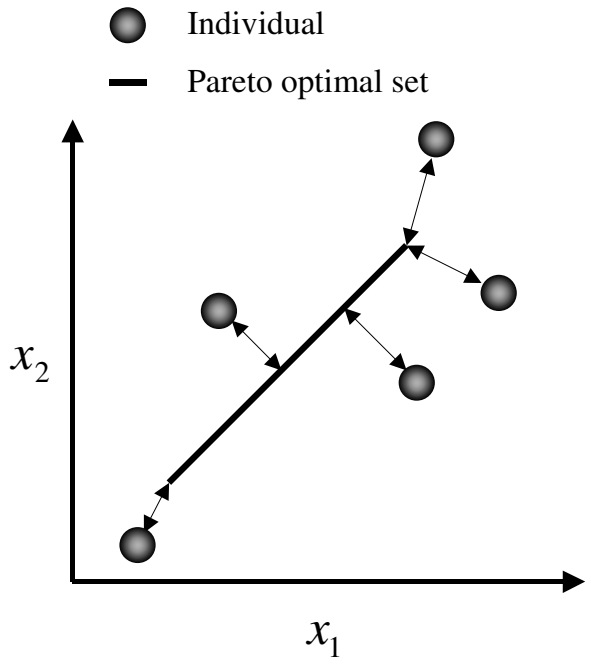


Figure 4.5: Mean absolute error measure.

Table 4.1: Experiment parameter.

Parameter	NSGA-II	MFE-DSGA
Evaluation	3000	
Population Size	100	
Children Size	100	8
$k_i'(i = 1, \ldots, n)$	—	310000(Noisy ZDT1,2) 31000(Noisy DTLZ2)
$\alpha_{ij}(i, j = 1, \ldots, n, j \neq i)$	—	0.1

4.4.2 Discussion of Results

Table 4.2 shows the results of experiments. It is understood that MFE-DSGA indicated the best performance on coverage, mean absolute error and mean estimation error. Especially, it shows very high performances on other methods for Noisy ZDT1 and Noisy ZDT2. Moreover, decrease of coverage is relatively small compared with the Noiseless NSGA-II. However, the proposed method only obtained a slightly better result than Sample-5 NSGA-II on Noisy DTLZ2 the three objective optimization problem used as the benchmark. The defined objective function space of DTLZ2 is comparatively small while the function spaces of ZDT1 and ZDT2 are large. Hence, because all methods are capable of obtaining good individuals in the early stages of

the search, it seems that the difference is minor. In comparison of mean estimation error, precision of the estimated value of MFE-DSGA is better than that of Sample-5 NSGA-II. Note that the performance difference in coverage comparison is the result of the following:

- Coverage of MFE-DSGA mainly is lost by α-domination strategy that cuts down on both ends of the Pareto frontier.
- Coverage of NSGA-II and Sample-5 NSGA-II is mainly lost as a result of bias in the population distribution.

Figures 4.6, 4.7 and 4.8 show the final generation population distribution of true fitness and estimated value (in NSGA-II, sampled value is used.). We can see that the estimated values of MFE-DSGA accurately approximate true Pareto frontier. This information is sufficient for decision making. In NSGA-II, the difference between sampled value distribution and true Pareto frontier is very large even though the population converged. Additionally, in Sample-5 NSGA-II, the convergence of population is not good despite the fact that the error ratio between true fitness and estimated value is small. Plural sampling can improve accuracy of estimated value. However, since the number of generation is few, the population hardly converges to the Pareto frontier. Moreover, we can confirm that the fake non-dominated individuals can be excluded by α-domination strategy, although they survived in the results of NSGA-II and Sample-5 NSGA-II.

In short, we can conclude that MFE-DSGA is effective for real problems where evaluating the fitness function many times is difficult.

4.5 Real Engine Experiment

In this section, we execute on-line multi-objective optimization of real engine in order to validate the effectiveness of proposed method. Here, the decision variables are the control parameters of Electronic Control Units (ECUs), and the objective functions are exhaust emissions. Electronic devices are precisely controlled, and the condition fluctuations of extended operation of the engine are relatively small. However, because of the measurement variation of exhaust emissions measured by the exhaust gas analyzer are large, this problem can be treated as a problem that has uncertainty fluctuation of observation values. Hence, MFE-DSGA can be expected to be effective.

4.5.1 Experiment Settings

The four stroke gasoline engine from a road sport motorcycle was used in the experiments to validate the effectiveness of proposed algorithm. MFE-DSGA were applied to a two-objective four-variable function optimization problem of the engine. Setting of the multi-objective optimization problem was as follows:

Table 4.2: Performance comparison of the three methods for the three test functions

Test Function	MOEAs	Coverage		Absolute Error		Estimation Error	
		Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
Noisy ZDT1	MFE-DSGA	**0.84267**	0.046591	**0.002035**	0.000748	**0.013190**	0.002637
	NSGA-II	0.78067	0.038679	0.024159	0.012559	0.140300	0.005006
	Sample-5 NSGA-II	0.69100	0.066662	0.073992	0.021228	0.037734	0.001971
	Noiseless NSGA-II	0.94367	0.021088	0.003778	0.006627	—	—
Noisy ZDT2	MFE-DSGA	**0.79700**	0.063634	**0.001463**	0.000590	**0.018623**	0.007046
	NSGA-II	0.68567	0.045840	0.017812	0.010971	0.135660	0.005888
	Sample-5 NSGA-II	0.41733	0.117090	0.088146	0.034367	0.037540	0.001937
	Noiseless NSGA-II	0.93900	0.023540	0.002680	0.004080	—	—
Noisy DTLZ2	MFE-DSGA	**0.51600**	0.020199	**0.095630**	0.012560	**0.034206**	0.002624
	NSGA-II	0.45578	0.029006	0.114080	0.012075	0.115860	0.003804
	Sample-5 NSGA-II	0.51367	0.028972	0.120890	0.018613	0.038379	0.002225
	Noiseless NSGA-II	0.53089	0.021495	0.089648	0.014595	—	—

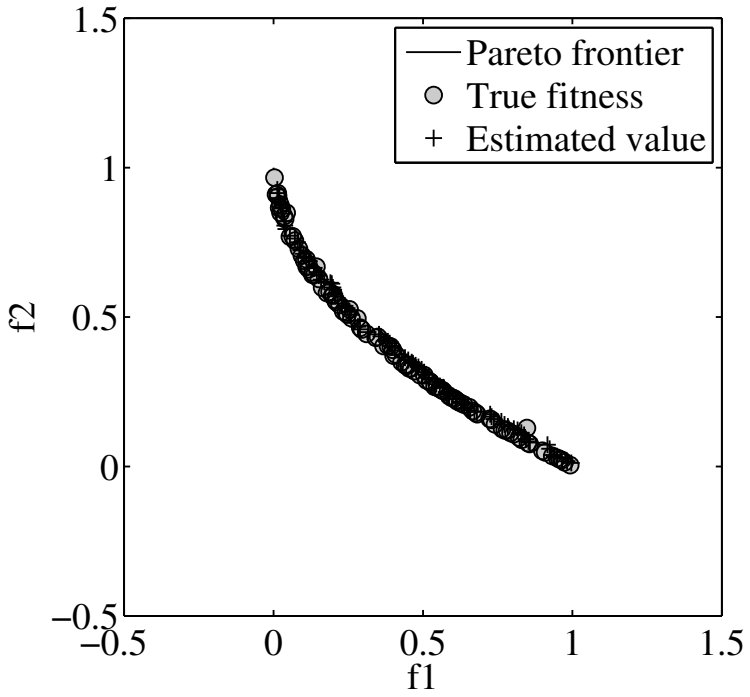


Figure 4.6: Population distribution with MFE-DSGA on Noisy ZDT1.

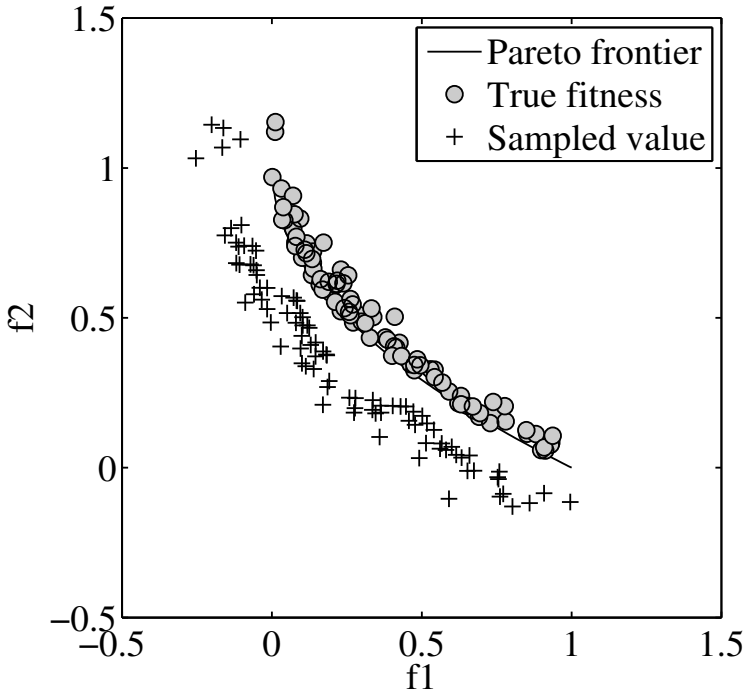


Figure 4.7: Population distribution with NSGA-II on Noisy ZDT1.

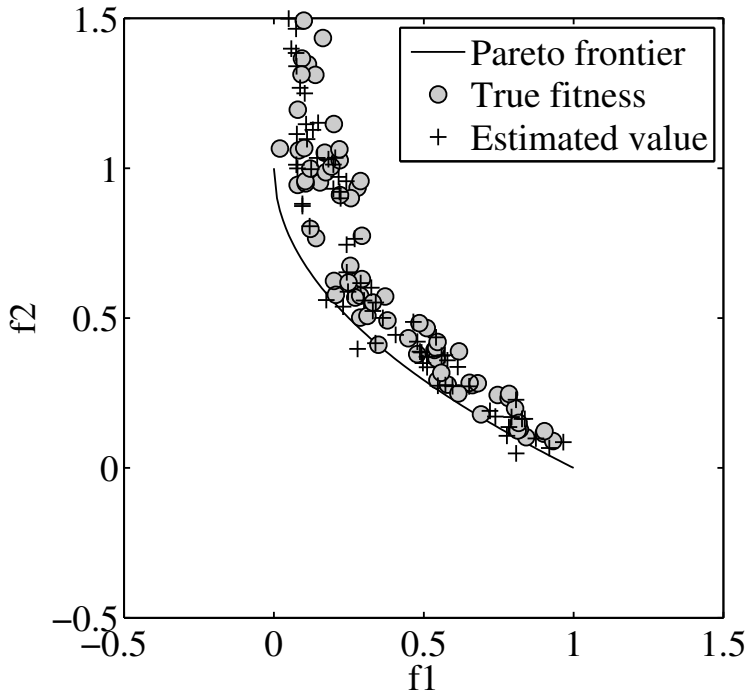


Figure 4.8: Population distribution with Sample-5 NSGA-II on Noisy ZDT1.

Objective Functions: HC (minimization), NO_x (minimization).

Decision Variables (Control Parameters): ignition timing (IGN), fuel-injection timing (INJ), intake valve timing (VT-IN), exhaust valve timing (VT-EX).

Generally, there is a trade-off between HC and NO_x. Also, changing the aforementioned decision variables, i.e., the engine control parameters, complicates the trade-offs. The HC and NO_x were measured at a constant air-fuel ratio and vehicle speed. Air-fuel ratio is ratio of air and gasoline mass, and is controlled by fuel-injection time of injector to achieve theoretical air-fuel ratio. Vehicle speed calculated by the computer is controlled by the opening angle of the throttle valve to track the desired speed.

MFE-DSGA were programmed into a calibration PC. In the calibration PC, the objective functions and decision variables were processed as normalized values. The calibration PC was connected to an ECU by a serial communication. An individual evaluation sequence is as follows:

1. Convert an evaluated individual into control parameters.

2. Transmit the control parameters to the ECU.

3. Receive the sampled data of the engine's output from the ECU and instruments for a prescribed period.

4. Calculate the estimated value of the true fitness for control parameters using the sampled data.

The experiment condition of MFE-DSGA is as follows: Population size $|P| = 50$ and offspring population size $|Q| = 8$ were used. The number of evaluations was 800 (parents in family population F was re-evaluated). The mean value of the sampled data was used as the estimated value. The evaluation time was about half a day.

MFE-DSGA employed UNDX as a crossover operator. The crossover rate was 1.0. Since UNDX shows good performance without mutation, we did not use mutation operator.

4.5.2 Discussion of Results

Figure 4.9 shows the population distribution of the estimated value in the objective function space. The horizontal axis indicates HC, and the vertical axis indicates NO_x respectively. Hence, the direction of optimization is in the lower left part of the graph. Each axis is shown in normalized scale. Moreover, Fig. 4.10 indicate population distribution in decision variable space. Each axis is also shown in normalized scale.

From Fig. 4.9, we can see that the approximation Pareto frontier shows convex shape. In addition, we can understand the properties of control parameters from Fig. 4.10:

- Ignition timing IGN was widely distributed. On the other hand, fuel-injection timing INJ and exhaust valve timing VT-EX converged in the vicinity of 0.9 and 1.0 respectively (see the upper right and left part of Fig. 4.10).
- A strong correlation exists between IGN and intake valve timing VT-IN. VT-IN approaches from 0.6 to 0.0 provided that the IGN moves from 0.3 to 0.8 (see the upper center part of Fig. 4.10).

Through discussion with engine calibration experts, we confirmed that the characteristic of the Pareto approximation set was appropriate as the performance of the engine. Hence, we can conclude that MFE-DSGA is effective for real engine calibration.

4.6 Summary

In this chapter, Memory-based Fitness Estimation and Distribution-based Selection GA (MFE-DSGA) was proposed to optimize MOPs that have observation noise as an uncertainty. First, the problems of conventional MOEAs for a noisy environment was introduced. Next, MFE-DSGA that is extension of MFEGA proposed by Sano and Kita [91, 93] for MOPs was proposed and its details were discussed. The effectiveness of the proposed method was validated through numerical experiments and a real engine experiment.

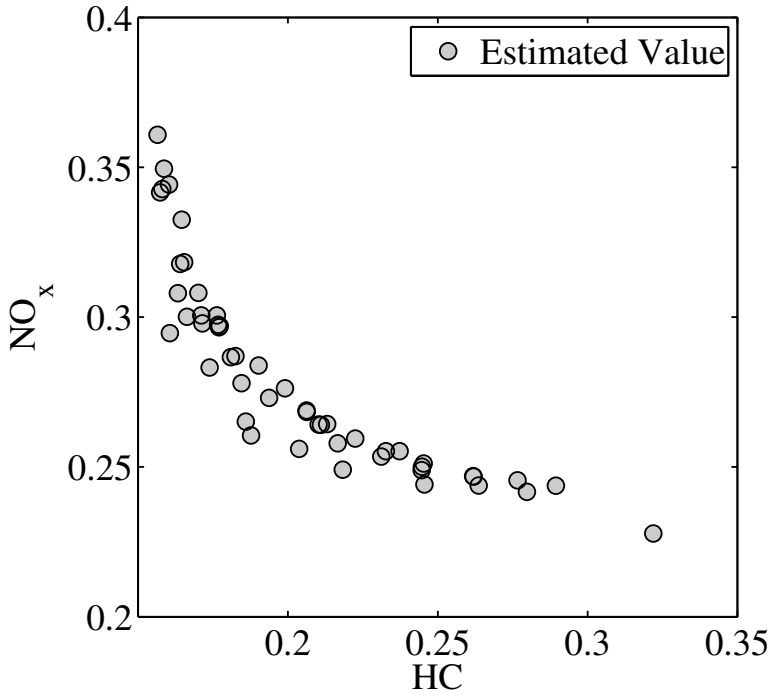


Figure 4.9: Population distribution on the objective function space in multi objective optimization of engine control.

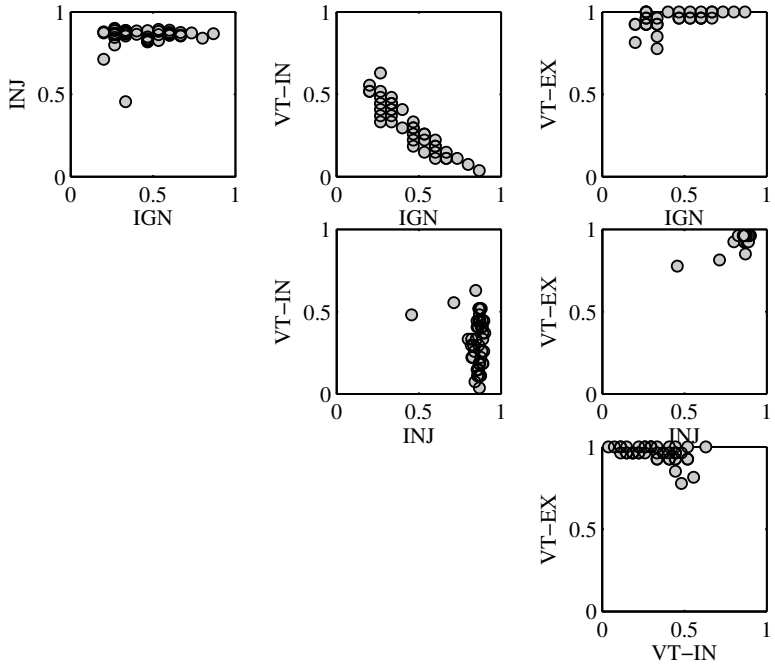


Figure 4.10: Population distribution on the decision variable space in multi-objective optimization of engine control.

Chapter 5

Crossover Operator for Periodic Functions

5.1 Outline

In engineering design, a timing optimization for an apparatus with a cycling mechanism such as that of an internal-combustion engine can be defined as an optimization problem of periodic function. For instance, the fuel injectors inject fuel into the cylinders once for 2 rotations of the crankshaft in a four stroke gasoline engine. As a result, the fuel injection timing 0 and 720 degrees BTDC (Before Top Dead Center) have the same combustion effect. Thus making the engine output become periodic functions for the fuel injection timing.

Additionally, the following real world applications including periodicity can be considered:

- Timing optimization for electric motors.
- Rotation pattern matching in image processing.
- Optimization of two corners in protein structure estimations.
- Position optimization of link mechanisms such as robotic arms.

In these problems, we can define continuous periodic function optimization problems that have 2π cycle when paying attention to one angular variable:

$$\min_{\theta} f(\theta) = f(\theta + 2N\pi), \theta \in \mathbf{R},\ N = 0, \pm 1, \pm 2, \ldots$$

To solve the aforementioned problems by using Genetic Algorithms (GAs), it is a common approach to convert to a non-periodic continuous function optimization problem by defining a domain that represents one cycle such as $[\theta_{\min}, \theta_{\max}] = [-\pi, \pi]$:

$$\min_{\theta} f(\theta), \quad \theta \in [\theta_{\min}, \theta_{\max}].$$

If we define a wider domain, plural optimal solutions that have the same fitness exist. Hence, GAs that are a multi-point search method based on crossover operators might search un-expected areas between the plural optimal solutions.

It is well known when GAs are applied to continuous function optimizations, the fact that Real-Coded GAs (RCGAs) that consider continuity of decision variables indicate higher optimization performance than bit-string GAs that use binary code [30, 31, 66, 80]. It can be expected that Unimodal Normal Distribution Crossover (UNDX) proposed by Ono et al. [80] and Blend Crossover (BLX-α) proposed by Eshelman et al. [30] have a higher search capability than One-point Crossover (1X) and Uniform Crossover (UX) for bit-string GAs [37], because they generate offspring by using interpolation and extrapolation based on continuousness of search space.

In this chapter, single-objective RCGAs are considered. A crossover operator to optimize periodic functions is proposed and its effectiveness is shown. At first, UNDX for Periodic function (UNDX-P), a new concept crossover operator for periodic functions, is proposed. Next, basic statistical properties of UNDX-P are considered. Additionally, the effectiveness of UNDX-P for periodic and non-periodic functions is demonstrated through numerical experiments.

5.2 Crossover for Periodic Functions

When a periodic function is optimized by RCGAs, the characteristics of the problem depend on the domain definition. Consider a unimodal periodic function depicted on the left side of Fig. 5.1. When the domain is defined through division in the vicinity of the optimal solution, this function has the global and a local optimal solution as shown on the right side of Fig. 5.1.

For these periodic functions, naive application of a RCGA faces two difficulties:

Sampling bias: Since most crossover operators of RCGAs such as BLX-α and UNDX often generate offspring in the vicinity of the center of the search space, it is difficult to optimize functions whose optimal solution exists near the boundary. The sampling bias can be reduced by defining a wider domain of search space for normal functions. However, it is difficult to apply this method to periodic functions because plural global optimal solutions appear [106, 101].

Evolutionary stagnation: When plural powerful local optimal solutions exist far apart, offspring are generated in the areas that step over them. As a result, the search stagnates [104].

5.2.1 Related Works

Tsutsui have proposed Boundary Extension by Mirroring (BEM) to reduce the influence of sampling bias [106]. This method introduces mirroring extension of the bound-

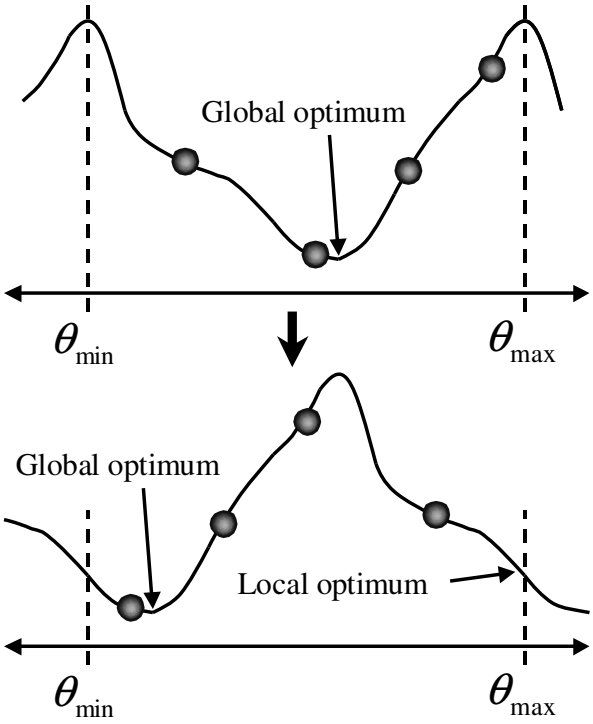


Figure 5.1: Landscapes of a periodic function by domains.

ary of function domain. However, evolutionary stagnation cannot be solved because local optimal solution cannot be excluded.

Someya and Yamamura have proposed Toroidal Search space Conversion (TSC) to solve this problem [101]. TSC can exclude the sampling bias by extending the mirror-copied domain and connecting it like torus to consider the continuousness of the boundary. The effectiveness of TSC has been demonstrated through the detailed comparative study for BEM. However, we have to make attention to the following two points:

- TSC requires a complex judgment process of the positions of parents for crossover operation.
- TSC leads a complexity of multimodality by extension operation of search space.

Tomobe et al. have proposed a method of generating offspring to supplementary angle domain, when the angle between parents becomes π or more for domain $[-\pi, \pi]$ [105]. However, following problems are considered:

- This method requires the judgment of supplementary angle for parents.
- It is difficult to apply to multi-parent crossover operators such as UNDX-m [67] and SPX [43].

Hence, a more generalized method is required.

In the following subsections, we first propose a 'crossover operator on hypersphere', and a 'crossover operator for periodic functions' as a special case of the crossover operator on hypersphere.

5.2.2 Crossover Operator on Hypersphere

$n-1$-dimentional unit hypersphere S^{n-1} on n-dimensional space $\mathbf{R}^n$ is defined as

$$S^{n-1} = \left\{ [x_1 \ \cdots \ x_n]^{\mathrm{T}} \in \mathbf{R}^n \ \middle| \ \sum_{i=1}^{n} x_i^2 = 1 \right\}. \tag{5.1}$$

A problem that obtains a n-dimensional unit direction vector can be generalized as a problem that the population distributes on S^{n-1}. For instance, a weighted sum vector decision problem for scalarization of plural objective functions is treated as such a problem. To solve these problems, we propose a simple extension of crossover operator on hypersphere. The algorithm is described below:

1. Select parents $\boldsymbol{x}^{\mathrm{p}}$ from the population distributed on S^{n-1}.
2. Apply a crossover operator for $\boldsymbol{x}^{\mathrm{p}}$ as points in $\mathbf{R}^n$ and generate offspring $\boldsymbol{x}^{\mathrm{c}\prime}$. If $||\boldsymbol{x}^{\mathrm{c}\prime}|| < \varepsilon$, re-generate offspring, where $||\boldsymbol{x}^{\mathrm{c}\prime}||$ is Euclidean norm of $\boldsymbol{x}^{\mathrm{c}\prime}$ and ε is a small positive value.
3. Normalize $\boldsymbol{x}^{\mathrm{c}\prime}$ by following equation:

$$\boldsymbol{x}^{\mathrm{c}} = \frac{\boldsymbol{x}^{\mathrm{c}\prime}}{||\boldsymbol{x}^{\mathrm{c}\prime}||}, \tag{5.2}$$

   and obtained offspring $\boldsymbol{x}^{\mathrm{c}}$ that is converted to the point on S^{n-1}.

We call the crossover operator on hypersphere by using UNDX [80] UNDX on Hypersphere (UNDX-H). Offspring generated by UNDX-H are distributed on the hypersphere. Hence it is able to search by RCGAs because the continuousness on hypersphere is kept.

5.2.3 Crossover Operator for Periodic Functions

Let us consider periodic function $f : \mathbf{R} \to \mathbf{R}$ that has the angle θ corresponding to a point $[\cos\theta \ \sin\theta]^{\mathrm{T}}$ on a one-dimensional unit-circle.

$$S^1 = \left\{ [x \ y]^{\mathrm{T}} \in \mathbf{R}^2 \ \middle| \ x^2 + y^2 = 1 \right\}. \tag{5.3}$$

We propose UNDX for periodic function (UNDX-P) based on UNDX-H to solve this problem. A conceptual diagram of UNDX-P is illustrated in Fig. 5.2, and the algorithm is as follows:

1. Obtain a point $[x^{\mathrm{p}}\ y^{\mathrm{p}}]^{\mathrm{T}} = [\cos\theta^{\mathrm{p}}\ \sin\theta^{\mathrm{p}}]^{\mathrm{T}}$ on a unit circle S^1 corresponding θ^{p} a variable of periodic function.

2. Apply UNDX for parents converted to points on a unit circle $S^1 = \{[x\ y]^{\mathrm{T}} \in \mathbf{R}^2 | x^2 + y^2 = 1\}$, and generate offspring $[x^{\mathrm{c}}\ y^{\mathrm{c}}]^{\mathrm{T}}$ as points on the two-dimensional space $\mathbf{R}^2$

3. Calculate the angular variable θ^{c} for offspring by using

$$\theta^{\mathrm{c}} = \begin{cases} \tan^{-1}\frac{y^{\mathrm{c}}}{x^{\mathrm{c}}} & \text{if } x^{\mathrm{c}} \geq \varepsilon \\ \mathrm{sgn}(y^{\mathrm{c}}) \cdot \left(\pi - \tan^{-1}\left|\frac{y^{\mathrm{c}}}{x^{\mathrm{c}}}\right|\right) & \text{if } x^{\mathrm{c}} \leq -\varepsilon \\ \mathrm{sgn}(y^{\mathrm{c}}) \cdot \frac{\pi}{2} & \text{if } |x^{\mathrm{c}}| < \epsilon,\ |y^{\mathrm{c}}| \geq \varepsilon \\ 0 & \text{if } |x^{\mathrm{c}}| < \varepsilon,\ |y^{\mathrm{c}}| < \varepsilon \end{cases}, \tag{5.4}$$

   where $\mathrm{sgn}(\cdot)$ is the signum function, and ε is a sufficiently small positive number. Generally, Eq. (5.4) is implemented in programming languages as a function called $\mathrm{atan2}(y, x)$.

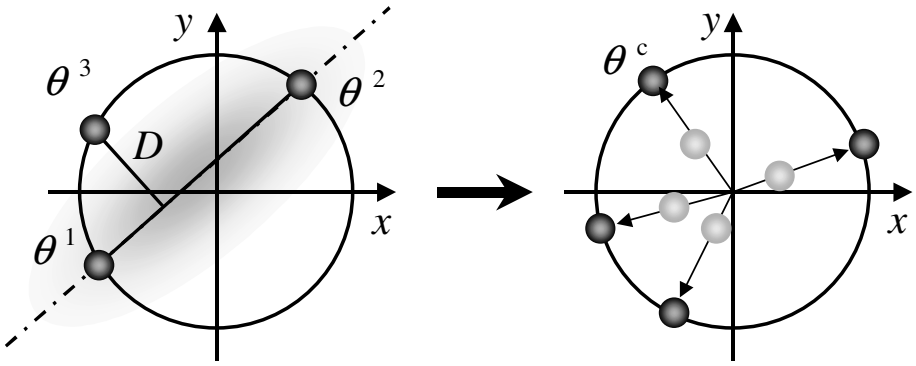


Figure 5.2: Conceptual diagram of UNDX for Periodic function (UNDX-P). UNDX is applied for three parents θ^1, θ^2 and θ^3 converted to the points on a unit circle S^1 (left). Angular variables θ_{c} of the generated offspring on the $\mathbf{R}^2$ are calculated by Eq. (5.4) (right).

When UNDX-P is applied for a n-dimensional periodic function $f : \mathbf{R}^n \to \mathbf{R}$ whose decision variable is n-dimensional angular vector $\boldsymbol{\theta} = [\theta_1\ \theta_2\ \cdots\ \theta_n]^{\mathrm{T}} \in \mathbf{R}^n$, the following procedure is executed.

1. Transpose each element of $\boldsymbol{\theta}$ to the point of S^1.

2. Apply UNDX for extended parent $[x_1\ y_1\ x_2\ y_2\ \cdots\ x_n\ y_n]^{\mathrm{T}} \in \mathbf{R}^{2n}$.

3. Calculate each angular variables.

When UNDX-P is applied to a periodic function $g : \mathbf{R} \to \mathbf{R}$ whose domain represents one cycle is $\phi \in (\phi_{\min}, \phi_{\max}]$, not $(-\pi, \pi]$, the following procedure is used.

1. Apply an affine transformation from $(\phi_{\min}, \phi_{\max}]$ to $(-\pi, \pi]$

$$\theta = 2\pi \cdot \frac{\phi - \phi_{\min}}{\phi_{\max} - \phi_{\min}} - \pi, \tag{5.5}$$

   before applying a crossover.

2. Apply an inverse affine transformation from $(-\pi, \pi]$ to $(\phi_{\min}, \phi_{\max}]$

$$\phi = (\phi_{\max} - \phi_{\min}) \cdot \frac{\theta + \pi}{2\pi} + \phi_{\min}. \tag{5.6}$$

By applying the crossover by the aforementioned procedure, a periodic function is optimized as a function that has substantially the same fitness landscape, and that does not depend on the domain definition. Therefore, since the problems such as sampling bias and evolutionary stagnation are avoided, improvement of the search performance can be expected.

5.3 Basic Property of UNDX-P

In this section, statistical properties of UNDX-P are investigated. Then, the distribution of the generated offspring is discussed.

5.3.1 Statistical Property

The statistical properties of offspring generated by UNDX have been investigated by Kita et al [66]. In UNDX-P, the following properties can be understood when a lot of offspring are generated from a lot of individuals by crossover operator.

Parents Uniformly Distributed on Unit Circle: This is a typical situation of the initial state of the search. In this case, the generated offspring are still uniformly distributed on the unit circle, because UNDX generates offspring from symmetric probabilistic density distribution.

Parents Converged Around Specific Angle: This is a typical situation of the last stage of the search to converge the population on a local or the global optimal solution. In this case, UNDX-P succeeds to the property of UNDX because a neighborhood of a specific angle of the unit circle can be seen as a straight line.

General statistical properties of UNDX-P excluding the aforementioned ones has been confirmed by the following numerical experiment. The steps of the experiment are described below:

1. Generate 10000 individuals uniformly randomly on the domain $(-\alpha\pi, \alpha\pi]$, shown in Fig. 5.3.

2. Apply UNDX to the individuals selected as parents randomly and generate offspring. In this operation, offspring (vector) are normalized to distribute them on the unit circle S^1.

3. Calculate the mean value and the standard deviation of x coordinates, and the standard deviation of y coordinates of the parents and offspring population as statistics.

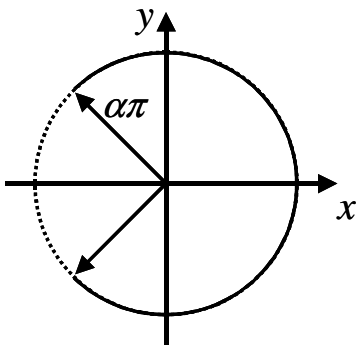


Figure 5.3: A method of generating parents.

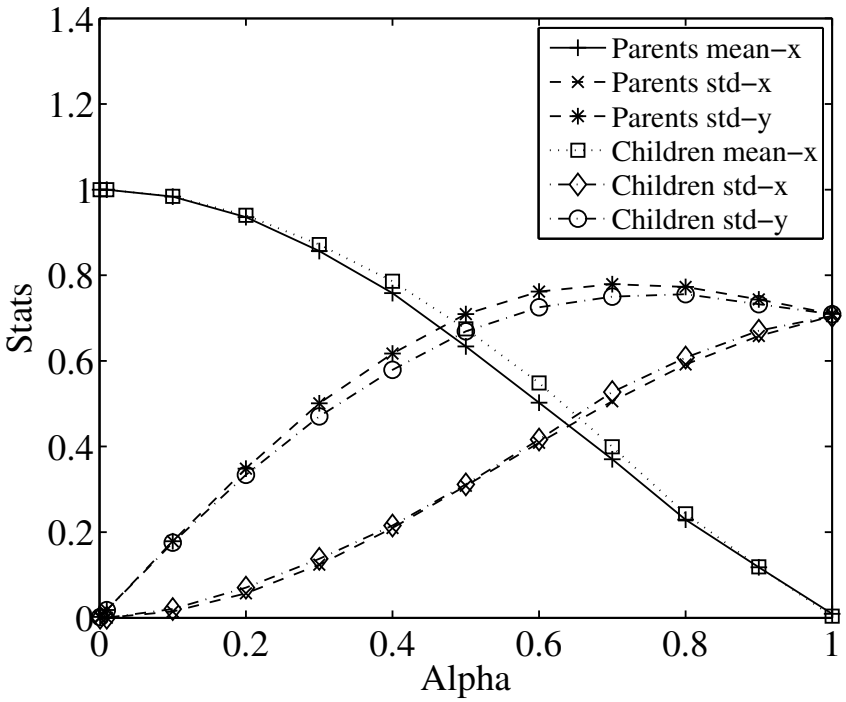
The result is shown in Fig. 5.4. Figure 5.4(a) shows each statistic for α. The mean value (parents: solid line with plus marker, offspring: dotted line with square marker) and standard deviation (parents: broken line with cross marker, offspring: dotted broken line with diamond marker) of x coordinate of offspring distribution is slightly larger than the parent distribution around from 0.2 to 0.8. Whereas, the standard deviations (parents: broken line with asterisk maker, offspring: dotted broken line with circle marker) of y coordinate are almost all the same. Figure 5.4(b) shows the same data on the logarithm scale. It is understood that each statistic of offspring distribution is corresponds well to the parent distribution in a minute angle of 0.1 or less. These results indicate that UNDX-P preserves the statistic of the parent distribution.

5.3.2 Generated Offspring Distribution

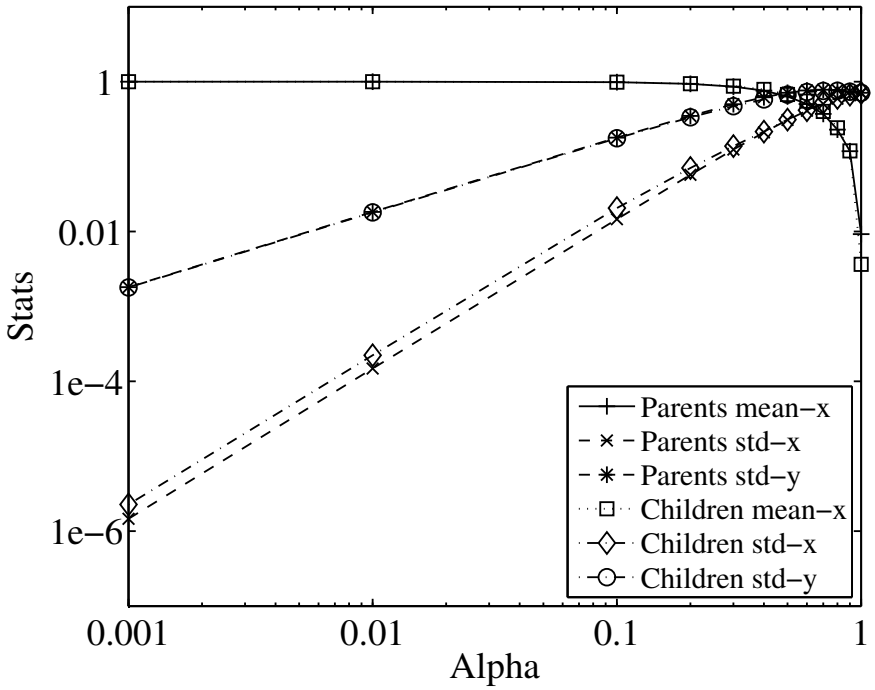
Next, the offspring distribution generated by UNDX-P was confirmed by the following numerical experiment. The parents were selected as follows, and UNDX-P generated 1.0×10^5 offspring:

$$\theta^1 = \alpha\pi,\ \theta^2 = -\alpha\pi,\ \theta^3 = 0,\ \alpha = 0.5, 0.3, 0.1.$$

The frequency distributions of the offspring were examined, where the domain $[-\pi, \pi]$ was divided into 100.



(a) Normal Scale



(b) Logarithmic Scale

Figure 5.4: Statistical measures of the UNDX-P.

The obtained histogram is shown in Fig. 5.5. When the distance between two parents is maximum on the S^1, that is $\alpha = 0.5$, the main search axis of UNDX runs through the center of unit circle of $\mathbf{R}^2$. Hence, the offspring are generated to contain the origin of $\mathbf{R}^2$. At that time, the offspring distribution generated by UNDX-P constructs bimodal distribution that centers on the parents θ^1 and θ^2 in S^1. In addition, offspring are generated to the entire domain. The main search axis leaves the center of the unit circle as the distance of the parents becomes small on S^1. Since the offspring comes to be distributed away from the center in $\mathbf{R}^2$, two peaks draw together gradually ($\alpha = 0.3$) and become the unimodal distribution ($\alpha = 0.1$).

Though the aforementioned discussions, it is confirmed that UNDX-P is a crossover operator which changes the offspring distribution from bimodal to unimodal depending on the positions of parents. Such a property is caused by the transformation operation from $\mathbf{R}^2$ to S^1. The offspring are generated around the neighborhood of the parents θ^1 and θ^2 instead of around the middle point of the parents by bimodal distribution when α is around 0.5. This property is desirable for search.

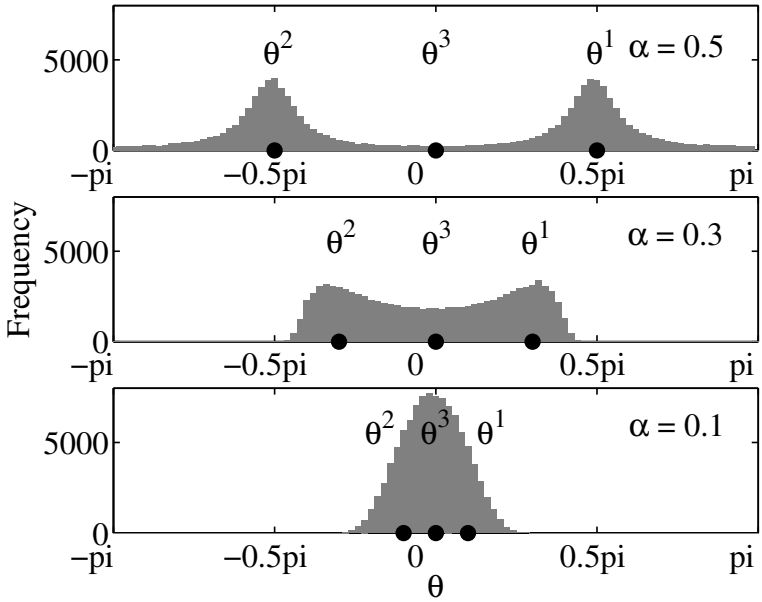


Figure 5.5: Frequency distributions of children by the UNDX-P.

5.4 Numerical Experiment of Periodic Function

5.4.1 Experiment Settings

In this section, the performance of UNDX-P is discussed through numerical experiment of periodic functions. The periodic functions used to experiment are shown below. All test periodic functions are minimization problems.

- Periodic Unimodal

$$f_{\mathrm{PU}}(\boldsymbol{\theta}) = n + \sum_{i=1}^{n} \cos(\pi\theta_i)$$
$$\theta_i \in [0, 2], \quad i = 1, \ldots, n.$$

This function has a unimodal landscape on the domain. The optimal solution is 0 at $\theta_i = 2N + 1,\ i = 1, \ldots, n,\ N = 0, \pm 1, \pm 2, \ldots$. The cycle is 2. In this experiment, $n = 10, 20$ are used.

- Periodic Multimodal

$$f_{\mathrm{PM}}(\boldsymbol{\theta}) = 10n + \sum_{i=1}^{n} (5\cos(\pi\theta_i) - 5\cos(10\pi\theta_i))$$
$$\theta_i \in [0, 2], \quad i = 1, \ldots, n.$$

This function has a multimodal landscape and a big-valley structure macroscopically on the domain. The optimal solution is 0 at $\theta_i = 2N+1,\ i = 1, \ldots, n,\ N = 0, \pm 1, \pm 2, \ldots$. The cycle is 2. In this experiment, $n = 4, 6$ are used.

- Fletcher and Powell

$$f_{\mathrm{FP}}(\boldsymbol{\theta}) = \sum_{i=1}^{n} (A_i - B_i)^2$$
$$A_i = \sum_{j}^{n} (a_{ij} \sin\alpha_j + b_{ij} \cos\alpha_j)$$
$$B_i = \sum_{j}^{n} (a_{ij} \sin\theta_j + b_{ij} \cos\theta_j)$$
$$\theta_i \in [-\pi, \pi], \quad i = 1, \ldots, n.$$

This function has a globally multimodal landscape [50] on the domain. The optimal solution is 0 at $[\alpha_1\ \alpha_2\ \cdots\ \alpha_n]$. There are plural optimal solutions that have the same fitness. Moreover, there are many local optimal solutions. The cycle is 2π. In this experiment, $n = 6, 9$ are used. The numerical value given in Bäck [3] as a constant $a_{ij}, b_{ij}, \alpha_i,\ i, j = 1, 2, \ldots, n$ is used.

The shapes of three periodic test functions are shown in Fig. 5.6.

To move the relative position of the global optimal solution in domain of each function, domain of f_{PU} and f_{PM} is defined by $\theta_i \in [0 + d, 2 + d]$, and domain of f_{FP} is defined by $\theta_i \in [-\pi(1+d), \pi(1+d)]$. d is the offset of the domain of periodic functions, and it takes $d = 0.0, 0.3, 0.6, 0.9$. The global optimal solution of f_{PU} and f_{PM} is the center of the domain if $d = 0$. Moreover, the global optimal solution approaches the

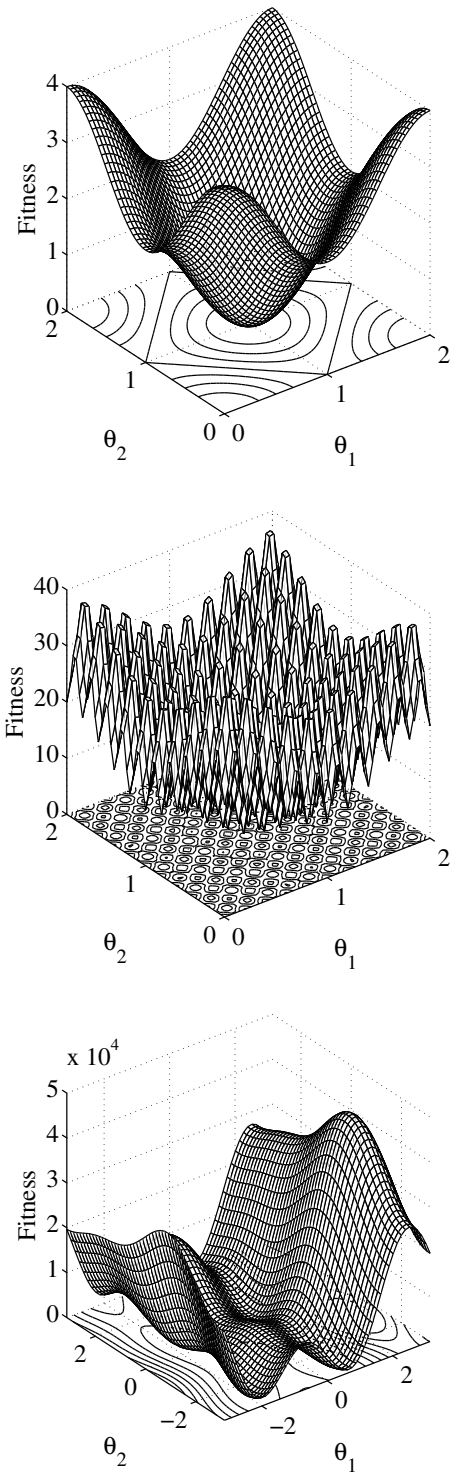


Figure 5.6: Shape of the periodic test functions (upper: Periodic Unimodal, middle: Periodic Multimodal, lower: Fletcher and Powell (2-dimension)).

boundary of the domain if d approaches 0.9. It is thought that the position of the global optimal solution of f_{FP} complexly changes depending on d. UNDX is used for comparison of UNDX-P in the numerical experiment. In UNDX, when an individual is generated at outside of domain, the individual is re-generated until it exists in the domain. This method is called **NoExt**. In numerical experiment, offset d was changed for UNDX-P and conventional method, and each different d was optimized 100 times. The performance of two crossover operators was evaluated by the number of optimization and transition of convergence successes.

In this experiment, it is assumed that an optimal solution was discovered if the function value becomes less than 1.0×10^{-7}. Minimal Generation Gap (MGG) proposed by Sato et al. [95] was used as a generation alternation. The population size $|P| = 100$ and the offspring population size $|Q| = 50$ were used. The crossover rate was 1.0. The maximum number of evaluations was 1.0×10^6.

5.4.2 Discussion of Results

The number of discovered global optimal solution in-100 trials is shown in Table 5.1. The bold-font shows the least number of discovered optimal solution for d of each problem.

Table 5.1: Performance comparison of the two methods for the three periodic test functions.

Function (Dimension)		UNDX-P				NoExt			
		0	0.3	0.6	0.9	0	0.3	0.6	0.9
Periodic Unimodal	10	100	100	100	100	100	100	100	**6**
	20	100	100	100	100	100	100	99	**0**
Periodic Multimodal	4	100	100	100	100	100	96	73	**65**
	6	98	100	**95**	100	98	37	13	**7**
Fletcher and Powell	6	100	**99**	100	100	100	100	75	**27**
	9	53	46	**44**	45	12	**9**	29	81

In Table 5.1, the number of discovered optimal solution of NoExt of f_{PU} and f_{PM} decreases as the position of optimal solution approaches the boundary of the domain, that is, d approaches 0.9. Whereas the result of UNDX-P was stable for changing d. Moreover, the transitions of the average of best fitness of the trial that discovered global optimal solution for f_{PU} ($n = 20$) and f_{PM} ($n = 6$) are shown in the upper part and the middle of Fig. 5.7. The left column shows the results of UNDX-P, and the right column shows the results of NoExt. For all test functions, it can be confirmed that the convergence speed of UNDX-P was the same not depending on d, although the convergence speed of NoExt slowed as d approached 0.9. These results indicate

that UNDX-P has substantially optimized the function that have the same landscape.

UNDX-P is hardly influenced by d, and has discovered the optimal solutions by all trials of f_{FP} ($n = 6$), although the number of discovered optimal solutions of NoExt decreased as d approached 0.9. In f_{FP} ($n = 9$), UNDX-P stably discovered optimal solutions through around 50 trials though the number of discovered optimal solutions of NoExt changed greatly. Hence, UNDX-P can stably optimize without dependence on the domain. The transitions of the average of best fitness of the trial that discovered global optimal solutions for f_{FP} ($n = 9$) are shown at the lower part of Fig. 5.7. Because the obtained optimal solutions were different at each trial, the convergence speed varied greatly in the last stage of the search. However, the convergence speeds of UNDX-P for each d were almost all same until 2.0×10^5 evaluations. Hence, it is understood that UNDX-P could stably search until converging to a neighborhood of plural global optimal solutions.

In the optimization of f_{FP}, two global optimal solutions and five local optimal solutions of $n = 6$, and eight global optimal solutions and 27 local optimal solutions of $n = 9$ were obtained by 800 trials (= 2 methods $\times$ 4 offsets $\times$ 100 trials), respectively. The positions and the discovered number of global optimal solutions and the powerful local optimal solutions are shown in Table 5.2. In Table 5.2, substantially equal solution of which position is different by the domain definition is counted as a same solution, The bold-font shows the maximum number of discovered solution of each condition.

At first, we examine the whole trend of obtained solutions. In UNDX-P, the tendency of obtained solution indicated almost the same without depending on the domain. For instance, Opt1 and Opt2 of $n = 6$ were obtained almost half each. Additionally, half the number of obtained solution of $n = 9$ was Loc1 that was a powerful local optimal solution. In contrast in NoExt, the tendency of obtained solution was varied. The performance of NoExt was dependent on the domain setting. NoExt could discover an optimal solution with a higher possibility than UNDX-P in one case, and then decreased the performance in another case. These phenomena are caused by the sampling bias that becomes easy to obtain the solution that exists in the vicinity of the center of domain. Hence, it is understood that the performance of NoExt depends on the domain setting.

Next, we examined the distribution of local optimal solution. Loc2 that was the most obtained solution on $n = 9$, $d = 0.6$ is the powerful local optimal solution of the Opt2 neighborhood. Loc2 has a element $\theta_3 = -1.256637$ that equals to the lower bound of domain $-\pi + 0.6\pi = -1.256637$. Opt2 is a known optimal solution defined by $[\alpha_1 \ \cdots \ \alpha_9]^{\mathrm{T}}$, that has element $\alpha_3 = -1.283410 < \theta_3$. Hence, it is considered that Loc2 is a local optimal solution generated on the periodic boundary. Similarly, four local optimal solutions of $n = 6$ and 22 optimal solutions of $n = 9$ were also generated solutions on the periodic boundary because they have at least one element equal to the boundary of the domain. In the powerful local optimal solutions shown in Table 5.2, Loc1 of $n = 6$ and Loc2, Loc3, and Loc6 of $n = 9$ were such local optimal solutions.

Note that these optimal solutions are obtained only by a specific offset of NoExt.

From the aforementioned discussions, we can understand that the positions of global and local optimal solutions are complexly moved by definition of the domain, if crossover operators of RCGAs that do not take into account the continuousness of periodic boundary are applied for periodic functions. When the domains are defined, a lot of new local optimal solutions are generated on the periodic boundaries. As a result, new local optimal solutions increase the difficulty of optimization. For this difficulty, UNDX-P does not need to care about the definition of a domain that makes new local optimal solutions by phase cutting. Hence, we can conclude that UNDX-P is an efficient crossover operator for periodic function optimization.

Table 5.2: Comparison of the solutions which is optimized by the two methods.

$n = 6$			UNDX-P				NoExt			
Name	Position	Fitness	0	0.3	0.6	0.9	0	0.3	0.6	0.9
Opt1	(0.436, 0.551,$\cdots$)	0	47	47	41	44	0	28	29	13
Opt2	(0.630, 0.693,$\cdots$)	0	**53**	**52**	**59**	**56**	**100**	**72**	**46**	14
Loc1	(0.428, 0.551,$\cdots$)	5.14	0	0	0	0	0	0	10	0
Loc2	(1.630, 0.976,$\cdots$)	810.17	0	1	0	0	0	0	12	**68**
$n = 9$			UNDX-P				NoExt			
Name	Position	Fitness	0	0.3	0.6	0.9	0	0.3	0.6	0.9
Opt1	(0.164, 0.585,$\cdots$)	0	16	10	12	5	3	1	4	1
Opt2	(0.436, 0.551,$\cdots$)	0	5	6	4	9	1	2	6	8
Opt3	(1.028, 0.679,$\cdots$)	0	2	3	1	0	0	0	2	15
Opt4	(1.207, 1.797,$\cdots$)	0	3	6	6	7	2	0	1	3
Opt5	(1.307, 1.115,$\cdots$)	0	6	8	3	2	2	0	4	11
Opt6	(1.671, 1.925,$\cdots$)	0	1	1	3	3	0	1	12	4
Opt7	(1.705, 1.708,$\cdots$)	0	12	11	13	14	4	4	0	1
Opt8	(1.851, 1.989,$\cdots$)	0	8	1	2	5	0	1	0	**38**
Loc1	(0.319, -1.827,$\cdots$)	1.28	**44**	**52**	**51**	**53**	4	4	2	5
Loc2	(0.456, 0.560,$\cdots$)	7.98	0	0	0	0	0	0	**26**	0
Loc3	(0.276, 0.656,$\cdots$)	107.92	0	0	0	0	0	23	0	0
Loc4	(0.937, -1.583,$\cdots$)	246.65	0	0	0	0	0	**28**	1	0
Loc5	(0.953, 0.344,$\cdots$)	248.77	3	2	5	1	**63**	23	4	0
Loc6	(0.949, -1.257,$\cdots$)	409.54	0	0	0	0	0	0	22	0

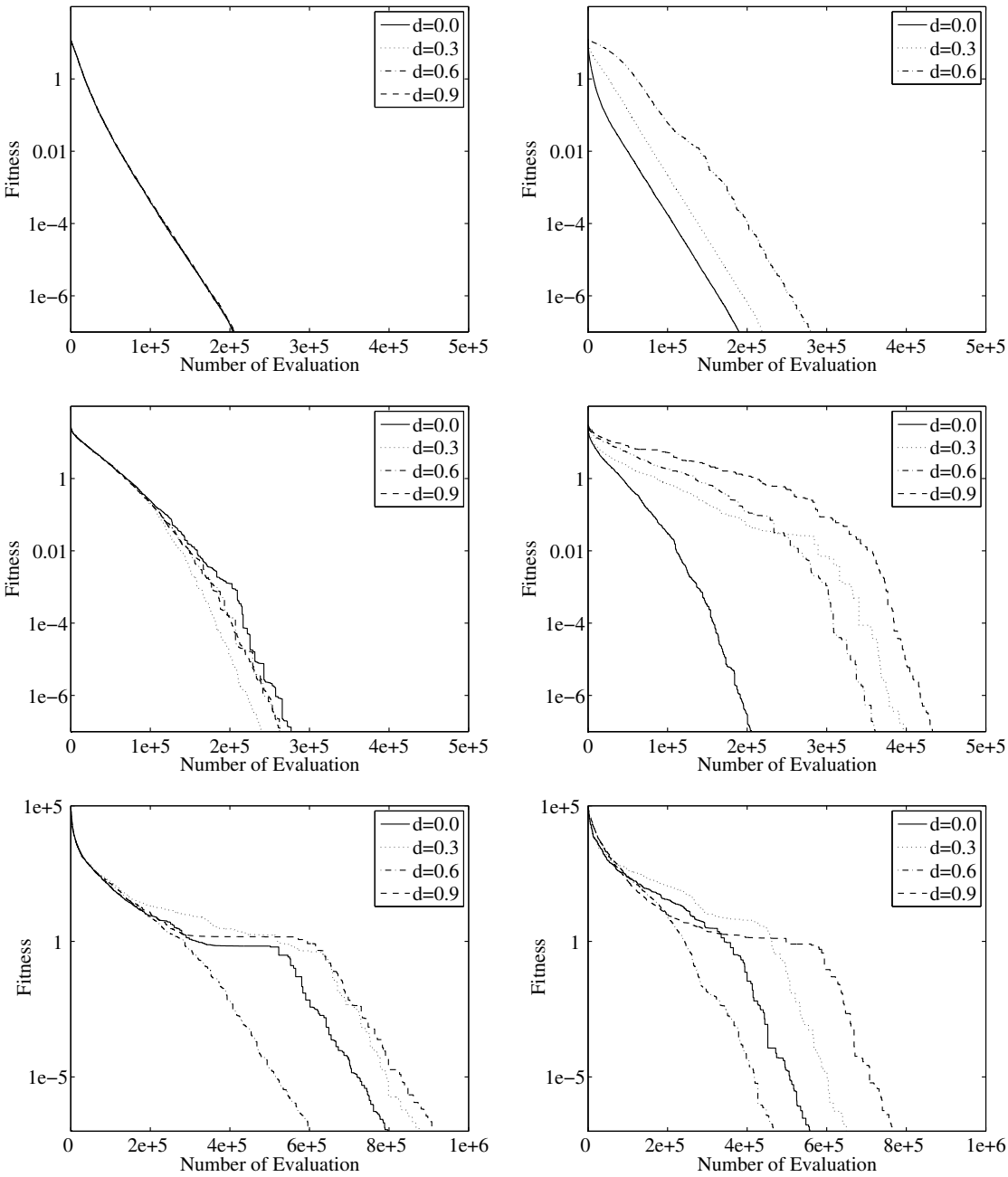


Figure 5.7: Convergence of the two methods (left: UNDX-P, right: NoExt) for the three periodic test functions (upper: f_{PU}, middle: f_{PM}, lower: f_{FP}).

5.5 Numerical Experiment of Non-periodic Function

In this section, the performance of UNDX-P for non-periodic functions is discussed. As previously discussed, crossover operators of RCGAs such as UNDX have sampling bias. Hence, it is known that optimization becomes difficult when the optimal solution is in

the vicinity of the boundary of the domain [31, 101]. Toroidal Search space Conversion (TSC) proposed by Someya and Yamamura [101] is one of the techniques used to cope with this problem. While, UNDX-P does not have the same level of sampling bias as TSC because of the mechanism of offspring generation. Moreover, UNDX-P does not need the judgment process of parents position. Hence, it is possible to easily apply it both periodic functions and non-periodic functions, while expecting a high level of effectiveness.

5.5.1 Experiment Settings

The effectiveness of UNDX-P is discussed through a numerical experiment.

- Rastrigin

$$f_{\text{Rastrigin}}(\boldsymbol{x}) = 10n + \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i))$$
$$x_i \in [-5.12, 5.12],\ i = 1, 2, \ldots, n.$$

  This function is a non-periodic multimodal function that has big valley structure. $f_{\text{Rastrigin}}$ takes minimum value 0 by coordinates $[0\ \cdots\ 0]$. In this numerical experiment, the number of dimension is $n = 4, 6, 8$.

The domain of $f_{\text{Rastrigin}}$ is defined as $x_i \in [-5.12 + d, 5.12 + d]$, to change the relative position of the global optimal solution in the domain as well as the experiment in the section above. $d = 0.0, 1.5, 3.0, 4.5$ are offsets of the domain. The position of the global optimal solution is the center of domain when $d = 0$, and it approaches to the boundary of domain when d approaches 4.5. In a numerical experiment, UNDX-P was applied to offset Rastrigin functions by following two methods:

1. UNDX-P is applied directly disregarding discontinuity in the boundary of the domain.

2. UNDX-P is applied after mirroring extension to solve discontinuity in the boundary of the domain. The conceptual diagram of the extended search space is shown in Fig. 5.8. The function value of the extended search space is calculated by the following equation:

$$f(\boldsymbol{x}) = f(\boldsymbol{y}) \tag{5.7}$$
$$\boldsymbol{y} = [y_1\ y_2\ \cdots\ y_n]^{\mathrm{T}} \tag{5.8}$$
$$y_i = \begin{cases} x_i & \text{if } x_i \leq x_{\max} \\ 2x_{\max} - x_{\min} - x_i & \text{if } x_i > x_{\max}. \end{cases} \tag{5.9}$$

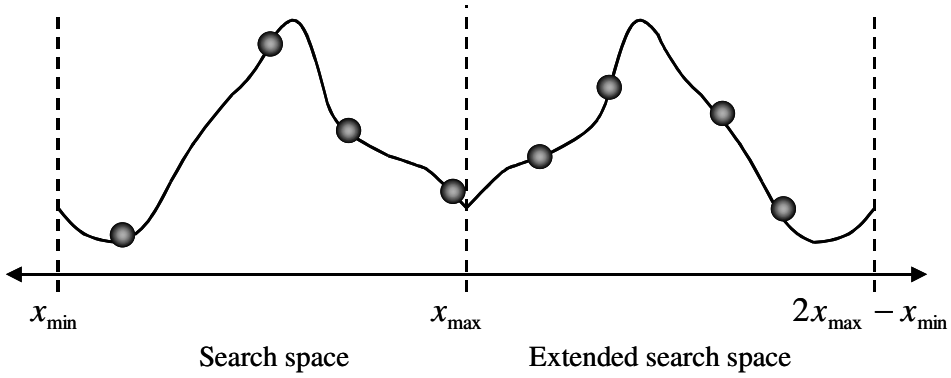


Figure 5.8: Extended search space for the UNDX-P-Ext.

In this thesis, method 1 is called UNDX-P and method 2 is called UNDX-P-Ext respectively. When UNDX-P is applied to a non-periodic function, the sampling bias can be exterminated without expanding the search space. However, it might have negative effect on the search performance if magnitude of boundary is extremely discontinuous, although it is not problem in a periodic function because the connected boundary is smooth. Whereas UNDX-P-Ext does not need the parents position judgment that TSC needs. However, the multimodal complexity might be increased by search space extension.

To compare it with the result shown in Someya's thesis [102], numerical experiments of UNDX-P and UNDX-P-Ext were executed under the same conditions. Population size $|P| = 100$ and offspring population size $|Q| = 50$ were used. the crossover rate was 1.0. Minimal Generation Gap (MGG) was used as the generation alternation model. The termination condition was defined as 1.5×10^6 evaluations or the best fitness achieving 1.0×10^{-7}. The number of trials of each condition was 100.

5.5.2 Discussion of Results

The result of our experiment is shown in Table 5.3. In the table, the results excluding UNDX-P and UNDX-P-Ext were citations from Someya's thesis [102]. Explanation of cited methods are as follows:

NoExt: Method that re-generates offspring if the offspring generates outside of the domain.

BEM: Boundary Extension by Mirroring proposed by Tsutsui [106]. Initial population is generated in original domain. The extension rate is 0.25.

BEMe: BEM that generates an initial population in extended domain.

TSC: Toroidal Search space Conversion proposed by Someya et al. [101]

Table 5.3: Performance comparison of the Rastrigin function for six methods, where the results of NoExt, BEM, BEMe and TSC are a quotation from the thesis of Someya [102].

	UNDX-P				UNDX-P-Ext				NoExt			
	0.0	1.5	3.0	4.5	0.0	1.5	3.0	4.5	0.0	1.5	3.0	4.5
4	100	100	100	**86**	100	100	100	100	100	99	94	**26**
6	85	87	89	**9**	98	62	**42**	100	100	67	20	**1**
8	65	59	40	**0**	56	6	**5**	94	84	27	3	**0**
	BEM				BEMe				TSC			
	0.0	1.5	3.0	4.5	0.0	1.5	3.0	4.5	0.0	1.5	3.0	4.5
4	100	100	99	**86**	100	97	98	**87**	100	100	100	100
6	99	77	57	**13**	98	84	80	**8**	96	79	**44**	100
8	91	26	10	**0**	79	34	22	**1**	79	35	**5**	97

The numbers in the Table 5.3 indicate the number of discovered optimal solutions, and bold-font numbers show the worst value in same n, when d is changed.

The discovered number of optimal solutions of UNDX-P decreased when the optimal solution approaches the vicinity of the boundary. In this point of view, the tendency of UNDX-P is similar to NoExt, BEM and BEMe. However, UNDX-P indicated the smallest deterioration and the best performance in six methods on $d = 1.5$ and $d = 3.0$. Therefore, it is understood that UNDX-P has stable search performance for the position of optimal solution. However, the performance of $d = 0.0$ is more inferior than other methods. UNDX-P does not have sampling bias in the early stage of the search and it generates offspring for any area evenly, while other methods generates offspring in the vicinity of the center of the domain due to sampling bias. The landscape of $d = 4.5$ does not have global smoothness, because of the discontinuous of boundary is extremely large. It is thought that this fact leads to the poor performance of $d = 4.5$. The tendency of UNDX-P-Ext was similar to TSC:

- The performance at $d = 4.5$ is very good.
- Since the extended landscape of $d = 4.5$ has a big valley structure globally, the search is easy. Whereas, the performance at $d = 1.5$ and $d = 3.0$ is extremely bad.

In these conditions, search becomes difficult because of increasing local optimal solution and complexity of multimodality. Hence, UNDX-P is an effective crossover operator for non-periodic function optimizations, because of a steady search performance for the position of optimal solution and easy implementation.

5.6 Summary

In this chapter, to solve continuous periodic function optimization problems which often appear in engineering problems, UNDX for Periodic function (UNDX-P) was proposed. At first, a crossover on hypersphere called UNDX on Hypersphere (UNDX-H), and a crossover for periodic functions called UNDX-P that is the special case of UNDX-H were defined. Next, through investigation of statistical property of UNDX-P, it was indicated that UNDX-P preserves statistics roughly. Moreover, it was demonstrated that UNDX-P can stably optimize periodic functions not to depend on the domain of function through numerical experiments. UNDX-P was also applied to non-periodic optimization problems because it does not have sampling bias in the early stage of the search. The search performance showed that it was comparatively robust for the position of optimal solution.

In Chapter 6 and 7, this method will be applied to a real world application, that is, engine calibration to confirm the effectiveness.

The concept of crossover operator for periodic functions can easily be applied to crossover operators of RCGAs other than UNDX. For instance, it is an interesting study to investigate the performance of UNDX-m-P that is extended UNDX-m proposed by Kita et al. [67] for periodic functions.

Chapter 6

Individual Evaluation Scheduling

6.1 Outline

Experiment-based optimization has to be carried out under uncertainty such as system and observation noise within a quite limited evaluation time which is restricted by the operation time and the durability of machine. Particularly, if the optimized object is a dynamical system, we have to wait until the transient response caused by the switching of system parameters is diminished to reduce the impact on the observed performances. Since we have to apply many parameter candidates distributed widely as population, we have to manage such problems in evolutionary approach, especially in MOEAs.

In this chapter, an influence of dynamics for search performance is focused. For reduction of loss time caused by such transient response in evaluation of criteria, two techniques called Evaluation Order Scheduling and Evaluation Time Scheduling are proposed. Numerical experiments using a formal test problem and experiment in a HILS environment for a real internal-combustion engine demonstrate the effectiveness of the proposed methods.

6.2 Influence of System Dynamics

Figure 6.1 shows a conceptual diagram of influence by uncertainty and transient response added to observed value through experiments. It can be considered that the observed value through experiments is composed by true output and following two elements:

- Uncertainty caused by observation noise that is added to measurement instruments.
- Influence of transient response to change of control parameters caused by system dynamics.

As an example of transient response caused by dynamics, transition of engine output torque with control parameter switching is shown in Fig. 6.2. We can see that the parameter switching leads to transient response of output torque.

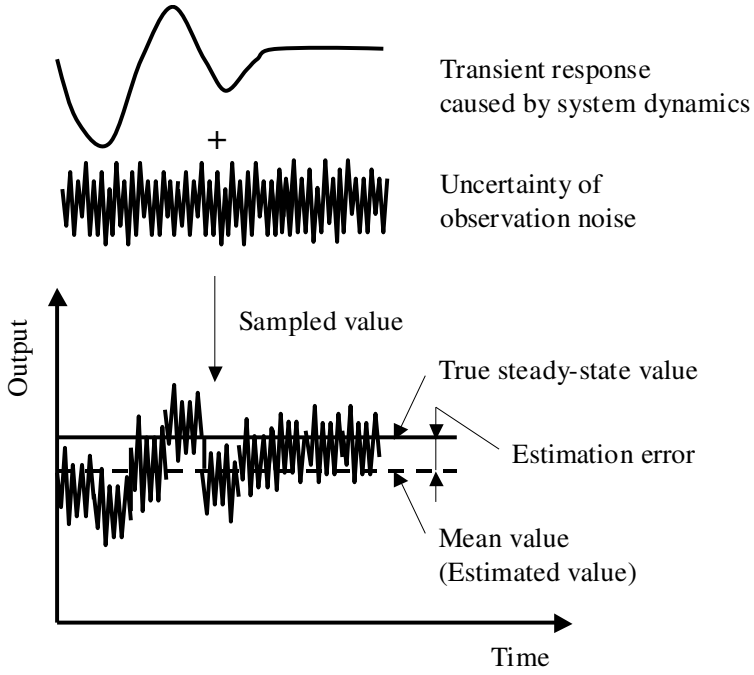


Figure 6.1: Conceptual diagram of uncertainties in estimated value.

When we estimate the 'true' steady-state output from an observation value of such a system, the general approach is as follow:

1. Wait on the observation value until it becomes stable.
2. Obtain the plural sampled value from observation.
3. Calculate the mean value of the sampled values as the estimated value of true steady-state output.

However, the aforementioned two elements generate estimation error between the true steady state output and estimated value.

If we can use a sufficiently long evaluation time, the estimation error will be small. However, when the MOEAs are applied for the optimization of system parameters of dynamical systems, the following dilemmas are caused:

- The performance of an individual should be measured after it settles enough to evade the influence of the transient response.
- The measurement time of an individual should be shortened as much as possible since MOEAs require a large number of evaluations under the limited total evaluation time.

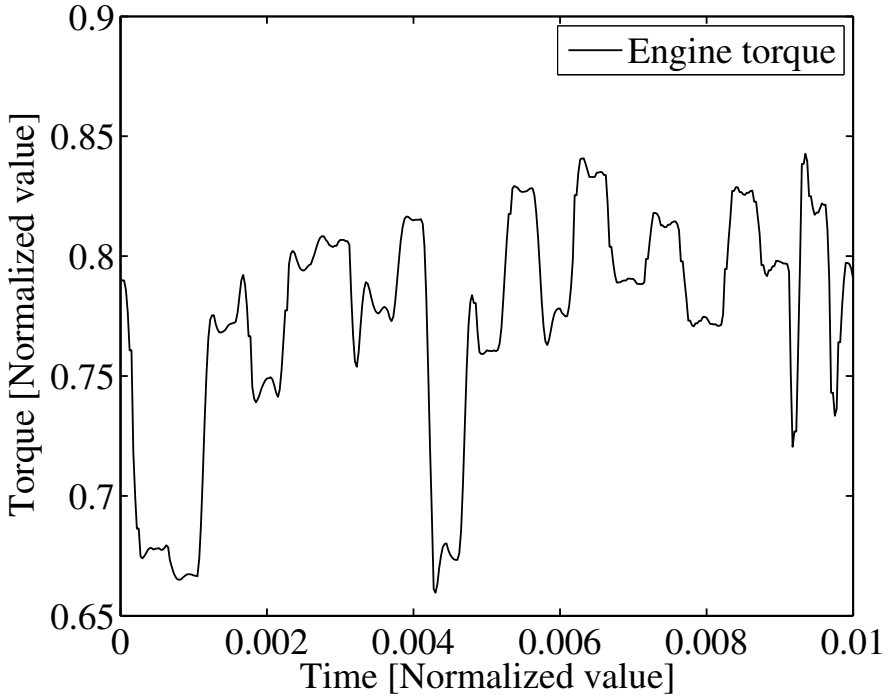


Figure 6.2: Transient response of an engine torque.

The waiting time of evaluation to diminish the influence of transient response is called Invalid Evaluation Time (IET). Fig. 6.3 shows a conceptual diagram of the invalid evaluation time generated by the transient response due to the switching of parameters. In fact, we can confirm the transient response of an engine torque caused by parameter switching from Fig. 6.4.

In this chapter, we propose a method called Individual Evaluation Scheduling (IES) to improve the performance of experiment-based optimization of a system having dynamics. IES consists of two ideas, i.e., Evaluation Order Scheduling (EOS) and Evaluation Time Scheduling (ETS). The former one is a technique used to decide evaluation order in the population to improve the accuracy of the performance by reducing the total magnitude of parametric change. The latter one is used to adjust the waiting time for the transient response. Figure 6.5 is a conceptual diagram of the effect of EOS and ETS on transient response caused by parameter switching.

6.3 Evaluation Order Scheduling

Let us consider a target to be optimized which is a stable dynamical system:

$$\dot{\boldsymbol{q}} = f(\boldsymbol{q}, \boldsymbol{x}), \tag{6.1}$$

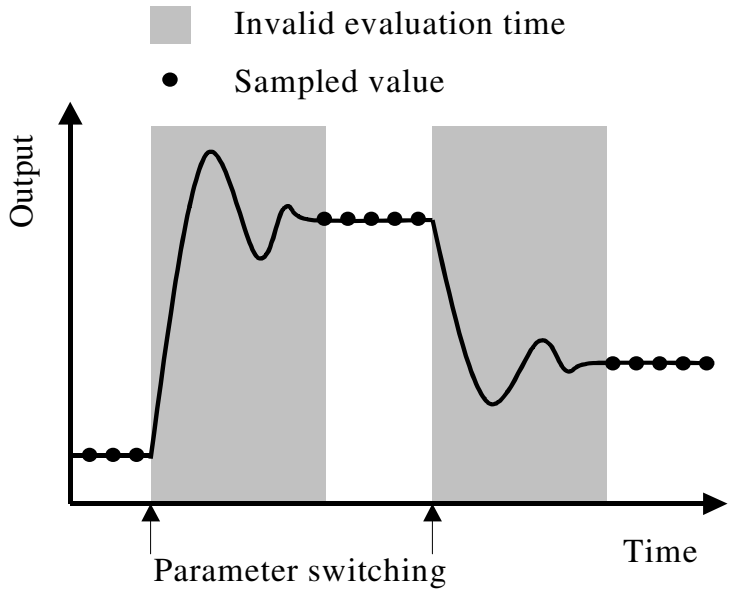


Figure 6.3: Invalid evaluation time.

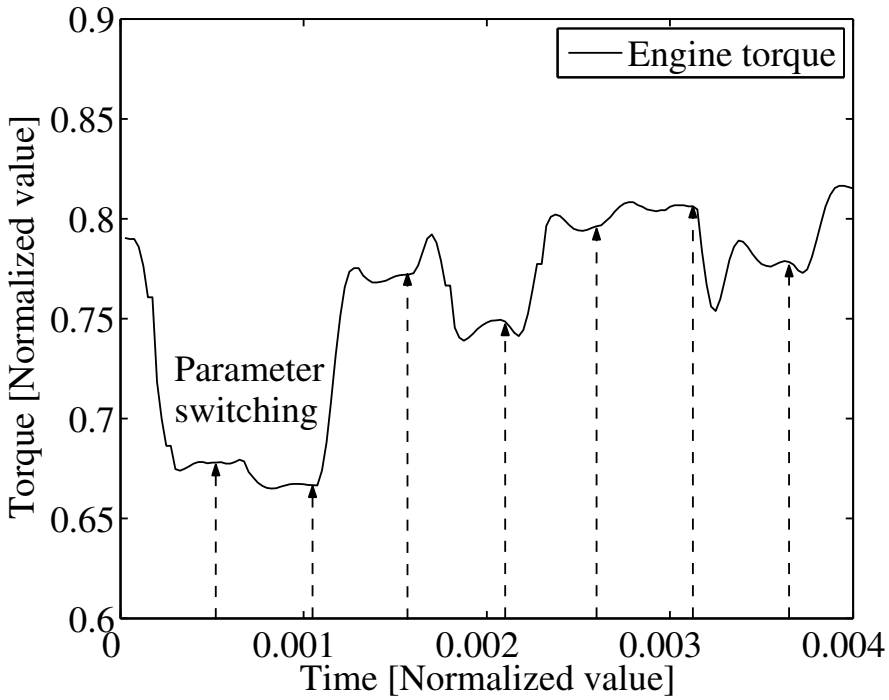


Figure 6.4: Transient response of an engine torque caused by parameter switching.

where $\boldsymbol{q} = [q_1\ q_2\ \cdots\ q_k]^{\mathrm{T}}$ is a state variable vector, and $\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_n]^{\mathrm{T}}$ is a system parameter vector represented by an individual of MOEAs and switched at prescribed intervals. In the following, we consider only the influence of transient response caused

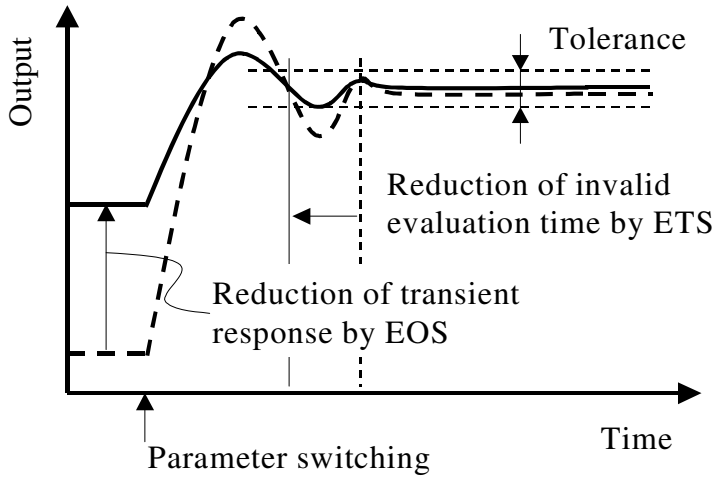


Figure 6.5: Conceptual diagram of the effect of the EOS and the ETS for transient response caused by parameter switching.

by system dynamics, and here in referred to as 'estimation error'.

It can be expected that the nearer parameter change makes the smaller equilibrium point changes excluding nonlinear phenomena such as bifurcation. As a result, the transient response can be expected to be suppressed, and if the system settles at its steady state faster, the accuracy of the estimated value can be improved. Hence, the problem is to find the evaluation order of the population that achieves the minimal total magnitude of parametric change. It can be formulated as problem similar to that of the Traveling Salesman Problem (TSP). For a TSP, when a certain tour shown in Fig. 6.6 is given, a tour replaced two arbitrary edges is called a 2-opt neighborhood. The local search using 2-opt neighborhood is known as a simple but effective heuristics of the TSP. Note that we have to find the shortest path with the starting point determined by the system parameter used in the current operation. EOS algorithm based on the 2-opt method is described as follows:

1. Define a population whose order is optimized as $P^0 = \{\boldsymbol{x}^0, P\}$, where $P = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^N\}$ as the population to be evaluated, N is the number of individuals, and $\boldsymbol{x}^0$ is an individual which was evaluated at last in the previous generation. Note that the order of $\boldsymbol{x}^0$ is fixed at the first one.

2. Define a permutation of P as $Z = (z_1, z_2, \ldots, z_N)$, and initialize Z.

3. Calculate the path length d_{total} by

$$d_{\text{total}} = \sum_{i=1}^{N} d_{z_{i-1},z_i}, \tag{6.2}$$

where $z_0 = 0$, $d_{z_{i-1},z_i} = \sqrt{\sum_{l=1}^{n} w_l (a_l^{z_{i-1}} - a_l^{z_i})^2}$, and w_l is the weight. It should

be noted that we do not need to obtain a closed path, and therefore we exclude the length return to $\boldsymbol{x}^0$.

4. Examine the path length of 2-opt neighborhoods of P^0 given by Z.

5. If there exists a path in the 2-opt neighborhood whose d_{total} is shorter than that of the current path, employ it as a new path, and then return to Step 4); otherwise, read out Z as a locally optimal permutation, that is, the evaluation order of P.

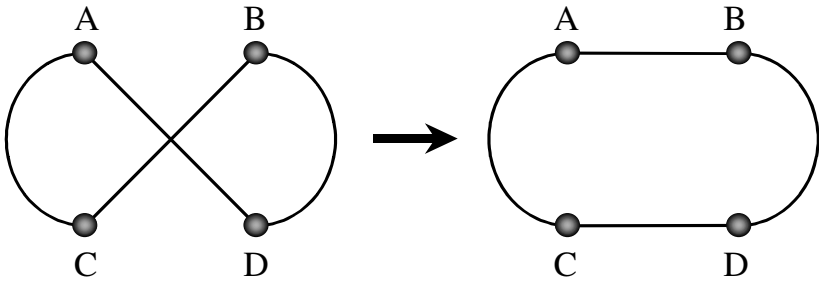


Figure 6.6: 2-opt neighborhood.

Fig. 6.7 shows a concept diagram of EOS. In EOS, adequate normalization of decision variables should be employed in advance, since we use distance among parameter sets.

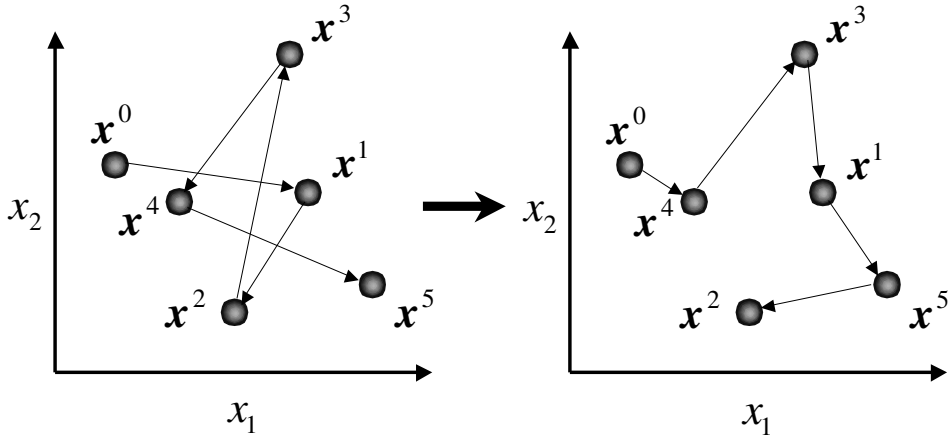


Figure 6.7: Conceptual diagram of the Evaluation Order Scheduling.

6.4 Evaluation Time Scheduling

Along with the convergence of the population, shift of the equilibrium point of a dynamical system gradually diminishes. Consequently, the transient response generated

by switching individuals gradually becomes small as well. Thus, we attempt to change the invalid evaluation time. Assume that EOS is adopted for the initial population $P(0)$ and the current population $P(t)$ as generation t. First, the mean edge length of initial population is calculated as

$$\bar{d}(0) = \frac{d_{\text{total}}(0)}{|P(0)| - 1},$$

where $d_{\text{total}}(0)$ is the path length of $P(0)$, and $|P(0)|$ is the size of $P(0)$. The invalid evaluation time of a individual $\boldsymbol{x}^i$ is adjusted as follows:

$$IET_i = IET_{\max} \cdot \frac{d_{i-1,i}}{\bar{d}(0)},\ i = 1, 2, \ldots, N, \tag{6.3}$$

where $IET_{\max}$ is called the maximum invalid evaluation time, and is a parameter to be set in advance.

Incidentally, the method of automatically judging the output that has reached a stable state might be applied instead of ETS. However, in actual EBEMO, not only the influence of the transient response caused by dynamics but also the uncertainty based on the observation noise is included in the observation value. Therefore, the judgment condition of steady state has to be decided appropriately based on the magnitude of observation noise, because there is a possibility that the steady state will be undetectable indefinitely. Moreover, the judgment condition settings are necessary for the plural outputs in EBEMO. Additionally, it is required that the settings be changed to reflect the targets of the modification specifications and measuring instrument changes. It is desirable to decrease such work as much as possible if operation on an actual calibration scene is considered. Hence, ETS whose configuration parameter is only $IET_{\max}$ is simple and practicable approach.

6.5 Numerical Experiment

6.5.1 Experiment Settings

In this section, the performances of the proposed and conventional methods were compared through simulation of a dynamical system. Let us consider the following system consisting of four independent mass-damper-spring systems that share common adjustable parameters K_1 and K_2 shown in Fig. 6.8:

$$M\ddot{q}_1 + D\dot{q}_1 + K_1 q_1 = -Mg \tag{6.4}$$

$$M\ddot{q}_2 + D\dot{q}_2 + K_2 q_2 = -Mg \tag{6.5}$$

$$M\ddot{q}_3 + D\dot{q}_3 + 2K_1 q_3 = -Mg \tag{6.6}$$

$$M\ddot{q}_4 + D\dot{q}_4 + 2K_2 q_4 = -Mg, \tag{6.7}$$

where q_i is the position of the mass in which the freedom length of the spring is assumed to be zero, M is the mass, D is a damping coefficient, and g is the gravity acceleration, respectively. Since the gravity acceleration g affects the vertical direction of each mass, the transient state is caused by the movement of equilibrium points when K_1, K_2 are switched. The mass-damper-spring system and the time series of Eq. (6.4) of which parameter K_1 was switched at random every five seconds are shown in Fig. 6.9. We can see that the transient response caused by switching K conforms to the example in Fig. 6.6 very well.

Let us consider a problem of putting positions q_1, q_2, q_3, q_4 to the desired position q_d by adjusting K_1, K_2. With this system, objective functions are defined as:

$$f_1 = \sum_{i=1}^{2} (q_\mathrm{d} - \hat{q}_i)^2 \tag{6.8}$$

$$f_2 = \sum_{i=3}^{4} (q_\mathrm{d} - \hat{q}_i)^2, \tag{6.9}$$

where $\hat{q}$ is the estimated value of the steady state position. Because K_1 and K_2 coded as individual of MOEAs are switched by a prescribed interval as shown in Fig. fig6-8, the estimation error for q caused by the transient response is included in f_1 and f_2. The true objective functions excluding the system dynamics are calculated theoretically as follows:

$$f_1^\mathrm{true} = \sum_{i=1}^{2} \left(q_\mathrm{d} - \frac{Mg}{K_i} \right)^2 \tag{6.10}$$

$$f_2^\mathrm{true} = \sum_{i=1}^{2} \left(q_\mathrm{d} - \frac{Mg}{2K_i} \right)^2 . \tag{6.11}$$

In summary, the aim of this problem is multi-objective optimization of true objective functions f_1^true and f_2^true by using their estimated values f_1 and f_2. Because equilibrium points determined by same spring rate are different between Eqs. (6.3)(6.4) and Eqs. (6.5)(6.6), in four mass–damper–spring systems, Eq. (6.3) and Eq. (6.5) that have K_1 and Eq. (6.4) and Eq. (6.6) that have K_2 cannot satisfy q_d simultaneously. Hence, f_1^true and f_2^true cannot equal to 0 simultaneously, and trade-off between f_1 and f_2 exists.

The sampling rate was 100ms on the simulation. For $\hat{q}$, we used the mean of sampled values taken in one second after the invalid evaluation time since an individual was switched. Hence, $\hat{q}$ includes a part of the transient response, when the sampling is executed before diminishing transient response sufficiently.

As for parameter values, $M = 1$, $D = 1$, $K_1, K_2 \in [1, 5]$, $g = 9.81$, $q_\mathrm{d} = -3$ and $w_1 = w_2 = 1$ were used. In this thesis, NSGA-II was employed as a MOEA. Individual was coded as $\boldsymbol{x} = [K_1\ K_2]$. The population size $|P| = 50$, and the offspring population

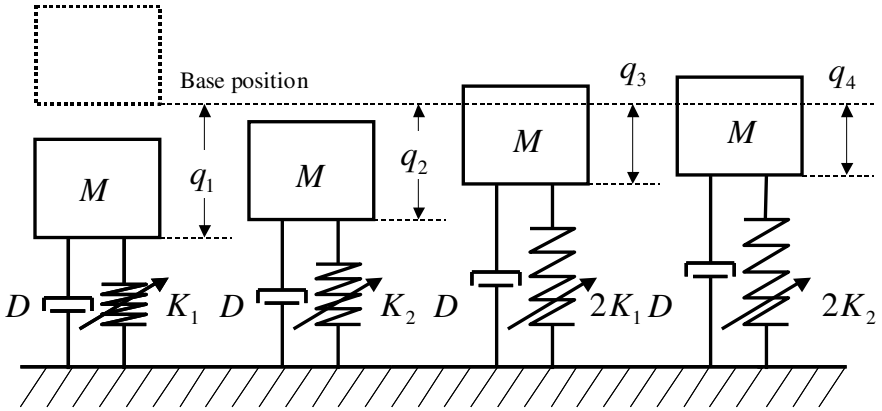


Figure 6.8: Mass–damper–spring system.

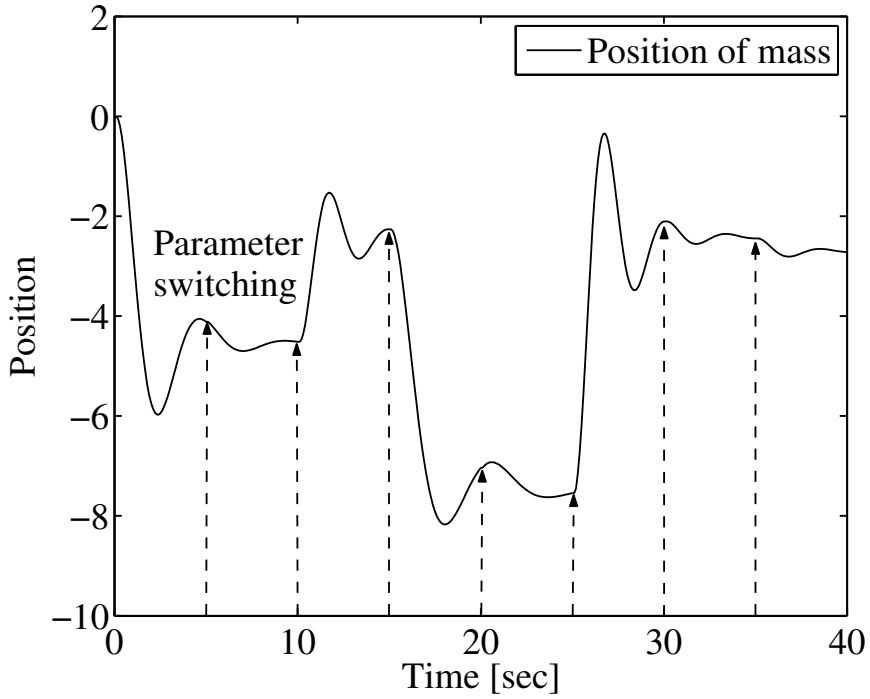


Figure 6.9: Transient response of the mass–damper–spring system.

size $|Q| = 50$ were used. UNDX [80] was used for crossover. Since the evaluation value includes uncertainty due to dynamic behavior of the system, the population was re-evaluated in this numerical experiment, i. e., $R(t) = P(t) \cup Q(t)$ was used as the evaluation population. The search was ended at the evaluation time of 5000 seconds on the simulation.

We compared the following cases:

Case 1: normal NSGA-II, $IET_{\max} = 1$ [sec].

Case 2: normal NSGA-II, $IET_{\max} = 4$ [sec].

Case 3: NSGA-II+EOS, $IET_{\max} = 1$ [sec].

Case 4: NSGA-II+EOS, $IET_{\max} = 4$ [sec].

Case 5: NSGA-II+EOS+ETS, $IET_{\max} = 4$ [sec].

In this experiment, 30 trials using different initial populations were executed, and the performances of the MOEAs were compared by the mean values of the trials for the two evaluation measures introduced in Chapter 4:

Coverage: In this experiment, $m = 2$ and $c_{\max} = 25$ were used.

Mean Absolute Error: In this experiment, the Pareto optimal set is analytically given by

$$X^* = \{x_1, x_2 \in [1.635, 3.27] | x_1 = x_2\}.$$

In addition, to investigate the influence of proposed method on estimated value, mean estimation error between true fitness and estimated value introduced in Chapter 4 is used as follows:

$$E_{\text{error}} = \frac{1}{|P|m} \sum_{i=1}^{|P|} \sum_{j=1}^{m} |f_j^{\text{true}}(\boldsymbol{x}^i) - f_j(\boldsymbol{x}^i)|. \tag{6.12}$$

where $|P| = 50$ ad $m = 2$ are used.

6.5.2 Discussion of Results

Table 6.1 shows the mean value and the standard deviation of coverage and mean absolute error obtained in the final generation of 30 trials. The bold-font and underlined number indicate the best and second-best performances in five cases respectively. Moreover, the mean value and the standard deviation of mean estimation error of 30 trials is shown in Table 6.2. From Table 6.1, Case 5 which is NSGA-II+EOS+ETS with $IET_{\max} = 4$ [sec] achieved the best mean absolute error and second-best coverage. It is understood to show a high level of performance overall in comparison with the others.

Figures 6.10,6.11 and 6.12 show the transition of coverage, mean absolute error and mean estimation error, respectively. Markers in the figures indicate the finish time of each generation. In Case 5, because the number of generations is different depending on the trial, the mean of finish time is indicated. Case 5 was executed an average of 28 generations, Case 2 and 4 were executed 10 generations, and Case 1 and 3 were executed 25 generations in the period of 5000sec. A detailed discussion of the effect of EOS and ETS is described as follows.

Effectiveness of EOS

From the comparison between Case 1 (NSGA-II, $IET_{\max} = 1$ [sec]) and 3 (NSGA-II+EOS, $IET_{\max} = 1$ [sec]), and between Case 2 (NSGA-II, $IET_{\max} = 4$ [sec]) and 4 (NSGA-II+EOS, $IET_{\max} = 4$ [sec]) in Table 6.1, it is understood that EOS improved the coverage and the mean absolute error. It is clearly indicated that the mean estimation error of Case 3 becomes about 10% of that of Case 1. In Case 4 the rate of mean estimation error is about 13% in Table 6.2. On the one hand, because the influence of the transient response caused by system dynamics is controlled by EOS, the estimation error decreased greatly, and the search of NSGA-II did not stagnate. On the other hand, the effect of the improvement of Case 4 was not larger than that of Case 3. This is because the effect of EOS became small since the transient response were settled by the enough invalid evaluation time in Case 2. Hence, the margin of improvement in Case 2 is smaller than that of Case 1, due to the estimation error in Case 2 is already small. As a result, EOS can improve the mean estimation error without depending on the invalid evaluation time. Especially it can be concluded that it is effective for real problems where it is difficult to secure the long invalid evaluation time needed.

Length of Invalid Evaluation Time

Regardless of EOS, the mean estimation error was improved to 20% by increasing IETmax from 1sec to 4sec as indicated in Table 6.2. The long invalid evaluation time can suppress the influence of the transient response and decrease estimation error. For the mean absolute error shown in Fig. 6.10, the convergence velocity in Case 3 was the fastest, because the number of generations could be increased in the same optimization time. The number of generations in Case 3 becomes large because the evaluation time per individual is short if the total optimization time is same. Hence, search is advanced easily in the early stage. Nevertheless, although the estimation error is small, the number of generations in Case 4 is not enough to improve the mean absolute error. In the early stage of the search shown in Fig. 6.11, the coverage of Case 3 was better than Case 4. However, the coverage in Case 4 overcame that in Case 3 at about 2000 seconds and obtained the best result in the end. The domination of Case 4 becomes clear through comparison of each generation. The same coverage as 25th generation in Case 3 was obtained in the fourth generation in Case 4. This result shows the necessity of a well distributed population by accurate ranking and crowding distance calculation in the early stage of the search, to obtain an good coverage.

Effectiveness of ETS

As shown in Fig. 6.10, for ETS, the coverage in Case 5 improved at a rising velocity comparably to that in Case 3, The coverage obtained in the end in Case 5 nearly

equals that in Case 4. Additionally, as shown in Fig. 6.11, the mean absolute error of Case 5 was slightly better than that of Case 4 after 3000 seconds. Figure 6.13 shows the transition of the mean invalid evaluation time of Case 5 for each generation. The invalid evaluation time of Case 5 in the early stage is longer than that of Case 3. Therefore, because the crowding distance of the population can accurately be judged, coverage equal to that in Case 4 is obtained. The invalid evaluation time of Case 5 was adjusted to 0.5sec by ETS in latter stage while that of Case 3 is 1sec. As shown in Fig. 6.12, we can see that the mean estimation error of Case 5 in the latter stage was equal to that of Case 3. This result implies Eq. (6.2) calculates invalid evaluation time adequately, and reduces non-productive time. As a result, the mean absolute error was improved because the number of generations in the same evaluation time can be increased in the latter stage of the search. Therefore, by using ETS the merits of both Cases 3 and 4 can be realized.

Table 6.1: Performance comparison of the test problem for five cases.

Case	Coverage		Mean Absolute Error	
	Mean	Std. Dev.	Mean	Std. Dev.
Case 1	0.66133	0.05251	0.04101	0.01050
Case 2	0.81533	0.02763	0.04765	0.00935
Case 3	0.85200	0.04888	<u>0.02740</u>	0.00746
Case 4	**0.90667**	0.03614	0.04106	0.00818
Case 5	<u>0.88733</u>	0.03503	**0.02501**	0.00615

Table 6.2: Mean estimation error of the test problem for five cases.

Case	Mean	Std. Dev.
Case 1	5.5623×10^{-2}	7.7027×10^{-3}
Case 2	9.7972×10^{-3}	1.6326×10^{-3}
Case 3	5.5454×10^{-3}	1.1936×10^{-3}
Case 4	1.3411×10^{-3}	2.0685×10^{-4}
Case 5	6.0018×10^{-3}	7.0957×10^{-4}

6.6 Real Engine Experiment

6.6.1 Experiment Settings

A four stroke gasoline engine of a motorcycle, which is a different type than that used in Chapter 4, was used in the experiments to validate the effectiveness of proposed IES

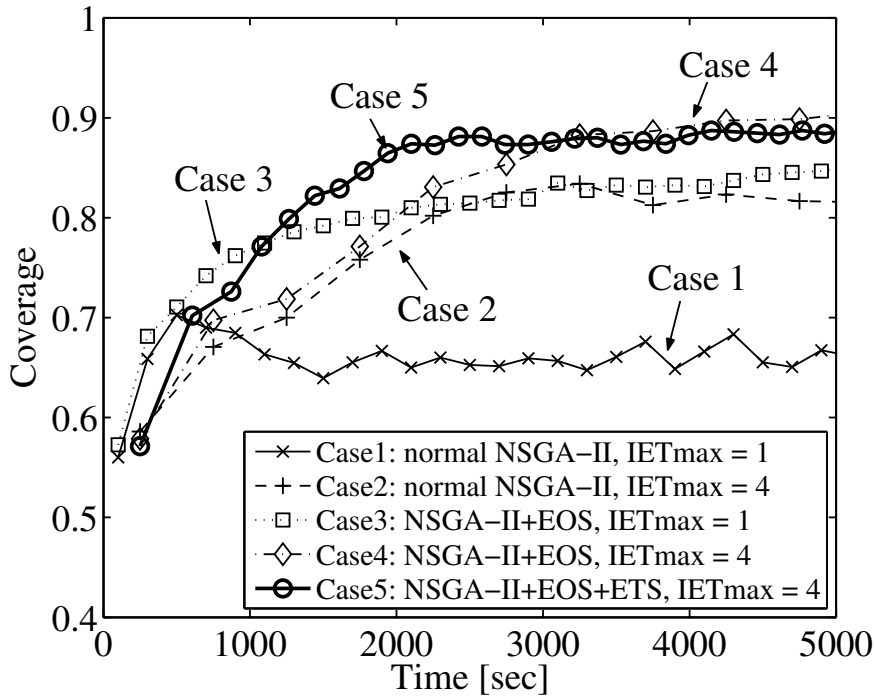


Figure 6.10: Transition of the coverage.

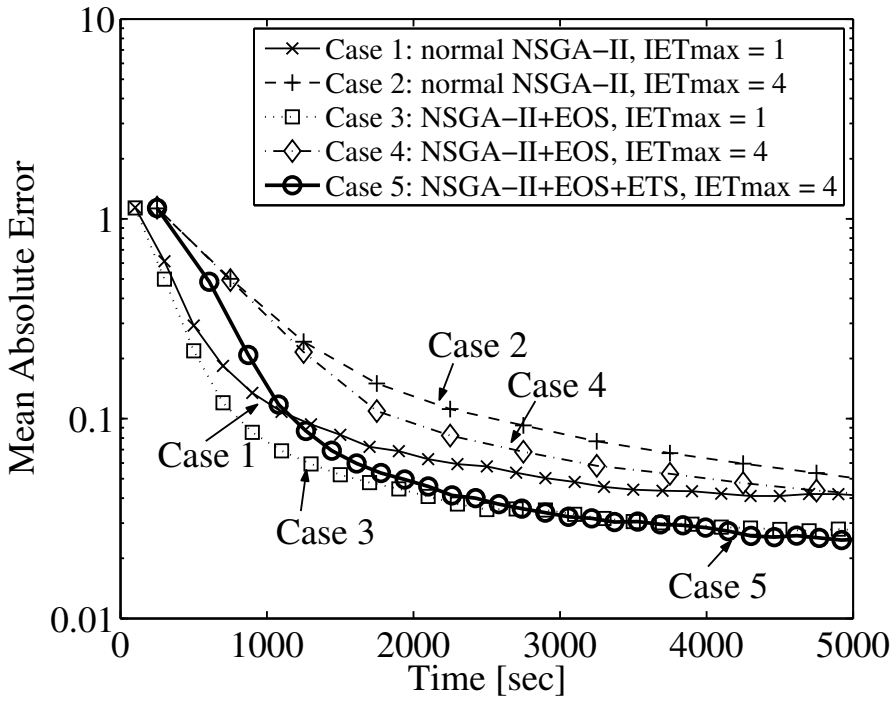


Figure 6.11: Transition of the mean absolute error.

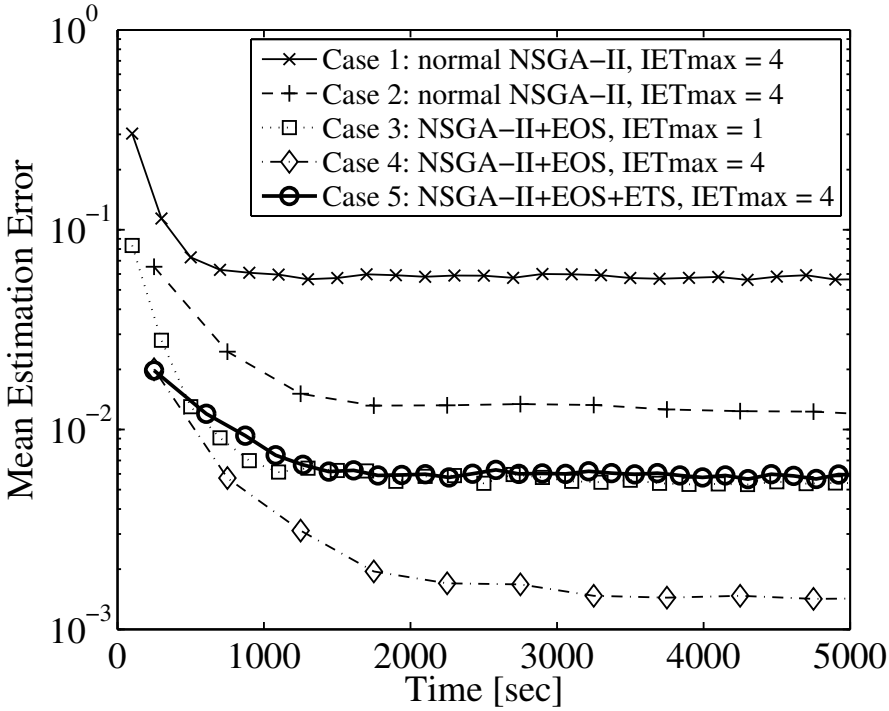


Figure 6.12: Transition of the mean estimation error.

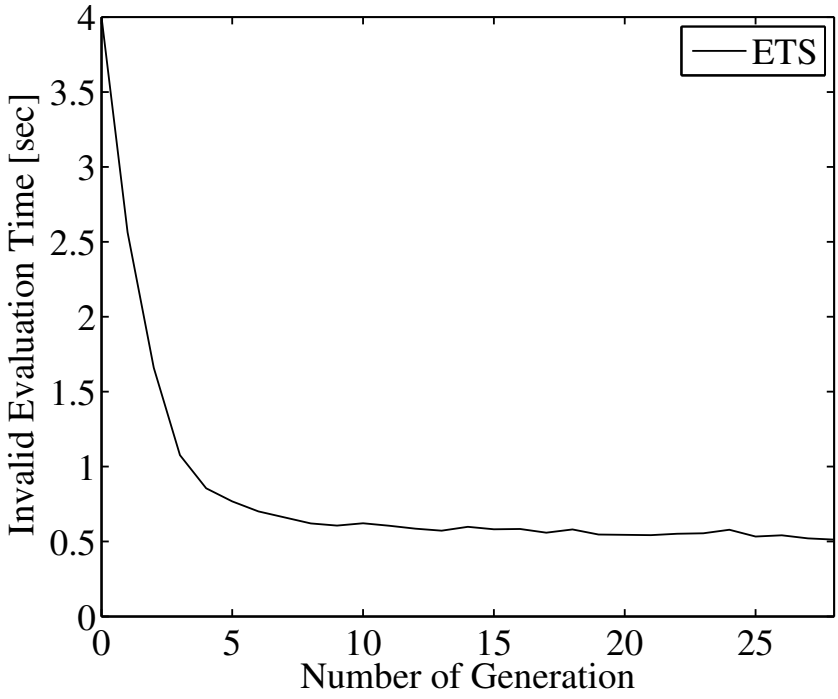


Figure 6.13: Transition of the invalid evaluation time.

algorithm. Normal NSGA-II and NSGA-II+IES, that is, NSGA-II+EOS+ETS were applied to a two-objective four-variable function optimization problem of the engine. The multi-objective optimization problem was set as follows:

Objective Functions: engine torque (maximization), fuel-consumption (minimization).

Decision Variables (Control Parameters): fuel-injection timing (INJ), ignition timing (IGN), desired air-fuel ratio (AFR), valve control parameter (VCP).

AFR means the desired air-fuel ratio, that is ratio of air and gasoline mass, of the feedback controller. VCP means desired value of the device that controls valve state. In general, there is a trade-off between the engine torque and the fuel-consumption. Additionally, changing the aforementioned decision variables, i.e., the engine control parameters, complicates the trade-offs. The engine torque and fuel-consumption were measured at a constant engine speed.

Both NSGA-II and NSGA-II+IES were programmed into a calibration PC. In the calibration PC, objective functions and decision variables were processed as normalized values. The calibration PC was connected to an ECU by a serial communication.

The experiment condition of NSGA-II is as follows: Population size $|P| = 30$ and offspring population size $|Q| = 30$ were used. The number of evaluations was 1530 (Population P was re-evaluated as we did in the numerical experiments). The mean value of the sampled data was used as estimated value.

Both NSGA-II and NSGA-II+IES employed UNDX-P as a crossover operator. The crossover rate was 1.0. Since UNDX-P shows good performance without mutation, we did not use a mutation operator.

After optimizations, the true fitness of the individuals were verified through a longer measurement period. These were used for comparison of the approximation Pareto frontier.

6.6.2 Discussion of Results

At first, we investigate the time series of the engine torque shown in Fig. 6.14. We can see a large amount of vibration throughout the time series of normal NSGA-II, where that of NSGA-II+IES changes more smoothly. These results indicate EOS determine adequate evaluation order of the population convergence in the vicinity of the Pareto optimal set. In addition, the suppression of transient response by EOS can improve the accuracy of the estimated values. Moreover, a secondary effect of the smooth output torque response is that it does not input rapid torque changes to engines. From the viewpoint of apparatus protections, it is a quite desirable feature in experiment-based optimizations.

Fig. 6.15 shows the population distribution of the estimated value and the true fitness in the objective function space. Each axis is shown in normalized scales. From

this figure, we can see that the Pareto approximation set shows a straight line shape, and the accuracy of estimated value of the NSGA-II+IES is higher than that of the normal NSGA-II. Table 6.3 denotes the mean estimation error calculated by Eq. (6.12). We can confirm that the error of NSGA-II+IES is about 30% less than that of normal NSGA-II. Hence, it is understood that the accuracy of the estimated value is improved by EOS.

Table 6.3: Mean estimation error of NSGA-II and NSGA-II+IES.

MOEA	Mean	Std. Dev.
NSGA-II	0.01099	0.00905
NSGA-II+IES	0.00779	0.00870

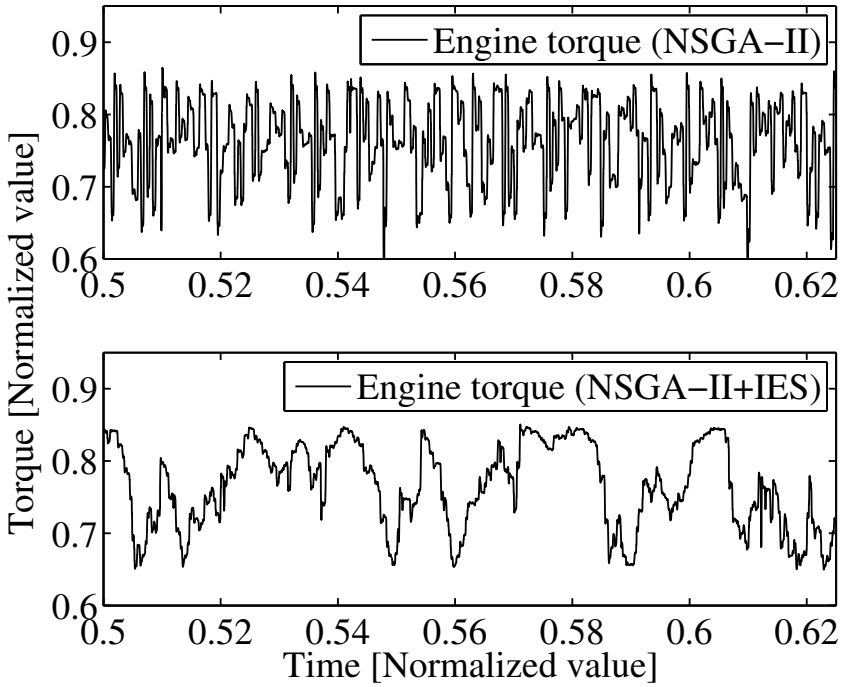


Figure 6.14: Time series of engine torque (upper: NSGA-II, lower: NSGA-II+IES).

We also show the comparison of the Pareto frontier approximations by true fitness in Fig. 6.16. This figure indicates that the NSGA-II+IES found a better convergence of the Pareto optimal set than the normal NSGA-II, especially in high torque areas around 0.8. This result indicates that the search process did not stagnate because NSGA-II+IES improved the accuracy of estimated value.

In addition, the decision variable space of the final population is shown in Fig. 6.17. Because the population of NSGA-II+IES is more converged than that of NSGA-II, we

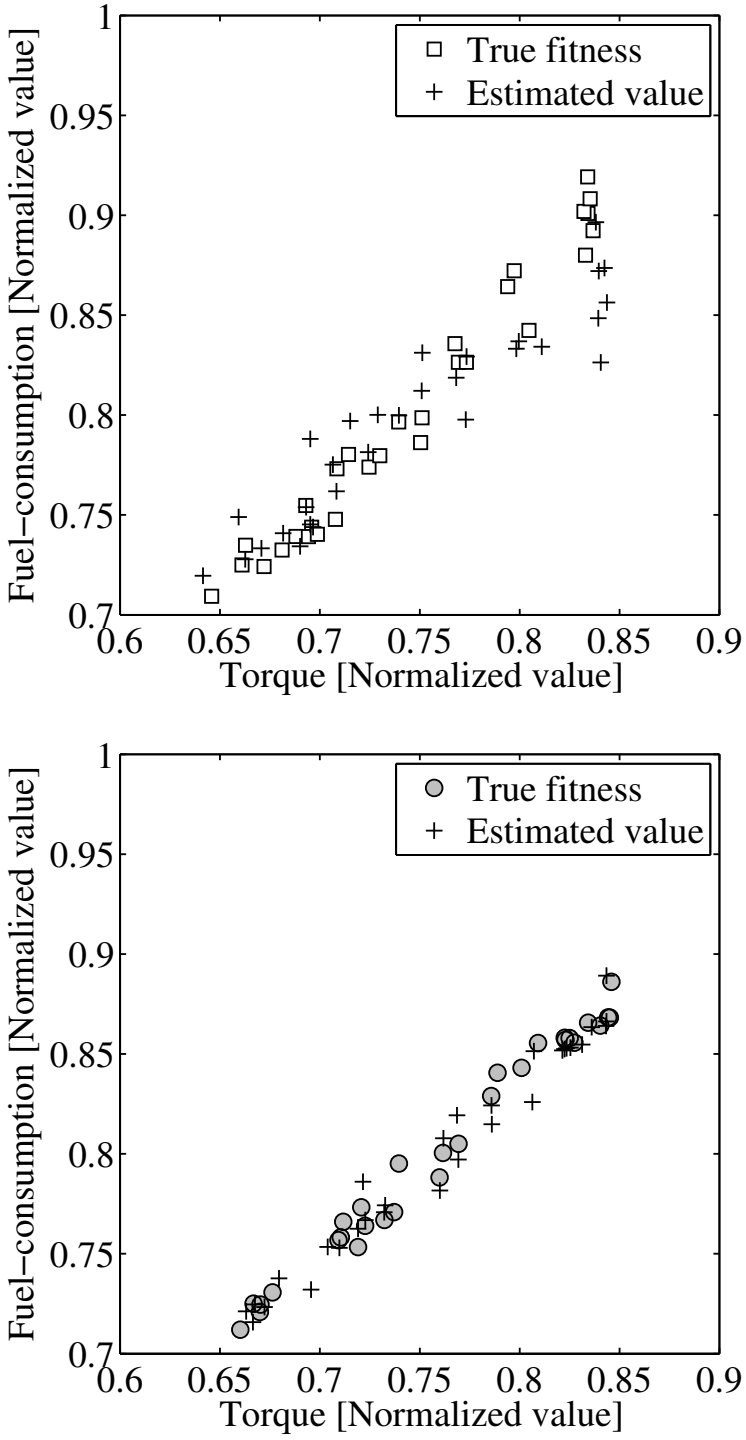


Figure 6.15: Population distribution in the objective function space (upper: NSGA-II, lower: NSGA-II+IES).

can confirm Pareto approximation solutions in decision variable space.

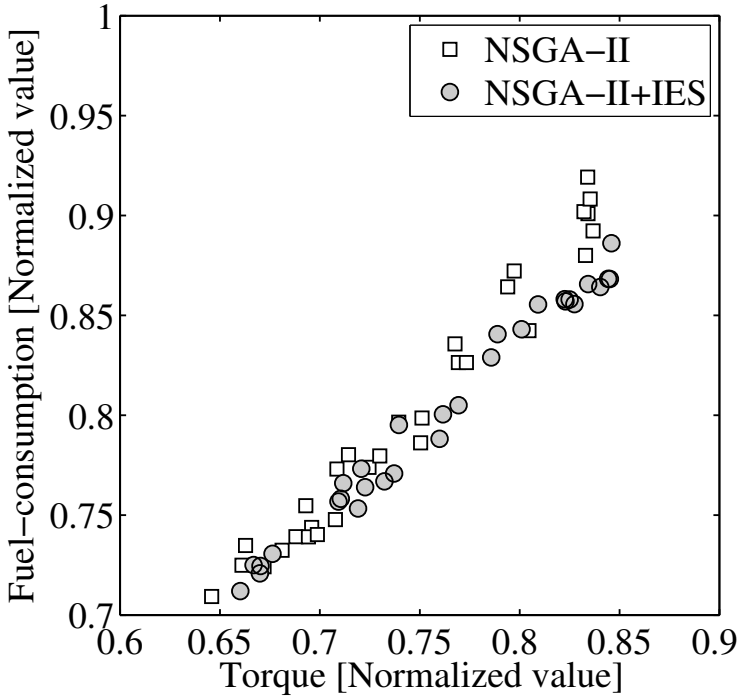


Figure 6.16: Comparison of the approximation Pareto frontier by true fitness.

Moreover, though the search accuracy was improved, the search time of NSGA-II+IES was shortened about 13% to that of the normal NSGA-II by ETS in the same number of evaluations. Figure 6.18 shows the transition of the mean invalid evaluation time for generations of NSGA-II+IES. The solid line indicates the mean IET of NSGA-II, and the broken line indicates the fixed IET of NSGA-II. The horizontal axis means the number of generation, and the vertical axis means normalized invalid evaluation time. In this case, IET is normalized by the fixed IET of NSGA-II. Therefore, IETmax of NSGA-II+IES is 2. We can confirm that NSGA-II+IES keeps IET about the 1st until the 10th generation and gradually decrease it to 0.4 after that. Hence, it is understood that ETS has reduced the invalid evaluation time as the distance between individuals becomes smaller by converging to the neighborhood of Pareto best solution.

To confirm the validity of invalid evaluation time, standard deviation of sampling data that is used to calculate the fitness of output torque and fuel consumption is investigated. Each piece of sampling data is normalized. Figure 6.19 shows the mean value of the standard deviation of each generation. Both of the minimum standard deviations were from the initial generation. This result implies that the IETmax was kind of large. The standard deviation of the output torque was stable after the 10th generation, although it gradually became large along with shortening of the invalid evaluation time. As for the standard deviation of the fuel consumption, it is understood

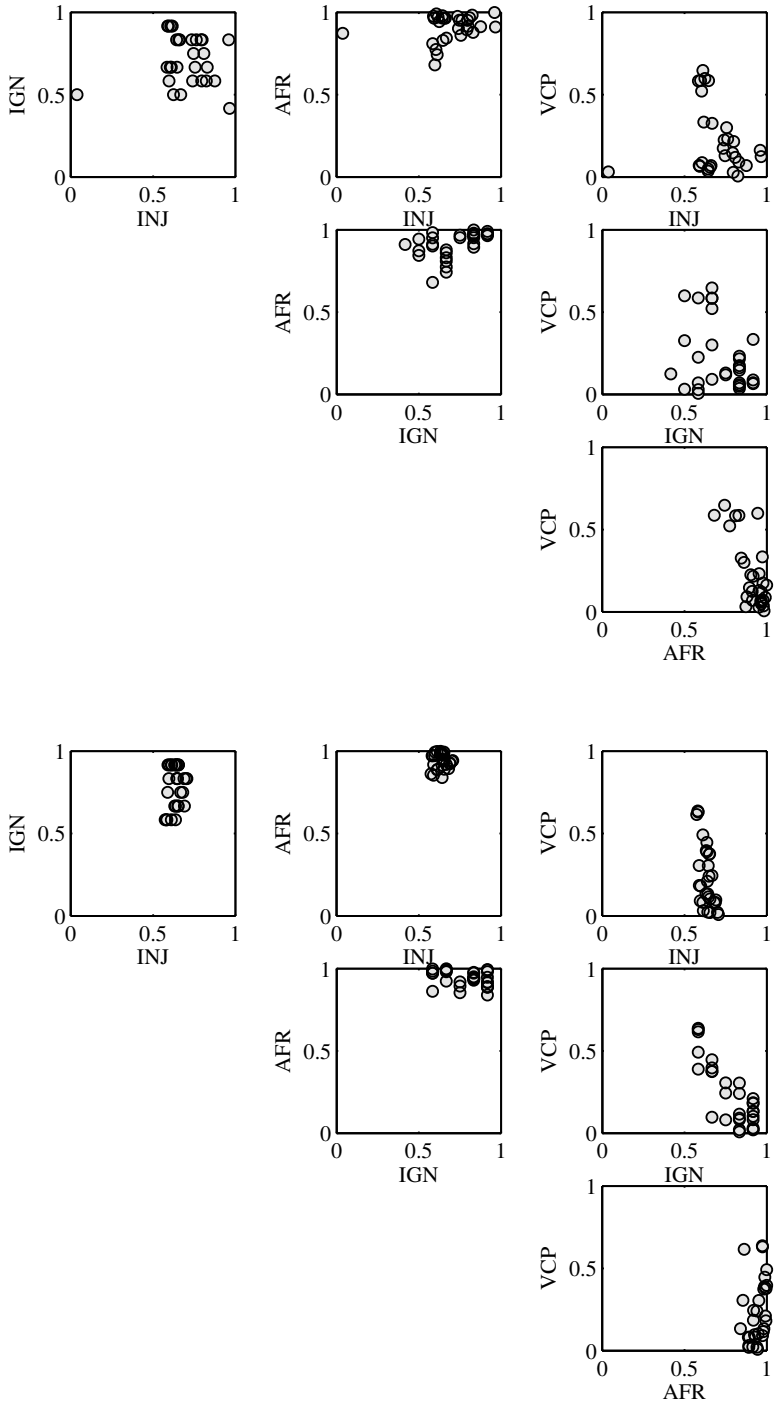


Figure 6.17: Population distribution in the decision variable space (upper: NSGA-II, lower: NSGA-II+IES).

that it gradually became small after it grew until the 10th generation. Hence, it can be understood that the influence of the transient response included in the estimated values was roughly constant through the generation. As a result, it can be concluded that ETS will appropriately calculate the invalid evaluation time.

Through discussion with experts of engine calibration, we confirmed that the characteristic of the Pareto approximation set was appropriate enough to represent the performance of the engine, although the optimization time was not reached as low as that of the operation time of the experts. Therefore, it can be concluded that IES is an effective technique for the experiment-based EMO of real engines.

6.7 Summary

In this chapter, Individual Evaluation Scheduling for the experiment-based evolutionary multi-objective optimization was proposed. Through numerical experiment using a formal test problem and experiment using a HILS environment for real engines, the effectiveness of EOS and ETS can be summarized as follow:

- EOS can suppress the transient response caused by system dynamics by suitable evaluation order of population. Hence, the estimation error for true fitness can be reduced. As a result, EOS contributes to the improvement of search accuracy because optimization process does not stagnate.

- ETS can adequately adjust the invalid evaluation time based on the distance between individuals. Hence the non-productive invalid evaluation time can be excluded. Therefore, ETS contributes to the improvement of search accuracy under same total evaluation time, and reduction of search time under same number of evaluation.

In conclusion, it was understood that the Pareto optimal solutions having the high coverage and the small mean absolute error were obtained by EOS and ETS. Because the proposed method is independent from specific MOEA, IES is applicable to various MOEAs.

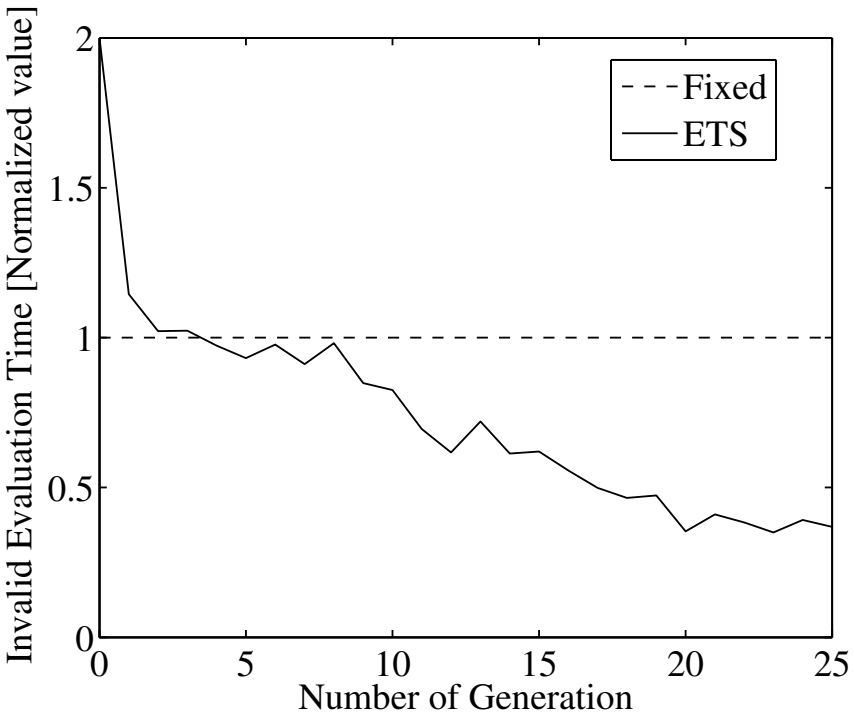


Figure 6.18: Transitions of invalid evaluation time.

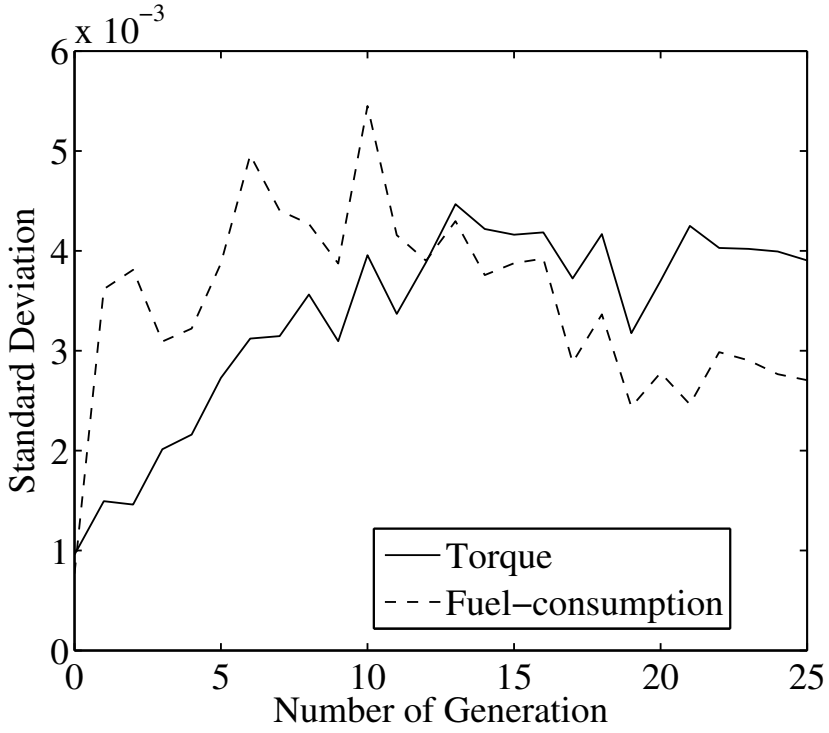


Figure 6.19: Transitions of standard deviations of estimated values.

Chapter 7

Acceleration Method using Fitness Estimation

7.1 Outline

In Chapters 4 to 6, methods to solve the first problem, that is, the search performance of MOEAs to the influence of observation noise, periodicity and system dynamics were developed. Although the improvement of search performance through real engine experiments were confirmed, a search time comparable to that of engine calibration done by experts has yet to be achieved. To make EBEMO using the HILS environment feasible, the most important pre-requisite is reduction of the number of necessary fitness evaluations. Parallelization is one of the solutions for real world problems that require enormous evaluation costs [22, 108]. However, the parallelization of engine HILS environments is not a realistic choice in view of the installation cost and space.

As another approach, acceleration methods for EAs mainly studied have been used to build approximation models from information of evaluated individuals in a real environment and then using them as a low-cost surrogate. Figure 7.1 presents a typical structure of acceleration method for EAs. EAs including MOEAs can adequately switch both real environments and approximation models. Individuals evaluated in real environments and these fitness are stored in the search history. The approximation models are constructed by the search history.

Generally speaking, it is possible to decrease the total evaluation time substantially because the evaluation cost of the approximation model is smaller than the real environment. These acceleration methods have been applied to simulations, which demand enormous evaluation costs such as Computational Fluid Dynamics (CFD) [11, 17, 52].

Therefore, we set out to solve this problem through the application of an acceleration method by evaluation reductions of EAs. In the EBEMO, the performance of the evaluation reduction under uncertainty such as observation noise is highly important, although the previous works often assume noise-free environments [17, 28, 29, 52, 62, 69, 70, 75].

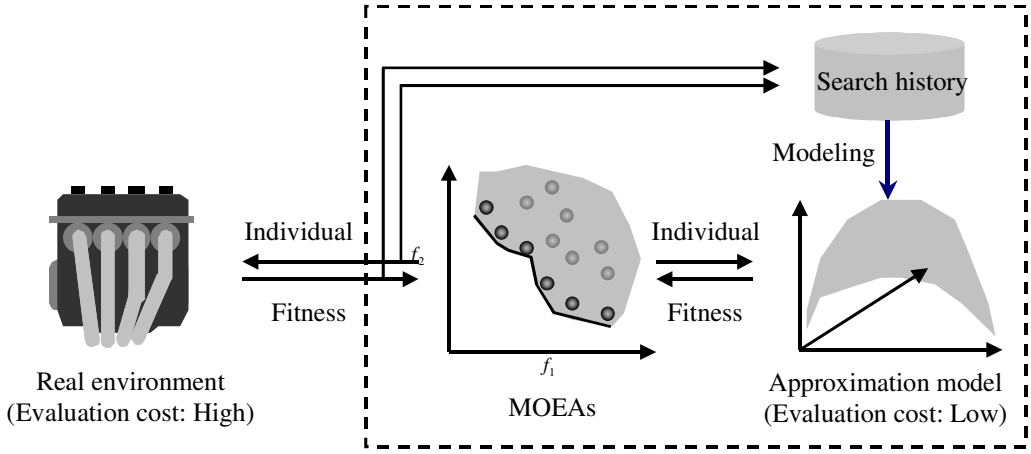


Figure 7.1: Schematic representation of the acceleration method for MOEAs.

In this chapter, an evaluation reduction is proposed to overcome the aforementioned problem by selecting the solution candidates by means of the estimated fitness (estimated value) before applying them to the real experiment in MOEAs. This technique is called 'Pre-selection'. For the estimation of fitness, we adopt Locally Weighted Regression (LWR) [2]. The effectiveness of the proposed method is examined by some numerical experiments and also a two-objective four-variable optimization problem of a real internal-combustion engine using HILS.

7.2 Acceleration Method for MOEAs

7.2.1 Related Works

Since EAs are stochastic optimization method, offspring may be generated at a position far from an area containing the optimal solution. It is undesirable to evaluate such non-promising offspring in the real environment of costly evaluations. If the offspring can be evaluated by the approximation model of the fitness, non-promising offspring can be excluded beforehand to make optimization efficient. Such a technique called Pre-selection or Pre-screening winnows promising offspring based on the estimated value obtained by the approximation model [27, 59]. The Pre-selection has the feature which does not lose the advantage of a direct search and updates the approximation model every generation.

Figure 7.2 depicts a flow diagram of a Pre-selection algorithm. First, candidate offspring are generated by using crossover and mutation operators and are evaluated on the approximation model. Second, promising candidate offspring having good fitness are evaluated in the real environment. Finally, generation alternation is executed through comparison of the fitness value.

In conventional evaluation reduction for EAs, many researchers have employed

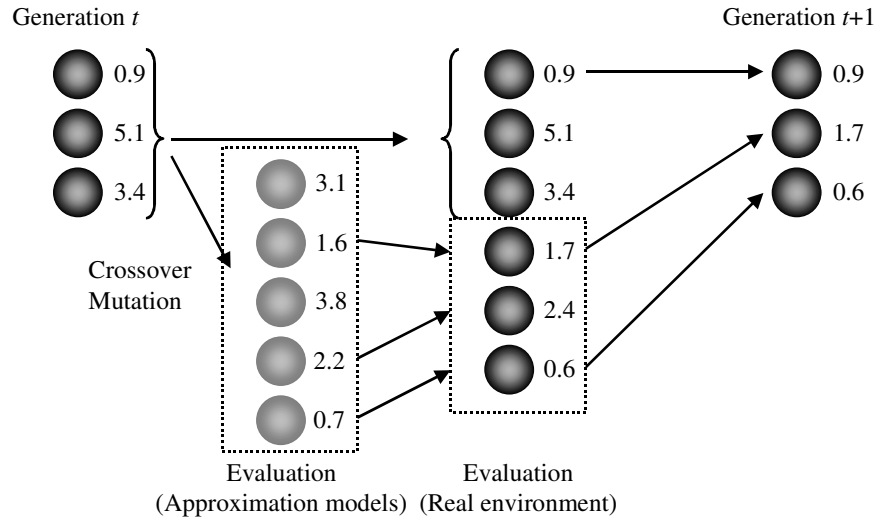


Figure 7.2: Flow diagram of the Pre-selection algorithm.

many kinds of approximation modeling techniques for objective functions. For example, polynomials [72], Artificial Neural Networks [53, 54, 55, 56, 59, 38], Radial Basis Functions [76, 79], Support Vector Regression [107], Gaussian Process [11] and Kriging (Design and Analysis of Computer Experiments, DACE [90]) [27, 110, 84, 113] are adopted.

For instance, Emmerich et al. have proposed a Pre-selection for MOEAs which used Kriging [28, 29]. They paid attention to a feature that Kriging can predict the mean square error of the estimated values. To select individuals that have a high possibility for improvement, they employed merit functions which consider the mean square error of Kriging models for the ranking operation.

However, because Kriging generates approximation models that pass through all the sample points, it is not suitable for the EBEMO in which evaluation values involve uncertainties such as observation noise. Similarly, previous works have assumed noise-free environments [17, 52, 62, 69, 70, 75].

Among the previous works, we paid special attention to the researches by Branke et al. [8, 9]. They have proposed a Pre-selection that used Locally Weighted Regression (LWR) [2] to estimate values of candidates as an evaluation reduction for single objective EAs, and have shown its effectiveness by detailed examinations. They also have demonstrated that the fitness estimation by LWR was effective for the problems that have uncertainty resulting from observation noise. Therefore, because our goal is to develop an evaluation reduction for experiment-based EMO including uncertainties such as observation noise, we employ LWR as an approximation modeling technique. Moreover, we propose a Pre-selection algorithm which does not depend on features of approximation modeling techniques.

7.2.2 Locally Weighted Regression

In this chapter, m-objective n-dimensional optimization problems in continuous space are considered. Locally weighted regression is a technique for fitting a local regression to nearby data. This approach is also referred to as memory-based approach. Consider a set of search history

$$H = \{(\boldsymbol{h}^1, \boldsymbol{f}(\boldsymbol{h}^1)), (\boldsymbol{h}^2, \boldsymbol{f}(\boldsymbol{h}^2)), \ldots, (\boldsymbol{h}^l, \boldsymbol{f}(\boldsymbol{h}^l))\},$$

which stores information of the search process of MOEAs, where $\boldsymbol{h} = [h_1 \ \cdots \ h_n]^{\mathrm{T}}$ is n-dimensional decision variable vector which was evaluated as an individual in a real environment. $\boldsymbol{f}(\boldsymbol{h}) = [f_1(\boldsymbol{h}) \ \cdots \ f_m(\boldsymbol{h})]^{\mathrm{T}}$ is m-dimensional objective function value vector, l is the number of individuals stored in H.

A low polynomial such as constant, linear, or second-order is usually used for local regression. A neighborhood set Ω of an individual (query) $\boldsymbol{x} = [x_1 \ \cdots \ x_n]^{\mathrm{T}}$is generated from H by k-Nearest Neighbors (k-NN) method based on the Euclidean distance

$$d_{\mathrm{E}}(\boldsymbol{h}, \boldsymbol{x}) = \sqrt{(\boldsymbol{h} - \boldsymbol{x})^{\mathrm{T}}(\boldsymbol{h} - \boldsymbol{x})}.$$

Then, weighted regression is executed for Ω locally. Hence, $l \geq k$ must be satisfied, where k is the size of Ω. If the dimension of decision variable vector is n, the minimum number of k required for local regression model is as follows:

- Linear polynomial: $n + 1$
- Second-order polynomial without interaction terms: $2n + 1$
- Second-order polynomial: $(n + 1)(n + 2)/2$

When we have to consider the influence of observation noise, we should set k depended on the noise level.

When a second-order polynomial without interaction terms is employed as a local regression model, the estimated value $\hat{\boldsymbol{f}}(\boldsymbol{x}) = \left[\hat{f}_1(\boldsymbol{x}) \ \cdots \hat{f}_m(\boldsymbol{x})\right]^{\mathrm{T}}$ is calculated by the following equations:

$$\hat{f}_i(\boldsymbol{x}) = \begin{bmatrix} 1 & x_1 & \cdots & x_n & x_1^2 & \cdots & x_n^2 \end{bmatrix} \boldsymbol{b}_i \tag{7.1}$$

$$\boldsymbol{b}_i = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{y}_i \tag{7.2}$$

$$\boldsymbol{X} = \begin{bmatrix} 1 & h_{11} & \cdots & h_{1n} & h_{11}^2 & \cdots & h_{1n}^2 \\ & & & \vdots & & & \\ 1 & h_{k1} & \cdots & h_{kn} & h_{k1}^2 & \cdots & h_{kn}^2 \end{bmatrix} \tag{7.3}$$

$$\boldsymbol{y}_i = \left[f_i(\boldsymbol{h}^1) \ f_i(\boldsymbol{h}^2) \ \cdots \ f_i(\boldsymbol{h}^k)\right]^{\mathrm{T}}, \tag{7.4}$$

where h_{ij} is jth element of $\boldsymbol{h}^i$ which is an individual near to ith from $\boldsymbol{x}$ in Ω, and $\boldsymbol{W}$ is called weighted matrix which is a diagonal matrix with diagonal elements

$$w_i = \sqrt{K(d_{\mathrm{E}}(\boldsymbol{h}^i, \boldsymbol{x}))}, \ i = 1, 2, \ldots, k. \tag{7.5}$$

$K(\cdot)$ is called weighted function or kernel function and is used to calculate the weight of search histories. In this thesis, Gaussian kernel

$$K(d) = \exp\left(-\frac{d^2}{u}\right) \tag{7.6}$$

is used, where u is a smoothing parameter and is set to be the distance to the kth nearest search history in this chapter.

In MOEAs, we can assume the distribution of the population gradually converges into the Pareto optimal set through the search. Hence, a lot of individuals near the Pareto optimal set will be preserved in the search history set H. If convergence takes place, a strong correlation between decision variables might appear when the effective dimension of the Pareto optimal set degenerates in the decision variable space. Therefore, multicollinearity should be considered. That is, estimation accuracy of regression coefficient vector $\boldsymbol{b}$ deteriorates because the matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X}$ gets close to singular and numerical calculation of the inverse matrix $(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X})^{-1}$ becomes unstable.

To avoid the influence of multicollinearity, we employ ridge regression [2, 46]. Instead of Eq. (7.2), ridge regression uses the following equation:

$$\boldsymbol{b}_i = \left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X} + \lambda\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{y}_i, \tag{7.7}$$

where λ is constant value, called the ridge parameter, used to avoid singularity of matrix $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{X}$ and to stabilize inverse matrix calculation.

7.2.3 Pre-selection for EBEMO

When Pre-selection is applied for single-objective EAs, the estimated value $\hat{f}(\boldsymbol{x})$ is used for promising offspring selection. Hence, it is natural extension that the estimated value vector $\hat{\boldsymbol{f}}(\boldsymbol{x})$ is used for the selection in ranking based MOEAs such as NSGA-II. In this case, an important point of Pre-selection for MOEAs is how to select the offspring that should be evaluated in the real environment when lots of promising candidate offspring of equal rank exist. Because we employ LWR for approximation modeling technique, instead of the mean square error prediction of the Kriging [28, 29], we use a sparsity criterion of each promising candidate offspring in the archived population for a useful and effective search. This algorithm is constructed in the manner that the offspring generated in an area having sparse distribution of non-dominated individuals in the archived population is preferentially selected. In this chapter, crowding distance proposed by Deb et al. [19] is used as a sparsity criterion.

Figure 7.3 shows the conceptual diagram of the proposed method. The prototype algorithm is shown below:

1. Evaluate all of fitness vector $\boldsymbol{f}(\boldsymbol{x})$ in the initial population $P(0)$, $t = 0$ in a real environment, and preserve individuals and their fitness vectors in search history set H.

2. Generate a candidate offspring population $Q_{\mathrm{C}}(t)$ from the archived population $P(t)$ by applying the selection, crossover, and mutation operators.

3. Calculate estimated value vectors $\hat{\boldsymbol{f}}(\boldsymbol{x})$ of $R_{\mathrm{C}}(t) = P(t) \cup Q_{\mathrm{C}}(t)$ by using the approximation model constructed by use of H.

4. Assign the rank of $R_{\mathrm{C}}(t)$ by using the α-domination strategy proposed by Ikeda et al. [49] based on $\hat{\boldsymbol{f}}(\boldsymbol{x})$.

5. Add a candidate offspring which becomes a non-dominated individual to $P(t)$, and calculate its crowding distance (This operation is applied for all the non-dominated candidate offspring.).

6. Select the evaluated offspring population $Q(t)$ from the q non-dominated candidate offspring assigned with good crowding distance.

7. Evaluate offspring of Q in the real environment, and preserve the offspring together with the fitness vector in H.

8. Assign the rank and the crowding distance of $R(t) = P(t) \cup Q(t)$ based on $\boldsymbol{f}(\boldsymbol{x})$ (noise-free environment) or $\hat{\boldsymbol{f}}(\boldsymbol{x})$ (noise environment) and select high-ranked $|P|$ individuals of R as next generation.

9. If the termination condition is satisfied, finish this procedure; otherwise, set $t := t + 1$ and return to Step 2).

If the number of individuals stored in the search history set H is insufficient for the construction of the approximation model, individuals are evaluated in the real environment until it reaches the necessary number. The minimum number of l that needs model construction depends on the problems and the modeling techniques. In the proposed method, LWR is employed as a modeling technique. Hence, individuals are only evaluated in the real environment when $l < k$. If we set $k \leq |P|$, Pre-selection can start after initial population evaluation.

Moreover, we employ the crowding distance proposed by Deb et al. [19] as a sparsity measure. In Step 4), α-domination strategy is used to exclude individuals treated as non-dominated individuals because of the estimation errors introduced in Chapter 4. Since the proposed method enables the preferential evaluation of offspring generated in a sparse area of distribution of non-dominated individuals in real environments, improvement of the search performance and the accuracy of approximation models can be expected.

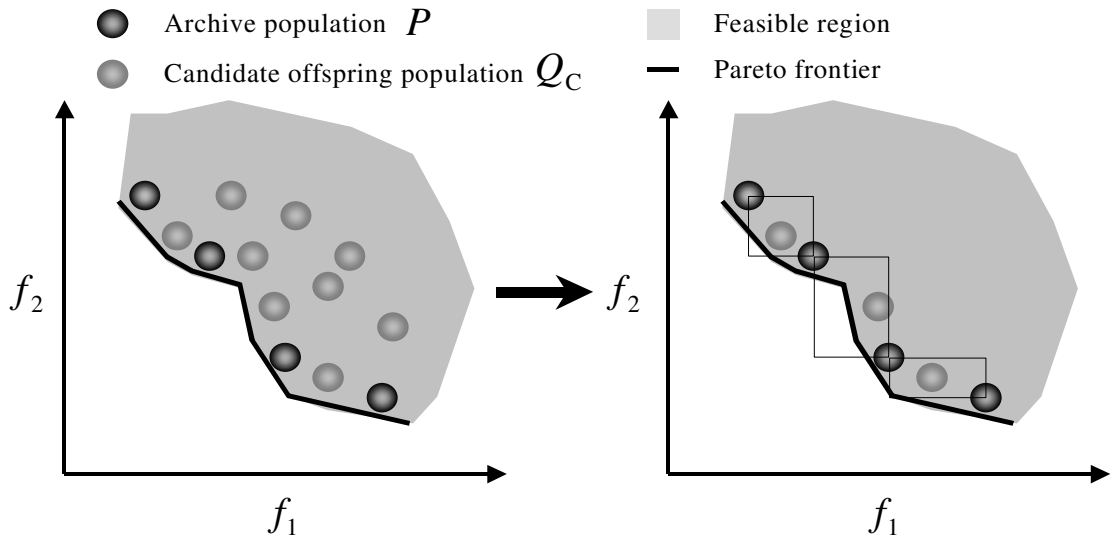


Figure 7.3: Conceptual diagram of the Pre-selection algorithms for EBEMO. The figure on the left shows the set $R_C = P \cup Q_C$, and the figure on the right depicts crowding distance calculation of candidate offspring which became non-dominated individuals.

7.3 Numerical Experiments

7.3.1 Experiment Settings

From the test problem set described by Deb [18], we employ two objective optimization problems SCH, FON, ZDT1, and three objective optimization problem DTLZ2 for numerical experiment.

- SCH

$$f_1(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i^2$$
$$f_2(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} (x_1 - 2)^2$$
$$x_i \in [-4, 4], \quad i = 1, 2, \ldots, n.$$

  This problem has a convex Pareto frontier.

- FON

$$f_1(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i-1}^{n}\left(x_i - \frac{1}{\sqrt{n}}\right)^2\right)$$
$$f_2(\boldsymbol{x}) = 1 - \exp\left(-\sum_{i-1}^{n}\left(x_i + \frac{1}{\sqrt{n}}\right)^2\right)$$
$$x_i \in [-2, 2], \quad i = 1, 2, \ldots, n.$$

This problem has a concave Pareto frontier.

ZDT1 and DTLZ2 were introduced in Chapter 4. The number of decision variables of all problems was $n = 10$.

In this chapter, NSGA-II [19] was employed as an MOEA. The archived population size $|P| = 30$, the candidate offspring population size $|Q_C| = 100$ and evaluated offspring population size $|Q| = 4$ were used for the Pre-selection NSGA-II. For crossover, UNDX proposed by Ono et al. was used [80]. Since it is known that the UNDX shows good performance without mutation, we did not use mutation operation.

A second polynomial without interaction terms was employed as a local model. Hence, the size of Ω have to satisfy $k \geq 2n + 1$. Because the dimension of decision variable space of test functions are $n = 10$, $k \geq 21$ should be used. In this experiment, the neighborhood set size for k-NN was calculated as follows:

$$k = \begin{cases} 30 & |H| \leq 600 \\ \lfloor 0.05|H| \rfloor & |H| > 600 \end{cases}, \tag{7.8}$$

where $|H|$ is the size of search history, $\lfloor a \rfloor$ is the maximum integer of a and under. The parameter of α-domination strategy employed $\alpha = 0.05$, and the ridge parameter was $\lambda = 0.0001$. Normal NSGA-II ($|P| = 30$,$|Q| = 30$) was used for comparison. Because NSGA-II does not use estimated values as like Pre-selection, both of NSGA-II re-evaluation P and no-evaluation P were employed. In re-evaluation, $F_i(\boldsymbol{x})$ is calculated by using new δ_i. The number of evaluations was 2030 (Pre-selection), 2040 times (NSGA-II, no-evaluation P), and 2070 times (NSGA-II, re-evaluation P), respectively. Because the number of function evaluations is the most restrictive factor for experiment-based optimization, comparison was carried out with the same number of function evaluation, although computation time is another candidate condition. Thirty trials with different initial populations were conducted.

Performances of two methods were compared by the mean values of the 30 trials for the two evaluation measures described below:

Coverage: In this experiment, $c_{\max} = 15$ ($m = 2$) and $c_{\max} = 6$ ($m = 3$) were used.

Mean Absolute Error: In this experiment, the Pareto optimal solution sets are given by

$$\begin{aligned} X^*_{\mathrm{SCH}} &= \{x_1, \ldots, x_n \in [0, 2] \mid x_1 = \ldots = x_n\} \\ X^*_{\mathrm{FON}} &= \{x_1, \ldots, x_n \in [-1/\sqrt{n}, 1\sqrt{n}] \mid x_1 = \ldots = x_n\} \\ X^*_{\mathrm{ZDT1}} &= \{x_1 \in [0, 1],\ x_2 = \ldots = x_n = 0\} \\ X^*_{\mathrm{DTLZ2}} &= \{x_1, x_2 \in [0, 1],\ x_3 = \ldots = x_n = 0.5\}. \end{aligned}$$

7.3.2 Performance Analysis under Noise-free Environments

First, we compare the results of the proposed and conventional method in a noise-free environment. The results are shown in Table 7.1. The best performance in each problem is indicated in the bold font. From Table 7.1, it is understood that the proposed method outperforms NSGA-II. As an example, transitions of coverage and mean absolute error for FON are shown in Figs. 7.4 and 7.5. The results indicate that the proposed method can reduce the number of evaluations from the 2000 needed using the conventional method to 400 evaluations. This tendency was similar on the other three test problems, too.

Table 7.1: Performance comparison of the NSGA-II with Pre-selection and the normal NSGA-II for four test functions in noise-free environment.

Test Functions	Method	Coverage		M. A. E.	
		Mean	S. D.	Mean	S. D.
SCH	Pre-selection	**0.98111**	0.01894	**0.11279**	0.01850
	NSGA-II	0.91444	0.03467	0.39274	0.07934
FON	Pre-selection	**0.98889**	0.01822	**0.03448**	0.00538
	NSGA-II	0.96556	0.03093	0.14048	0.02386
ZDT1	Pre-selection	**0.98333**	0.02099	**0.00577**	0.00412
	NSGA-II	0.42000	0.18967	0.07670	0.02241
DTLZ2	Pre-selection	**0.37068**	0.05258	**0.10044**	0.01692
	NSGA-II	0.24877	0.04334	0.15341	0.02265

Next, the influences of the Pre-selection parameters were examined on the following condition. In this experiment, neither k nor α were analyzed because they are the setting parameters for estimation.

- Candidate offspring population size $|Q_{\mathrm{C}}| = 30,\ 100,\ 300$
- Evaluated offspring population size $|Q| = 1,\ 4,\ 10$
- Parameters of the UNDX [1]
  $(\sigma_\xi, \sigma_\eta) = (0.25, 0.175/\sqrt{n}), (0.5, 0.35/\sqrt{n}), (1.0, 0.7/\sqrt{n})$

These parameters were compared with the basic setting of the algorithm $|Q_{\mathrm{C}}| = 100$, $|Q| = 4$, $(\sigma_\xi, \sigma_\eta) = (0.5, 0.35/\sqrt{n})$. The FON ($n = 10$) was used as a test function. Table 7.2 shows the result of the experiment. The following is understood from Table 7.2.

[1] $\sigma_\xi = 0.5$ and $\sigma_\eta = 0.35/\sqrt{n}$ are recommended parameters in [80].

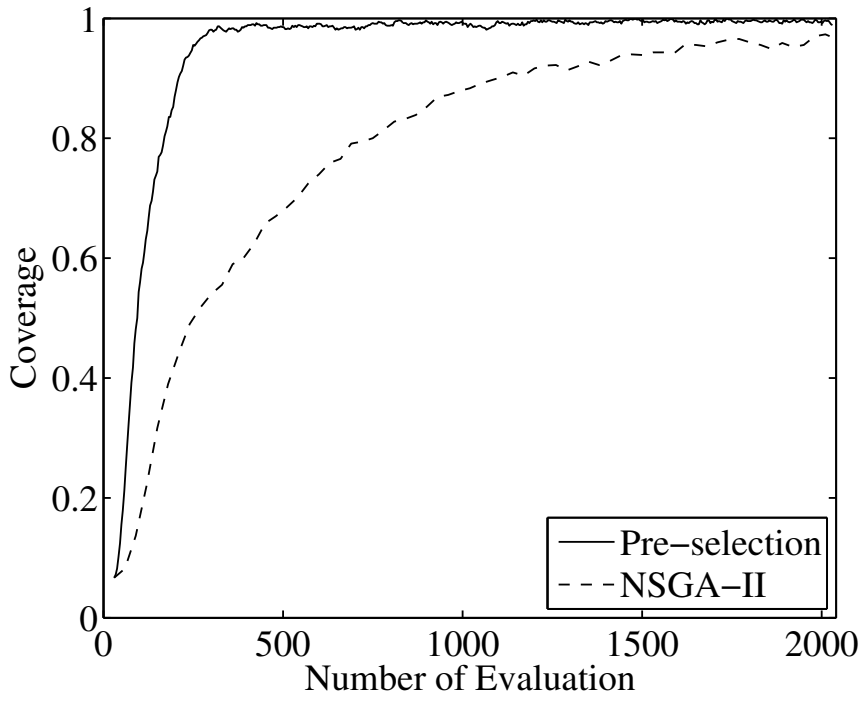


Figure 7.4: Transitions of the coverage of FON in the noise-free environment.

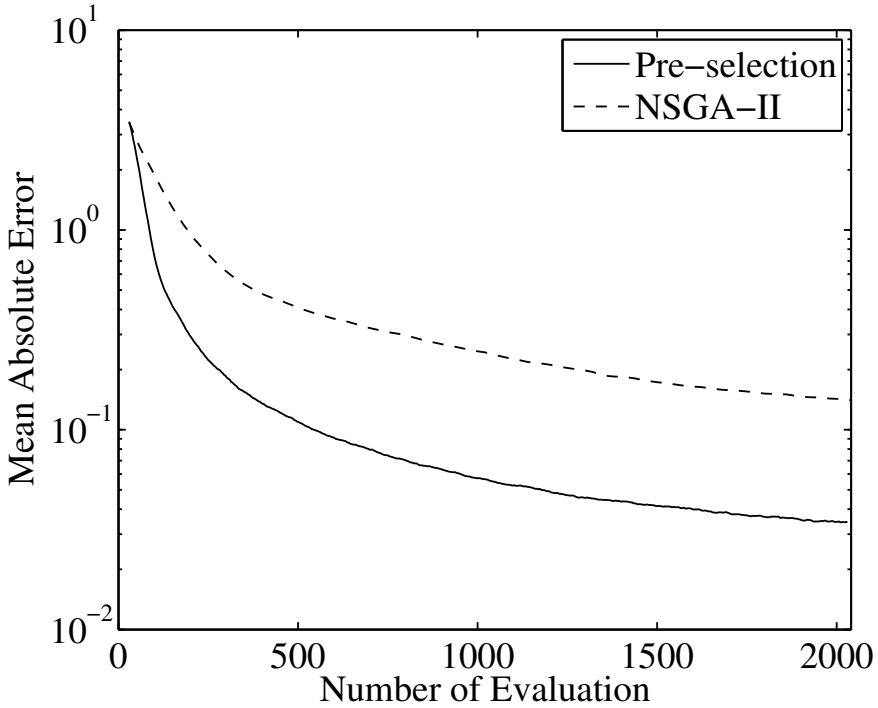


Figure 7.5: Transitions of the mean absolute error of FON in the noise-free environment.

- The search performances slightly improved whenever $|Q_C|$ was increased from 30 to 300. However, since the difference between the performance of $|Q_C| = 100$ and $|Q_C| = 300$ was minor, it is not a good approach to simply increase the size of Q_C in from the perspective of calculation cost.

- The performances of $|Q| = 1$ and $|Q| = 4$ were better than $|Q| = 10$. This result indicates that the promising offspring were surely added to the archived population P when the size of evaluated offspring population was small. However, the number of candidate offspring per evaluated offspring increases if $|Q|$ is set small. Hence, it should be adjusted according to the calculation cost of the approximation model.

- For σ_ξ and σ_η, the recommended values got the best result. When the parameters are small, that is, the offspring are generated in small region, extrapolation is hardly expected. Therefore, mean absolute error deteriorated extremely. However, when they are generated in large region, there is only a minor difference in the coverage for the recommended values while the mean absolute error is decreased. Thus, if the parameters are too large, the density of the candidate offspring generated in the vicinity of the Pareto optimal set becomes small, and convergence of the population slows.

The difference of each performance excluding small value of the UNDX parameters was relatively small for the performance of the normal NSGA-II. Hence, the proposed method has robustness for the design parameter selection.

Table 7.2: Performance comparison of design parameters of the pre-selection algorithm and the UNDX for FON.

Parameters	Value	Coverage		M. A. E.	
		Mean	S. D.	Mean	S. D.
Base	–	0.98889	0.01822	0.03448	0.00538
$\|Q_C\|$	30	0.99111	0.01499	0.04987	0.00712
	300	0.98556	0.02087	0.03205	0.00457
$\|Q\|$	1	1.00000	0.00000	0.02544	0.00444
	10	0.98667	0.01878	0.04987	0.00766
$(\sigma_\xi, \sigma_\eta)$	$(0.25, \frac{0.175}{\sqrt{n}})$	0.98000	0.03456	0.12192	0.04172
	$(1.0, \frac{0.7}{\sqrt{n}})$	0.99556	0.01153	0.06065	0.00697

7.3.3 Performance Analysis under Observation Noise Environments

In this subsection, the performance under the observation noise was examined because it is the objective of this chapter to apply evaluation reduction to experiment-based EMO. A simple expression of noisy test function is defined as follows:

$$F_i(\boldsymbol{x}) = f_i(\boldsymbol{x}) + \delta_i, \quad \delta_i \sim N(0, \sigma_i^2), \quad i = 1, 2, \ldots, m,$$

where, $N(0, \sigma_i^2)$ is the normal distribution random number, and the standard deviation σ_i is set at 10% of the i th range of the Pareto frontier.

In LWR, the observation noise can be filtered by appropriately adjusting parameters such as parameters of weighted function $K(\cdot)$, smoothing parameter u, ridge parameter λ, and choice of local model structure [2]. In this experiment, the ridge parameter was employed for filtering the observation noise. The performances of $\lambda = 0.01,\ 0.1,\ 1$ were examined with the basic setting $\lambda = 0.0001$ which was used in a noise-free environment to avoid the influence of the multicollinearity.

Table 7.3 show the results of experiment. From these results, it is understood that the effect of the ridge parameter adjustment is obvious, but it depends on the test problems. Parameter value $\lambda = 0.01$ indicates a more stable performance than the other values. The ridge parameter should be determined by using cross validation etc., but its calculation cost cannot be neglected. Thus, an indicator of the selection of a more appropriate ridge parameter is one of the future tasks. However, while it is derived from a small amount of experience, we propose the use of $\lambda = 0.01$ or less under noise environments, and set a small value to such an extent that the inverse matrix calculation does not become unstable under the noise free environment as the primal choice.

The transitions of coverage and mean absolute error of FON are shown in Figs. 7.6 and 7.7. These figure indicate that the performance was improved by adjusting the ridge parameter. The result in the noise-free environment showed improvement as well.

From the above-mentioned discussion, it was shown that the proposed method performed better than normal NSGA-II under the observation noise environment.

7.4 Real Engine Experiment

7.4.1 Experiment Settings

A four stroke gasoline engine was used in the experiments to validate the effectiveness of proposed Pre-selection algorithm. Normal NSGA-II and NSGA-II with Pre-selection were applied to a two-objective four-variable function optimization problem of the

Table 7.3: Performance comparison of the NSGA-II with Pre-selection and the normal NSGA-II for four test functions in observation noise environment.

Test Functions	Method	Coverage		M. A. E	
		Mean	S. D.	Mean	S. D.
SCH	Pre-selection	**0.53444**	0.08991	0.87487	0.12791
	PS ($\lambda = 0.01$)	0.51333	0.10042	0.92023	0.15100
	PS ($\lambda = 0.1$)	0.52889	0.13440	0.92978	0.11092
	PS ($\lambda = 1$)	0.23889	0.16449	**0.66298**	0.21311
	NSGA-II	0.53889	0.12505	1.39919	0.25304
	NSGA-II re-eval	0.42889	0.17326	1.21128	0.20914
FON	Pre-selection	0.58667	0.30895	0.24803	0.07067
	PS ($\lambda = 0.01$)	0.81333	0.20652	0.23320	0.03756
	PS ($\lambda = 0.1$)	**0.91444**	0.03784	0.21444	0.02942
	PS ($\lambda = 1$)	0.88667	0.08867	**0.16708**	0.01619
	NSGA-II	0.59111	0.17572	0.46592	0.11830
	NSGA-II re-eval	0.55000	0.25846	0.39342	0.12063
ZDT1	Pre-selection	0.70111	0.15073	**0.02299**	0.00572
	PS ($\lambda = 0.01$)	0.85444	0.10810	0.02391	0.00608
	PS ($\lambda = 0.1$)	**0.86778**	0.07605	0.02490	0.00504
	PS ($\lambda = 1$)	0.44333	0.16682	0.02721	0.00427
	NSGA-II	0.27111	0.13326	0.16417	0.02460
	NSGA-II re-eval	0.18889	0.13542	0.16063	0.02552
DTLZ2	Pre-selection	**0.37006**	0.04029	**0.09112**	0.01048
	PS ($\lambda = 0.01$)	0.33056	0.05498	0.10956	0.01624
	PS ($\lambda = 0.1$)	0.28889	0.04043	0.13826	0.02093
	PS ($\lambda = 1$)	0.13920	0.04737	0.21268	0.02051
	NSGA-II	0.17870	0.05481	0.16272	0.02175
	NSGA-II re-eval	0.31914	0.05128	0.10804	0.01503

engine. In the following, NSGA-II with Pre-selection is called 'Pre-selection'. The setting of multi-objective optimization problem that was used in Chapter 6 is as follows:

Objective Functions: engine torque (maximization), fuel-consumption (minimization).

Decision Variables (Control Parameters): fuel-injection timing (INJ), ignition timing (IGN), desired air-fuel ratio[2] (AFR), valve control parameter (VCP).

Both NSGA-II and NSGA-II with Pre-selection were programmed into a calibration PC. In the calibration PC, objective functions and decision variables were processed

[2]It is ratio of air and gasoline mass

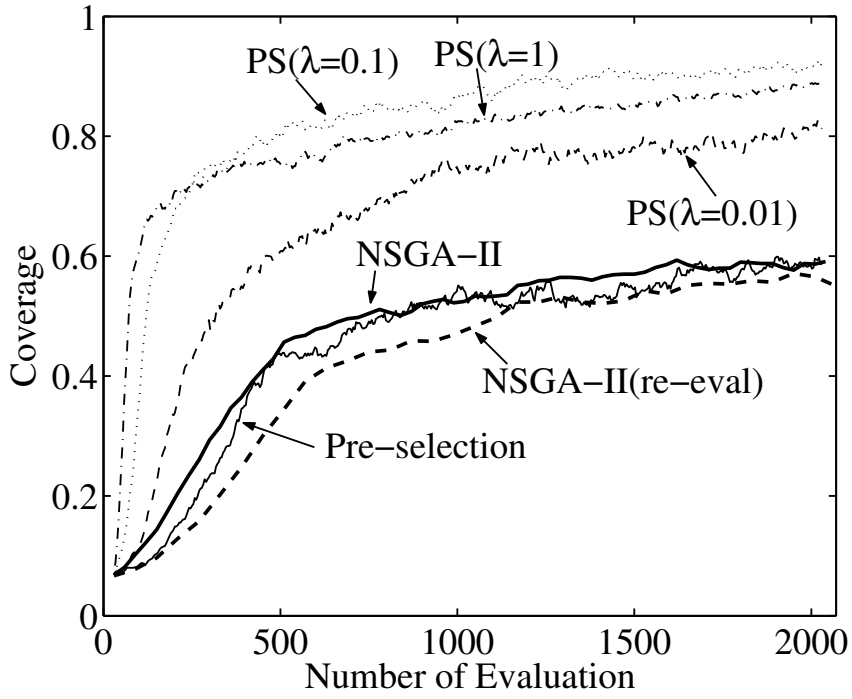


Figure 7.6: Transitions of the coverage of FON in the observation noise environment.

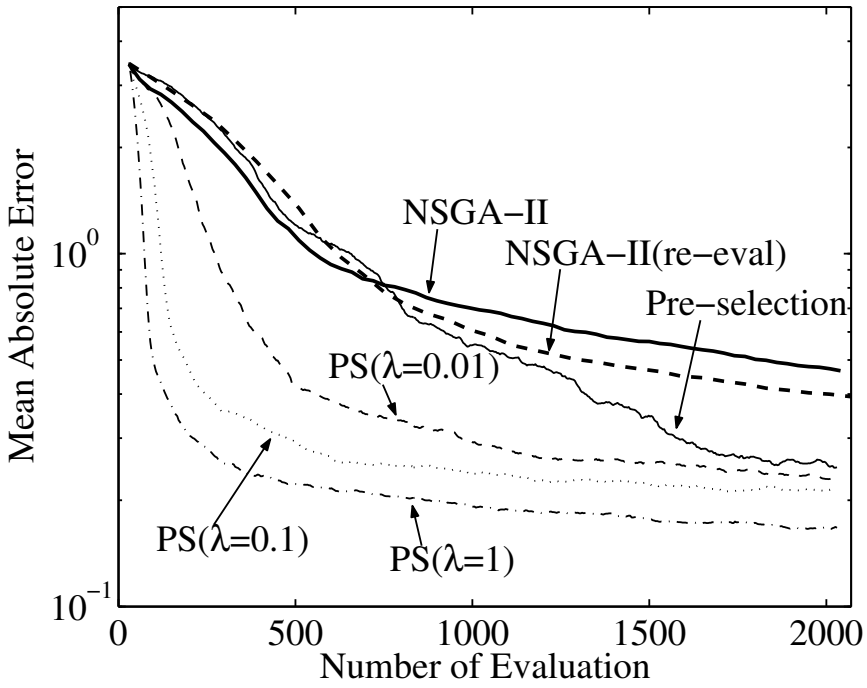


Figure 7.7: Transitions of the mean absolute error of FON in the observation noise environment.

as normalized values. The calibration PC was connected to an ECU by a serial communication.

The experiment condition of NSGA-II is as follows: Population size $|P| = 30$ and offspring population size $|Q| = 30$ were used. The number of evaluations was 1530 (Population P was re-evaluated). The mean value of sampled data was used as estimated value.

The experiment conditions of Pre-selection were as follows: The population size $|P| = 30$, candidate offspring population size $|Q_{\mathrm{C}}| = 100$ and evaluated offspring population size $|Q| = 4$ were used. The number of evaluations was 1530 (P was not re-evaluated). Estimated value was calculated by LWR. Since the dimension of decision variable space is $n = 4$, required size of Ω is $k \geq 9$. In this experiment, k was determined by Eq. (7.8). Additionally, we used $\alpha = 0.05$ as parameter of α-domination strategy, and $\lambda = 0.01$ as ridge parameter.

Both NSGA-II and Pre-selection employed UNDX-P as a crossover operator. The crossover rate was 1.0. Mutation operators were not used.

After optimizations (NSGA-II: 1530 evaluations, Pre-selection: 130 and 1530 evaluations), the true fitness of the individuals were verified through a longer measurement period. These were used for comparison of the approximation Pareto frontier.

7.4.2 LWR for Periodic Functions

As shown in Fig. 7.8, when the k-NN method is used in the vicinity of the edge of the domain of the periodic function, the neighborhood set Ω cannot step over both ends of the domain. Therefore, the approximation model is not smoothly connected and the approximation accuracy is decreased in the vicinity of both ends of the domain. To cope with this problem, we extend the k-NN method and LWR to approximate periodic functions via the following procedures.

Assuming that the elements $x_1, \ldots, x_p$, $p \leq n$ of an individual $\boldsymbol{x}$ represented by n-dimensional vector have a periodicity of 2π cycle by suitable normalization, the elements $x_{p+1}, \ldots, x_n$ do not have periodicity, and domain of each element is normalized to $[0, 1]$.

1. Define extended individuals of $\boldsymbol{x}$ and $\boldsymbol{h}$ stored in search history H as follows:

$$
\begin{aligned}
\boldsymbol{x}_{\mathrm{P}} &= [\cos\theta_1 \ \sin\theta_1 \ \cdots \ \cos\theta_p \ \sin\theta_p \ x_{p+1} \ \cdots \ x_n] \\
\boldsymbol{h}_{\mathrm{P}} &= [\cos\phi_1 \ \sin\phi_1 \ \cdots \ \cos\phi_p \ \sin\phi_p \ h_{p+1} \ \cdots \ h_n] \\
\theta_i &= 2\pi x_i - \pi, \ \phi_i = 2\pi h_i - \pi, \ i = 1, \ldots, p.
\end{aligned}
$$

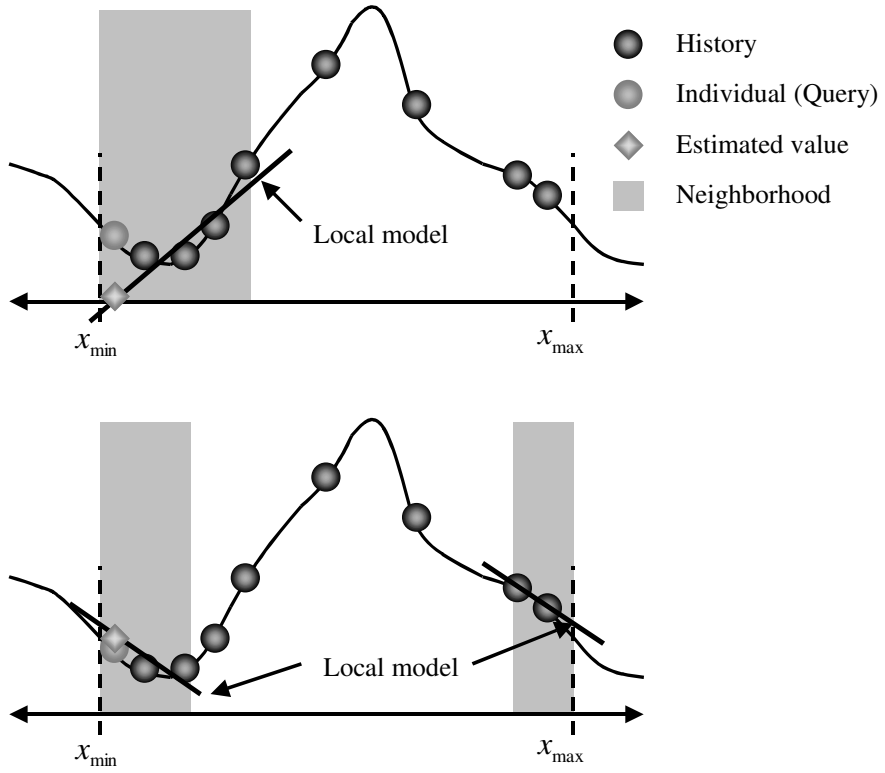


Figure 7.8: Conceptual diagram of the LWR for periodic functions. The upper figure shows the result of conventional LWR and the lower figure shows the result of extended LWR for periodic functions.

2. Define distance between $\boldsymbol{h}_{\mathrm{P}}$ and $\boldsymbol{x}_{\mathrm{P}}$ as the following equation:

$$d_{\mathrm{P}}(\boldsymbol{h}_{\mathrm{P}}, \boldsymbol{x}_{\mathrm{P}}) = \left(\sum_{i=1}^{p} \left(\frac{\cos^{-1}(\cos\phi_i \cos\theta_i + \sin\phi_i \sin\theta_i)}{2\pi} \right)^2 + \sum_{i=p+1}^{n} (h_i - x_i)^2 \right)^{\frac{1}{2}}, \tag{7.9}$$

where $\cos^{-1}$ is treated as a monotonously decreasing function by restricting value region to $[0, \pi]$.

3. Generate the neighborhood set Ω by k-NN method based on d_{P}.

4. Calculate estimated value $\hat{f}(\boldsymbol{x}_{\mathrm{P}})$ of $\boldsymbol{x}_{\mathrm{p}}$ by applying LWR for Ω, where, following

equations are used instead of Eqs. (7.1), (7.3), (7.5).

$$\hat{f}_i(\boldsymbol{x}_{\mathrm{P}}) = \begin{bmatrix} 1 & \cos\theta_1 & \cdots & x_n & x_{p+1}^2 & \cdots & x_n^2 \end{bmatrix} \boldsymbol{b}_i \tag{7.10}$$

$$\boldsymbol{X} = \begin{bmatrix} 1 & \cos\phi_{11} & \cdots & h_{1n} & h_{1,p+1}^2 & \cdots & h_{1n}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos\phi_{k1} & \cdots & h_{kn} & h_{k,p+1}^2 & \cdots & h_{kn}^2 \end{bmatrix} \tag{7.11}$$

$$w_i = \sqrt{K(d_{\mathrm{P}}(\boldsymbol{h}_{\mathrm{P}}^i, \boldsymbol{x}_{\mathrm{P}}))},\ i = 1, 2, \ldots, k. \tag{7.12}$$

5. Employ $\hat{f}(\boldsymbol{x}) = \hat{f}(\boldsymbol{x}_{\mathrm{P}})$ as the estimated value of $\boldsymbol{x}$.

To eliminate the influence of periodic functions, first, each periodic variable is converted to a corresponding point on the unit circle in Step 1). Second, the distance of circular arc is calculated by $\cos^{-1}$ under assumption that an angle between unit vectors exists in $[0, \pi]$. Finally, function approximation is done in a higher dimensional space in Step 4). Incidentally, $\cos^{-1}$ in Eq. (7.9) is a costly function. Therefore to reduce calculation cost, a linear decreasing function can be used instead of $\cos^{-1}$. As another approach, the Euclidean distance between extended individuals can also be employed in place of d_{P}.

7.4.3 Discussion of Results

Analysis in Objective Function Space

Figures 7.9, 7.10, and 7.11 show the population distribution of the estimated value and the true fitness obtained by the measurement for a longer period in the objective function space. Additionally, Fig. 7.12 shows the comparison of the approximation Pareto frontiers by true fitness. The horizontal axis indicates engine torque, and the vertical axis indicates fuel-consumption respectively. Each axis is shown in normalized scale. Hence, the direction of optimization is in the lower right part of the graph.

- The Pareto frontier approximated by population distributions shows straight line shape. The torque proportionally moves from 0.5 to 0.8 provided that the fuel-consumption changes from 0.55 to 0.8 in objective function space.

- In the comparison of true fitness, two approximation Pareto frontiers by Pre-selection are especially improved around fuel-consumption of 0.8 compared with that by NSGA-II. This is because Pre-selection does not receive the influence of noise easily as a result of using the estimation value to search, since LWR can calculate accurate estimated values for data which contain uncertainties.

- In Pre-selection, the estimation error of the fuel-consumption was larger than that of the torque, because the response delay of the fuel-consumption is larger than that of the torque in these experiments.

- In Pre-selection, there were few differences between the approximation Pareto frontiers of 130 evaluations and that of 1530 evaluations. This result indicates that the Pre-selection works effectively and accelerates the search process remarkably.

Analysis in Decision Variable Space

Figures 7.13, 7.14, and 7.15 indicate population distribution in decision variable space. Each axis is also shown in normalized scale. They indicate the following results:

- A strong correlation exists between fuel-injection timing INJ and the valve control parameter VCP. INJ approaches from 0.5 to 0.8 provided that the VCP moves from 0 to 1.
- A weak correlation is seen between ignition timing IGN and VCP.
- Because the desired air-fuel ratio AFR converged towards the vicinity of 1, which is the limit, the improvements in performance can be expected by expanding the search space. In fact, engine calibration experts pointed out that the upper limit of AFR is lower than usual, and that the results are reasonable under the experiment conditions.
- VCP is distributed widely in the range from 0 to 0.5 compared with other parameters. It is thought that the trade-off strongly depends on VCP.
- In the comparison of decision variable space, population of Pre-selection showed better convergence than that of NSGA-II.

Effectiveness of Pre-selection

In the experiment of NSGA-II, it was found that the population converged at around 1000 evaluations, via comparison with the population of final generation. Therefore, the optimization time was decreased to about 13% of the original time needed by applying Pre-selection. Through discussion with experts of engine calibration, we confirmed that the characteristics of the Pareto approximation set were an appropriate representation of the performance of the engine, and that the optimization time achieved a practical level when compared to the operation time needed by experts. Hence, Pre-selection is an essential technique of EBEMO for engine calibration was validated.

7.5 Summary

In this chapter, a Pre-selection algorithm for EBEMO was proposed. First, the performance of proposed method was examined through numerical experiments. As a result,

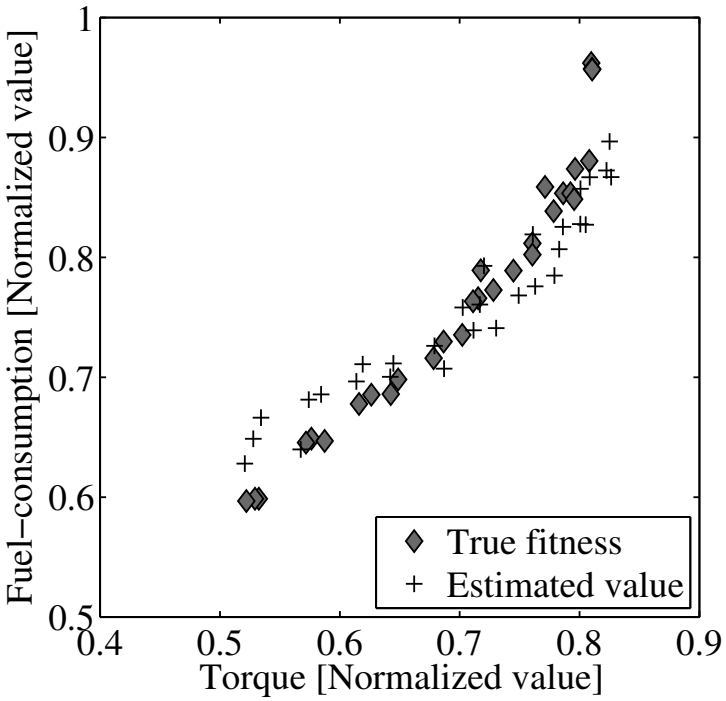


Figure 7.9: Population distribution in the objective function space (NSGA-II, 1530 evaluations).

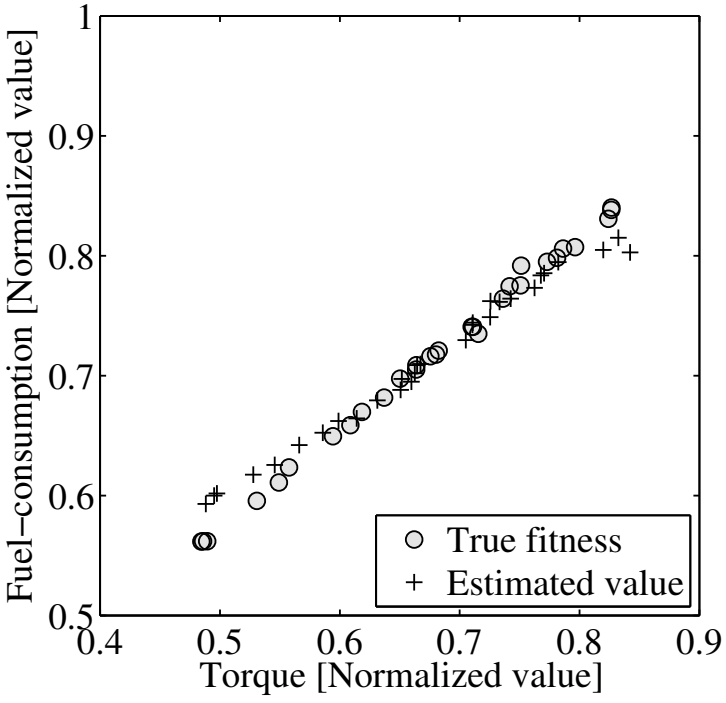


Figure 7.10: Population distribution in the objective function space (Pre-selection, 130 evaluations).

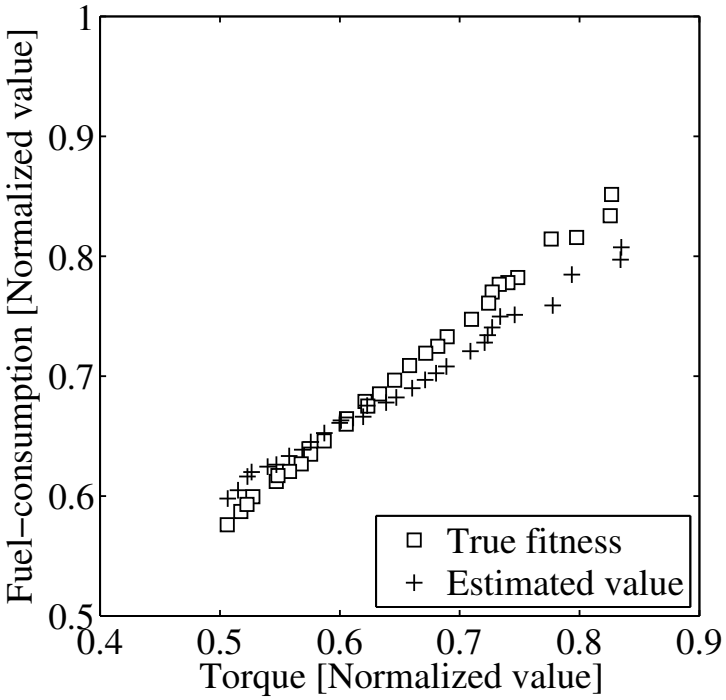


Figure 7.11: Population distribution in the objective function space (Pre-selection, 1530 evaluations).

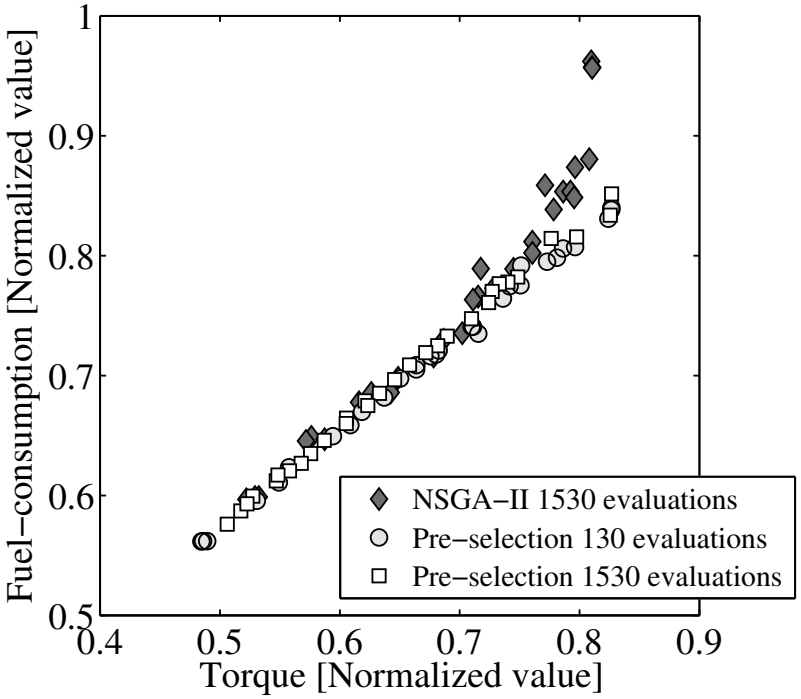


Figure 7.12: Comparison of the approximation Pareto frontier in the objective function space by using true fitness.

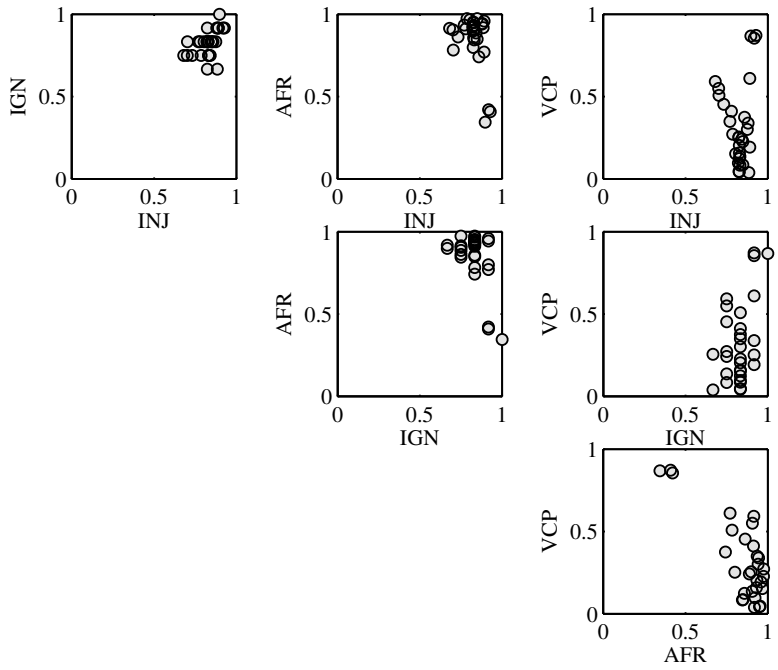


Figure 7.13: Population distribution in the decision variable space (NSGA-II, 1530 evaluations).

it was confirmed that the Pre-selection was able to reduce the number of evaluations greatly for four test functions despite the presence of the observation noise. Moreover, the search performance did not change significantly when the setting parameters of Pre-selection and UNDX were changed. These facts indicate that the proposed method has robustness for the setting parameters.

Next, the Pre-selection was applied to EBEMO of real internal-combustion engines. To demonstrate that MOEAs can optimize real engines in practical time by applying Pre-selection, a real engine was optimized as a two-objective optimization problem. As a result, it was confirmed that NSGA-II with Pre-selection was able to reduce optimization time to about 13% together with improving search accuracy, and that the achieved the optimization time at a practical level.

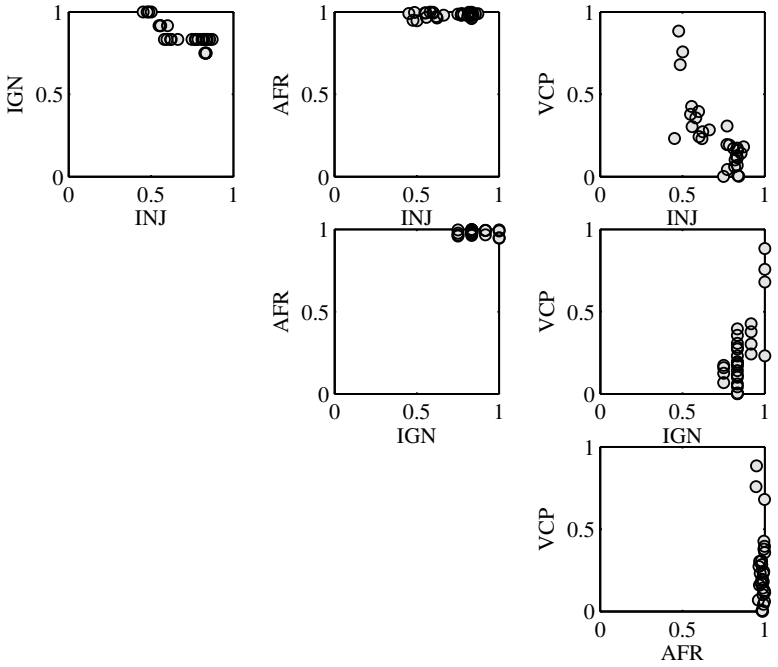


Figure 7.14: Population distribution in the decision variable space (Pre-selection, 130 evaluations).

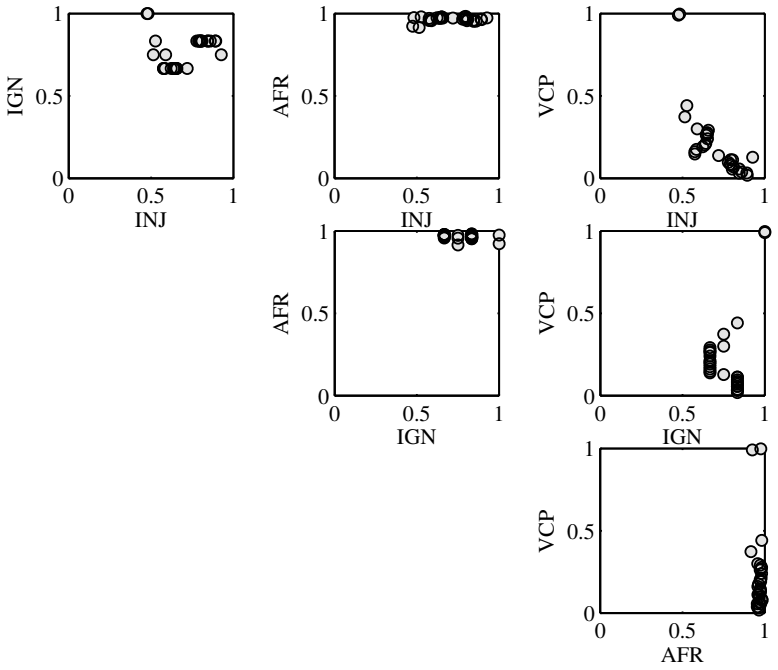


Figure 7.15: Population distribution in the decision variable space (Pre-selection, 1530 evaluations).

Chapter 8

Parametric Multi-objective Optimization

8.1 Outline

In a map-based control, the maps programmed in Electronic Control Units (ECUs) are represented by a grid of engine operating conditions defined by input variables such as engine speed and load torque. As described in Chapter 2, desired values of control devices are stored as parameters corresponding to the operating points on the grid, and calculated by linear interpolation of these values for the input variables that change continuously. Therefore, we have to find optimal control parameters by experiment on each operating point, and integrate them as maps.

Figure 8.1 shows a conceptual diagram of engine calibration using EBEMO. First, an operating condition is chosen. Second, a Pareto approximation set that is a set of control parameters such as ignition timing and fuel injection timing are obtained by MOEAs under plural evaluation criterion such as emission of NO_x and HC. Third, the most preference solution is selected. Finally, the corresponding control parameters are stored at the operating condition on the maps. Therefore, to complete the entire maps, we have to solve many MOPs defined by operating conditions by the aforementioned steps one at a time. In this thesis, variables such as the operating condition of engine calibration are called the 'condition variable'.

We focus on the problems that have to solve plural MOPs defined by condition variables. In real problems such as engine calibration, even to solve a single MOP requires a lengthy computation time, and hence in solving sets of MOPs, reduction of computation time is a very important issue. In this chapter, first, Parametric Multi-objective Optimization Problems (PMOPs) are defined as a new class of MOPs. Then, an initialization method for MOEAs is proposed to reduce the total computation time. In the proposed method, the Pareto approximation set of a MOP not already solved is estimated by using plural Pareto approximation sets already obtained and then used to initialize a new MOEA process. Moreover, the effectiveness of proposed method is

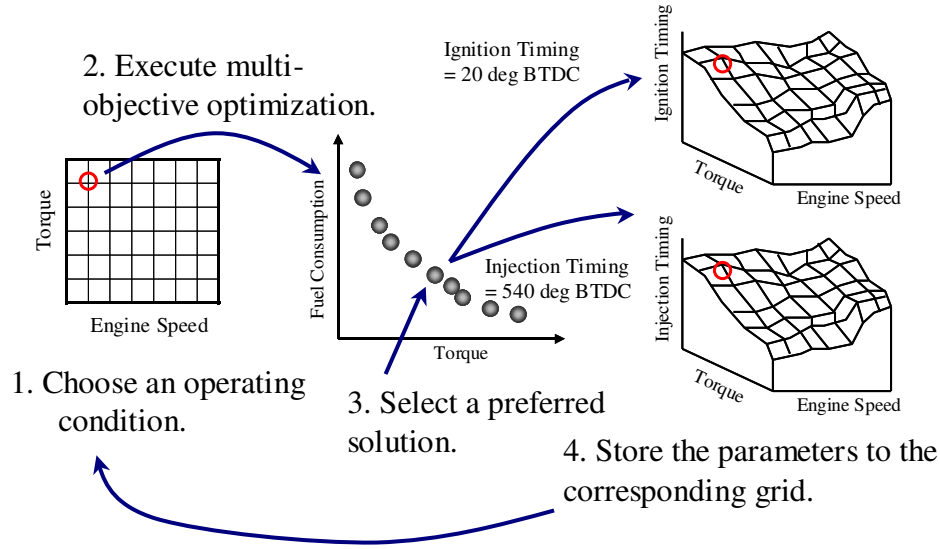


Figure 8.1: Multi-objective optimization of engine control parameters under an operating condition.

confirmed through simple numerical experiments.

8.2 Problem Formulation

In formulation of decision making problems, we have to consider two types of variables. The first variables to decide are the variables needed to achieve design objectives, and the second variables are the ones given externally. In this thesis, we call the first ones 'decision variables' and the second ones 'condition variables'. A MOP that has condition variables is defined as follows:

MOP($\boldsymbol{u}$):

$$\begin{aligned} &\text{Given } \boldsymbol{u} \in U \subset \mathbf{R}^l \\ &\min_{\boldsymbol{x}} \ \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) = [f_1(\boldsymbol{x}, \boldsymbol{u}) \ \cdots \ f_m(\boldsymbol{x}, \boldsymbol{u})]^{\mathrm{T}} \\ &\text{subject to } \boldsymbol{x} \in X(\boldsymbol{u}) \subset \mathbf{R}^n, \end{aligned}$$

where $\boldsymbol{x}$ is n-dimensional decision variable vector, $\boldsymbol{u}$ is l-dimensional condition variable vector and $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ is m-dimensional objective function vector. $X(\boldsymbol{u})$ is the feasible region of $\boldsymbol{x}$ determined by $\boldsymbol{u}$, U is a condition variable set. Figure 8.2 depicts the objective function landscape of MOP with condition variables and their Pareto optimal sets $X^*(\boldsymbol{u})$. If condition variables are changed from $\boldsymbol{u}^1$ (Fig. 8.2(a)) to $\boldsymbol{u}^2$ (Fig. 8.2(b)), the landscape of objective functions change, and consequently the Pareto optimal solution also changes.

In the study of Evolutionary Multi-objective Optimization (EMO), research that

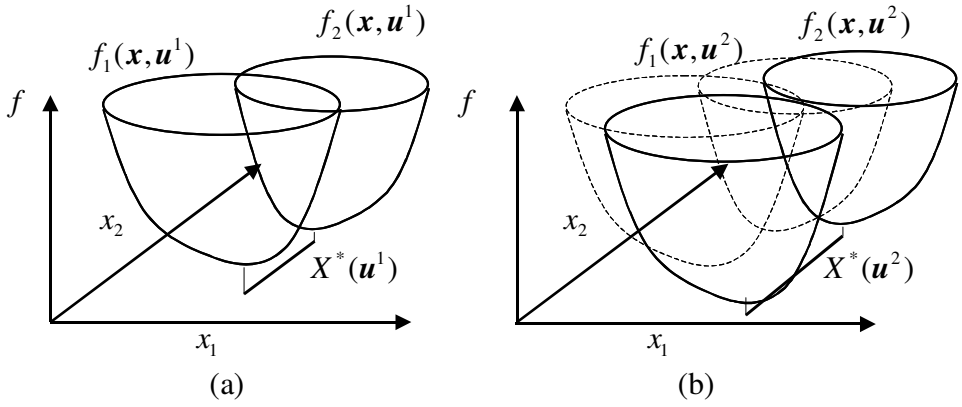


Figure 8.2: Schematic diagram of MOPs with condition variables.

treats the condition variables explicitly has recently been increasing because of the need to deal with various real problems [32, 57, 58, 83, 112].

8.2.1 Related Works

Various MOPs can be formulated with various condition variable $u \in \mathbf{R}$. An optimization process of a normal MOP is represented by $u = \text{const}$. Random initialized population converges gradually to the Pareto optimal set $X^*(u)$.

If the condition variable changes randomly, and is not available in the deciding of $\boldsymbol{x}$, it represents an uncertainty MOP such as experiment-based or simulation-based optimization including randomness. Many methods have been proposed to obtain a more probable Pareto approximation solution [58].

A MOP that changes depending on a time t, it is called a Dynamic MOP (DMOP) [32, 57, 73]. One requirement of MOEAs for DMOPs is to track $X^*(u)$ that continuously changes. To achieve this goal, Zhou et al. have proposed a re-initialization method by using current and previous generation populations to predict the next generation population [112].

Ponweiser et al. have proposed Multiple-MOP (M-MOP) [83]. M-MOP is defined as a problem that gains a solution set that satisfies simultaneously plural MOPs defined by condition variables. They applied this formulation to the parameter optimization of image data processing.

8.2.2 Parametric Multi-objective Optimization Problems

In this chapter, we formulate a new class of MOPs called Parametric Multi-objective Optimization Problems (PMOPs). PMOP is defined as a problem to obtain the Pareto optimal sets of plural MOPs defined by the condition variable vectors. When a set of L condition variable vectors is given, PMOP is defined as a problem that has L MOPs

as sub-problems. Formulation of PMOP is as follows:

PMOP:

$$\text{Solve MOP}(\boldsymbol{u}^i), \text{ for all } i = 1, 2, \ldots, L$$

where MOP($\boldsymbol{u}^i$) is defined by

$$\begin{aligned} \min_{\boldsymbol{x}} \ & \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}^i) = [f_1(\boldsymbol{x}, \boldsymbol{u}^i) \ \cdots \ f_m(\boldsymbol{x}, \boldsymbol{u}^i)]^{\mathrm{T}} \\ & \text{subject to } \boldsymbol{x} \in X(\boldsymbol{u}^i) \subset \mathbf{R}^n, \ \boldsymbol{u}^i \in U \subset \mathbf{R}^l. \end{aligned}$$

A family of sets $\mathcal{X}^* = \{(\boldsymbol{u}^1, X^*(\boldsymbol{u}^1)), \ldots, (\boldsymbol{u}^L, X^*(\boldsymbol{u}^L))\}$ is the solution of PMOP, where $X^*(\boldsymbol{u})$ is the Pareto optimal set of MOP($\boldsymbol{u}$).

Figure 8.3 indicates a optimization process of PMOP that have $U = \{\boldsymbol{u}^1, \boldsymbol{u}^2, \boldsymbol{u}^3, \boldsymbol{u}^4\}$ by MOEAs. In this thesis, the population of MOEAs for MOP($\boldsymbol{u}$) in tth generation is called a 'halfway population', and is denote by $P(t, \boldsymbol{u})$. The population in the final generation is called Pareto approximation set, and is denote by $P(*, \boldsymbol{u})$. MOEAs are applied independently for each MOP. Finally, the approximation solution of $\mathcal{X}^*$ is obtained as a family of sets of Pareto approximation solution $\mathcal{P}^* = \{(\boldsymbol{u}^1, P(*, \boldsymbol{u}^1)), \ldots, (\boldsymbol{u}^L, P(*, \boldsymbol{u}^L))\}$. In previous research, such as Ponweiser's M-MOP, MOPs are optimized only once. Whereas PMOP is optimized through plural MOPs optimization. This is the major difference between the problem formulation of PMOPs and others.

8.3 Requirements for MOEAs in Real Applications

When a PMOP is given by numerical formulas and the evaluation cost is small enough, it only has to optimize all MOPs to solve the PMOP. However, in experiment-based and simulation-based optimization used in real applications, they require large evaluation costs even for a single MOP. Hence, the total evaluation cost of the PMOP becomes enormous. For example, it is common that 100 or more condition variable vectors are set in development of a map in engine calibration. If the number of condition variable vectors is 100 and MOEAs can solve a MOP in one hour, the total optimization time to gain $\mathcal{P}^*$ takes 100 hours.

Along with the increasing complexity of engine control systems, engine calibration is becoming a time consuming process in engine development [88, 89]. To accomplish this process efficiently, shortening of the calibration time is a major request from the automotive industry. Therefore, the reduction methods of evaluation cost for PMOPs are indispensable.

When a calibration engineer calibrates a map manually, it is natural approach to choose the next selected condition variable on the basis of past experience. For example, the landscapes of engine torque and fuel-consumption to the control parameters

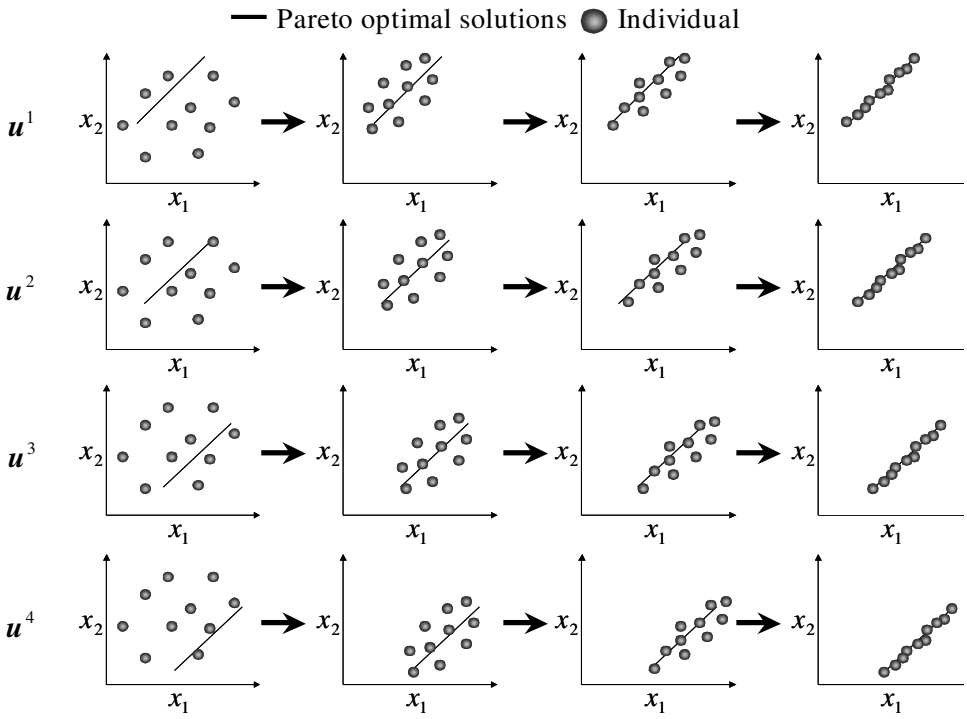


Figure 8.3: Optimization process in decision space of PMOP by MOEAs.

(air fuel ratio AFR, valve control parameter VCP) of a real engine operated at four engine speeds is shown in Figs. 8.4 and 8.5. Each axis is normalized. In Fig. 8.4, the engine torque landscape is inverted, that is, the low torque index indicates large torque. We can see that the landscapes change continuously depending on the engine speed.

We assume a continuity of objective functions for condition variables. Under this assumption, we try to improve the search efficiency of MOEAs for PMOP by using the history data obtained in the past.

Figure 8.6 shows a conceptual diagram of our proposal to improve the efficiency by using interpolation of plural Pareto approximation sets for initialization operation. First, MOP($\boldsymbol{u}^1$) and MOP($\boldsymbol{u}^4$) are optimized by corresponding initial populations $P(0, \boldsymbol{u}^1)$ and $P(0, \boldsymbol{u}^4)$ that are initialized randomly. Then, $P(0, \boldsymbol{u}^2)$ is generated by the interpolation of two Pareto approximation sets $P(*, \boldsymbol{u}^1)$ and $P(*, \boldsymbol{u}^4)$, and MOP($\boldsymbol{u}^2$) is optimized by using $P(0, \boldsymbol{u}^2)$ as the initial population. After that, MOP($\boldsymbol{u}^3$) is optimized by using $P(0, \boldsymbol{u}^3)$ that is interpolated from $P(*, \boldsymbol{u}^2)$ and $P(*, \boldsymbol{u}^4)$. As a result, the total evaluation cost of PMOP can be reduced because the gray area can be omitted (As can be seen when comparing Fig. 8.6 to Fig. 8.3 by the aforementioned operation.).

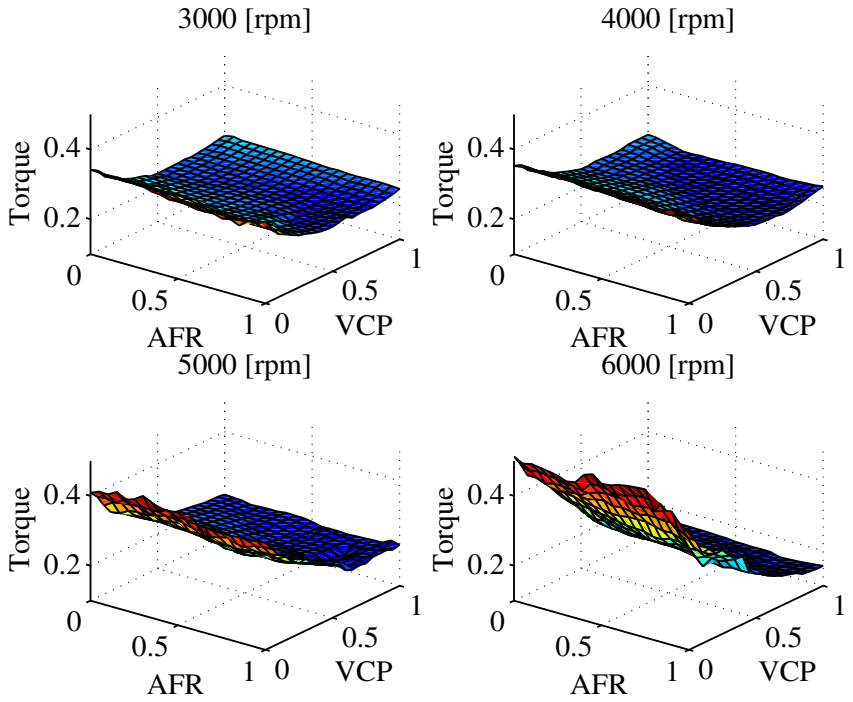


Figure 8.4: Change of engine torque landscape by engine speed.

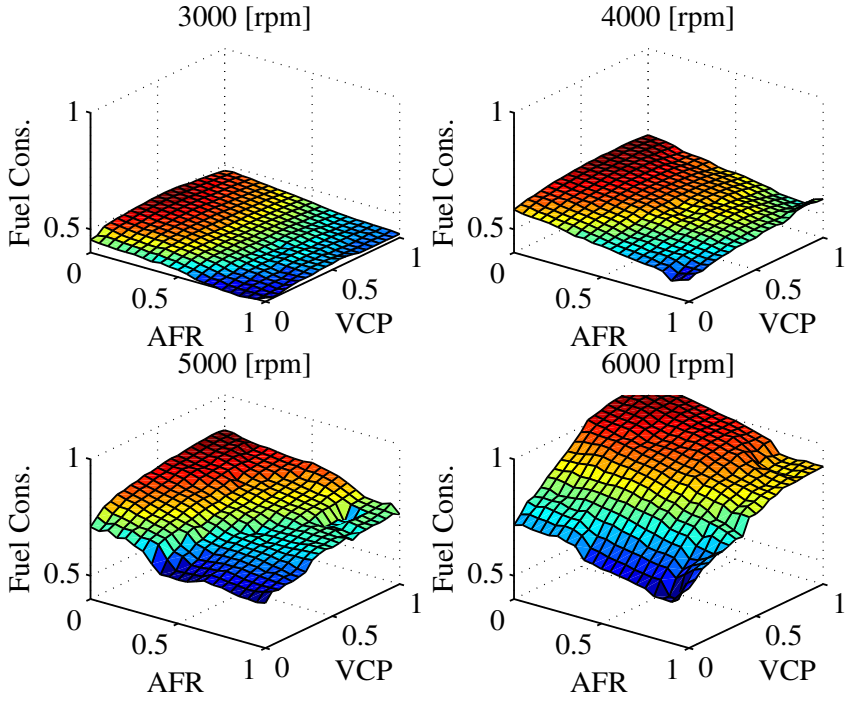


Figure 8.5: Change of fuel-consumption landscape by engine speed.

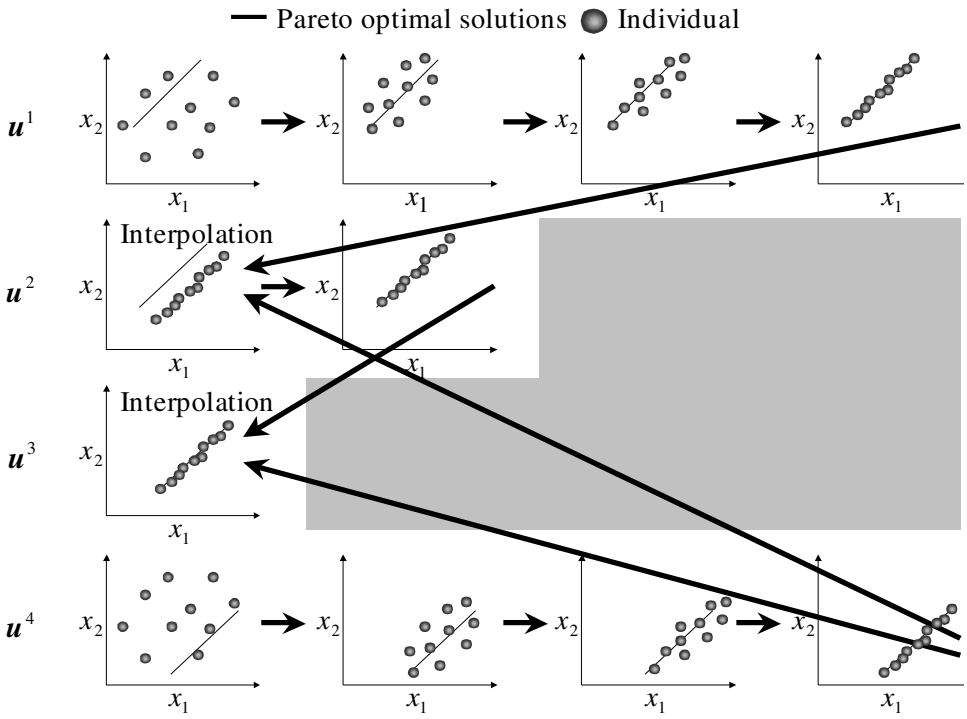


Figure 8.6: Optimization process in decision space of PMOP by MOEAs using an interpolative initialization method.

8.4 MOEAs for Parametric Multi-objective Optimization Problems

In this section, we propose MOEAs for PMOPs. The proposed method is composed by following three modules and a memory shown in Fig. 8.7:

Condition Variable Scheduler: This module selects a condition variable $\boldsymbol{u}^{\mathrm{q}}$ that determines solved MOP($\boldsymbol{u}^{\mathrm{q}}$) from the condition variable set U.

Initialization Operator: This module generates an initial population $P(0, \boldsymbol{u}^{\mathrm{q}})$ of MOP($\boldsymbol{u}^{\mathrm{q}}$) by following two methods:

1. **Random Initialization Method**: An initial population is randomly generated in the search space by using a uniform random number.
2. **Interpolative Initialization Method**: An initial population is generated as an interpolated Pareto approximation set of MOP($\boldsymbol{u}^{\mathrm{q}}$) by using condition variable vectors and corresponding halfway and Pareto approximation sets stored in the search history that is described below.

These two methods should be used properly according to the number of condition variable vectors stored in the search history.

Multi-objective Evolutionary Algorithm: This module optimizes MOP($\boldsymbol{u}^{\mathrm{q}}$) by using $P(0, \boldsymbol{u}^{\mathrm{q}})$ generated by the initialization operator. Standard MOEAs such as NSGA-II can be adopted for this module. In this chapter, the population size $|P|$ is same for MOP($\boldsymbol{u}^i$), $i = 1, \ldots, L$.

Search History: This module stores each condition variable vector and corresponding halfway and Pareto approximation solutions as pair $(\boldsymbol{u}, P(t, \boldsymbol{u}))$. Preservation of the halfway populations is for extension of the algorithm in the future and is not essential in this chapter.

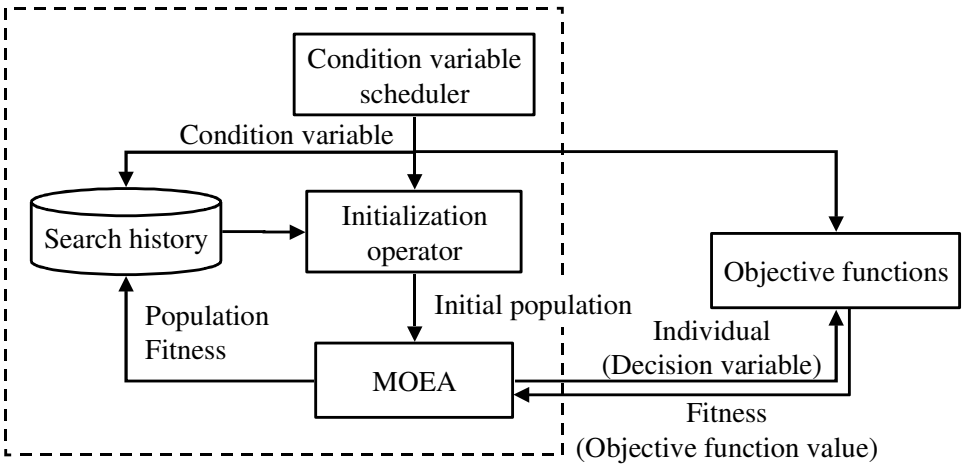


Figure 8.7: Block diagram of a MOEA for PMOP.

8.4.1 Interpolative Initialization Method

We assume continuity of objective functions and condition variables of PMOPs we proposed. To reduce the total evaluation cost under this assumption, an interpolative initialization method that interpolates plural Pareto approximation sets previously searched and utilization of the interpolated population for initial population of MOP($\boldsymbol{u}^{\mathrm{q}}$) to be solved. The conceptual diagram of the interpolative initialization method is shown in Fig. 8.8, and the algorithm is as follows:

1. Construct a neighborhood set Ψ of query $\boldsymbol{u}^{\mathrm{q}}$ by k-nearest neighbor (k-NN) method from condition variable vectors stored in the search history. Size of Ψ should be $k \geq l + 1$ when first order polynomial is used for regression, where l is the dimension of condition variable space.

2. Compose a design matrix $\boldsymbol{U}$ by using k condition variable vectors, where $\boldsymbol{U}$ is

$k \times (l+1)$ dimensional matrix

$$\boldsymbol{U} = \begin{bmatrix} 1 & u_{11} & \cdots & u_{1l} \\ & & \vdots & \\ 1 & u_{k1} & \cdots & u_{kl} \end{bmatrix}, \tag{8.1}$$

and u_{ij} is jth element of ith nearest vector $\boldsymbol{u}$ to $\boldsymbol{u}^{\mathrm{q}}$ in Ψ.

3. Let $P^1 \in \Psi$ be population corresponding $\boldsymbol{u}$ that is the nearest to $\boldsymbol{u}^{\mathrm{q}}$. Make up pairs for all individual in the reference population P^1 and $k-1$ populations $P^2, \ldots, P^k$, where $P^1, \ldots, P^k$ are the Pareto approximation sets corresponding Ψ.

4. Set $i = 1$.

5. Compose a vector $\boldsymbol{y}_j^i = [x_j^{1,i} \ \cdots \ x_j^{k,i}]^{\mathrm{T}}$ by jth elements of $\boldsymbol{x}^{1,i}, \ldots \boldsymbol{x}^{k,i}$, where $\boldsymbol{x}^{1,i}$ is ith individual in P^1, $\boldsymbol{x}^{2,i}, \ldots \boldsymbol{x}^{k,i}$ are individuals that make up pairs with $\boldsymbol{x}^{1,i}$ in $P^2, \ldots, P^k$.

6. Set $j = 1$.

7. Calculate jth element of a estimated individual $\hat{x}_j^i$ by

$$\hat{x}_j^i = [1 \ u_1^{\mathrm{q}} \ \cdots \ u_l^{\mathrm{q}}]\boldsymbol{\gamma}_i^j \tag{8.2}$$
$$\boldsymbol{\gamma}_j^i = (\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})^{-1}\boldsymbol{U}\boldsymbol{y}_j^i. \tag{8.3}$$

8. If $j = n$, compose the estimated individual $\hat{\boldsymbol{x}}^i = [\hat{x}_1^i \ \ldots \ \hat{x}_n^i]$; otherwise set $j := j + 1$ and return to Step 7).

9. If $i = |P|$, output $\hat{\boldsymbol{x}}^i, i = 1, \ldots, |P|$ as $P(0, \boldsymbol{u}^{\mathrm{q}})$; otherwise set $i := i + 1$ and return to Step 5), where $|P|$ is the population size of MOEAs.

In Step 3), optimal matching that minimizes the sum of the Euclidean distance between two sets is executed to calculate the interpolated population. This problem can be treated as a linear programming problem. However, 2-opt method is used to solve optimal matching in this thesis.

In Step 7), the calculation of the inverse matrix $(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})^{-1}$ becomes unstable numerically, if the dimension of the neighborhood set Ψ reduces. In this thesis, the random initialization method is used to avoid this problem when the minimum absolute value of the eigenvalues of $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}$ are smaller than a small positive constant ε.

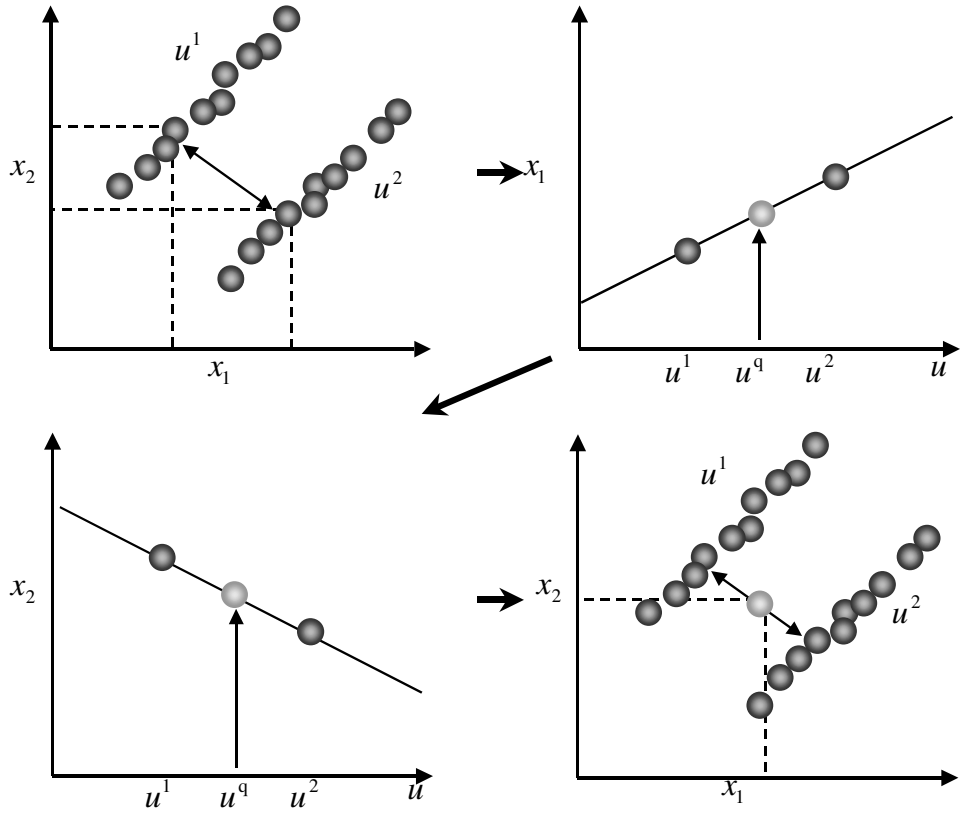


Figure 8.8: Conceptual diagram of interpolation of Pareto approximation set ($k = 2, n = 2, l = 1$).

8.4.2 Condition Variable Scheduling

The interpolative initialization method generates the initial population by using plural Pareto approximation sets stored in the search history. Hence, the order of selecting condition variable vectors from the set U have a large influence on the efficiency of solving PMOPs.

The following two types of selection order can be considered:

Type A: First condition variable vectors are selected to cover the entire condition variable space, then interpolation is repeated after that.

Type B: A condition variable vector of neighborhood is selected, and extrapolation is repeated.

Figure 8.9 illustrates the conceptual diagrams of the two scheduling algorithms of U in two-dimensional space.

In this chapter, Type A and B algorithms are used for condition variable scheduling as example implementation. The type A scheduling algorithm is shown below:

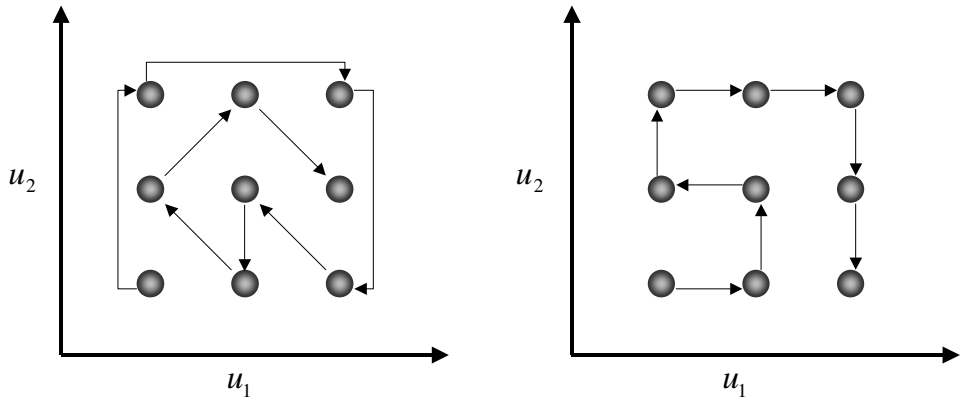


Figure 8.9: Scheduling of condition variable vectors (left: Type A, right: Type B).

1. Obtain a subset U^+ that forms the convex hull [4] of U, and add them to a list Z.
2. Select a condition variable vector $\boldsymbol{u}^* \in U^-$ that has a maximum Euclidean distance from U^- to the nearest condition variable vector in U^+. $U^- := U \backslash U^+$ is the complementary set of U^+.
3. Add $\boldsymbol{u}^*$ to the tail of Z, and update $U^+ := U^+ \cup \{\boldsymbol{u}^*\}$, $U^- := U \backslash U^+$.
4. If $U^- = \emptyset$, output Z as the scheduled list of U; otherwise, return to Step 2).

In addition, the type B scheduling algorithm is describe as follows.

1. Obtain a subset U^+ that forms a first evaluated condition variable vector in U, and add them to a list Z.
2. Select $\boldsymbol{u}^* \in U^-$ that has minimum sum of Euclidean distance from U^- to all condition variable vectors in U^+
3. Add $\boldsymbol{u}^*$ to the tail of Z, and update $U^+ := U^+ \cup \{\boldsymbol{u}^*\}$, $U^- := U \backslash U^+$.
4. If $U^- = \emptyset$, output Z as the scheduled list of U; otherwise, return to Step 2).

8.5 Numerical Experiment

8.5.1 Experiment Settings

The effectiveness of proposed method was examined through a numerical experiment. We used the following two-condition two-objective n-variable problems:

- TF-1

$$f_1(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{n-2}\left(\sum_{i=1}^{2}(x_i - u_i^2)^2 + \sum_{i=3}^{n}(x_i - 1)^2\right) \tag{8.4}$$

$$f_2(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{n-2}\left(\sum_{i=1}^{2}(x_i - u_i^2)^2 + \sum_{i=3}^{n}(x_i + 1)^2\right), \tag{8.5}$$

$$x_i \in [-2, 2], \quad i = 1, \ldots, n \tag{8.6}$$

$$u_j \in [0, 1], \quad j = 1, 2 \tag{8.7}$$

  where $\boldsymbol{x}$ is the decision variable vector, and $\boldsymbol{u} = [u_1\ u_2]^{\mathrm{T}}$ is the condition variable vector. Note that u_j is squared in Eqs. (8.4)(8.5). This operation intentionally causes regression error by the first order polynomial. Hence, the influence of the regression error can be confirmed. The Pareto optimal set of TF-1 is $X^*(\boldsymbol{u}) = \{x_1 = u_1^2, x_2 = u_2^2, x_3 = x_4 = \cdots = x_n, x_i \in [-1, 1], i = 3, \ldots, n\}$.

- TF-2

$$f_1(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{n-2}\left((x_1 - \sin(u_1\pi))^2 + (x_2 - \sin(u_2\pi))^2 + \sum_{i=3}^{n}(x_i - 1)^2\right) \tag{8.8}$$

$$f_2(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{n-2}\left((x_1 - \sin(u_1\pi))^2 + (x_2 - \sin(u_2\pi))^2 + \sum_{i=3}^{n}(x_i + 1)^2\right) \tag{8.9}$$

$$x_i \in [-2, 2], \quad i = 1, \ldots, n \tag{8.10}$$

$$u_j \in [0, 1], \quad j = 1, 2 \tag{8.11}$$

  In this problem, the position of the Pareto optimal set does not correspond to the condition variable vectors individually. The four corners of condition variable space $[0\ 0], [0\ 1], [1\ 0], [1\ 1]$ have same Pareto optimal set especially. Hence, if these four points are selected as the initial condition variable vectors, the interpolative initialization method does not work well. The Pareto optimal set of TF-2 is $X^*(\boldsymbol{u}) = \{x_1 = \sin(u_1\pi), x_2 = \sin(u_2\pi), x_3 = x_4 = \cdots = x_n, x_i \in [-1, 1], i = 3, \ldots, n\}$.

In this experiment, $n = 20$ is used. The codomain of condition variable vector is $[u_1\ u_2] \in \{0.0, 0.1, \ldots, 1.0\}^2$. All the combinations of u_1 and u_2 are in U. Therefore, the size of U is $L = 121$.

In this chapter, NSGA-II was employed as a MOEA. In the experiment settings of NSGA-II, population size $|P| = 100$ and offspring population size $|Q| = 100$ were used. UNDX proposed by Ono et al. [80] was used as a crossover operator for real-coded GA. The crossover rate was 1.0. Mutation operators were not adopted in this experiment.

In the proposed method, U is scheduled at first and $\boldsymbol{u}^{\mathrm{q}}$ is selected from the list Z one at a time. Both, Type A and B algorithms are applied to TF-1 and TF-2. Next,

if the number of $\boldsymbol{u}$ stored in the search history does not reach the neighborhood size k, random initialization method is executed. Otherwise, interpolative initialization method is executed. After initialization operation, MOEA optimizes MOP($\boldsymbol{u}^{\mathrm{q}}$) until Tth generations. The halfway population $P(t, \boldsymbol{u}^{\mathrm{q}})$ and the final population $P(T, \boldsymbol{u}^{\mathrm{q}})$ are stored to the search history. In this experiment, $T = 100$ and $k = 4 > l+1 = 2+1$ were used, where $l = 2$.

8.5.2 Evaluation Measures

Following two evaluation measures were used:

- Minimum generation in which the proposed method achieved the same precision of Pareto approximation set obtained by random initialization method.
- Relative precision of Pareto approximation set obtained by using the random and interpolative initialization method at Tth generation.

At first, 100 points including both ends of the Pareto optimal set $X^*(\boldsymbol{u})$ are sampled at equal intervals. This set is called the reference Pareto optimal set $X'(\boldsymbol{u})$. The precision of $P(*, \boldsymbol{u})$ is represented by a similarity measure α with $X'(\boldsymbol{u})$.

The similarity measure $\alpha(R, S)$ of two sets R and S was defined by the following equation:

$$\alpha(R,S) = \frac{1}{2}\left(\frac{1}{|R|}\sum_{i=1}^{|R|} \min_{k=1}^{|S|} \sqrt{\sum_{j=1}^{n}(x_j^i - x_j'^k)^2} + \frac{1}{|S|}\sum_{i=1}^{|S|} \min_{k=1}^{|R|} \sqrt{\sum_{j=1}^{n}(x_j'^i - x_j^k)^2} \right) \quad (8.12)$$

where $|R|$ and $|S|$ are the size of set, x_j^i is the jth element of solution $\boldsymbol{x}^i$ in R, and $x_j'^k$ is the jth element of solution $\boldsymbol{x}'^k$ in S. A conceptual diagram of α is shown in Fig. 8.10.

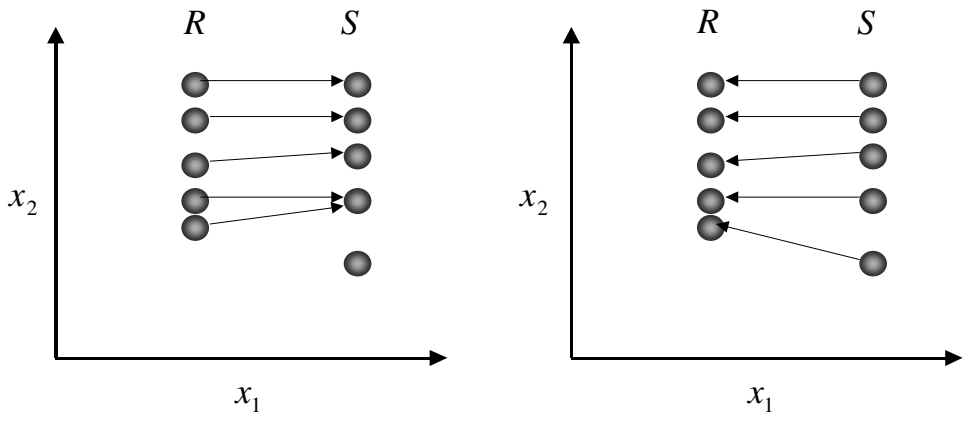


Figure 8.10: Conceptual diagram of similarity measure.

First, the scheduling list Z is generated by using a condition variable scheduling algorithm. Second, a test problem optimized by using only the random initialization method and the mean value of 30 trials of $\alpha(P_{\mathrm{Random}}(T, \boldsymbol{u}^i), X'(\boldsymbol{u}^i)), i = 1, \ldots, L$ is calculated as

$$\bar{\alpha}_{\mathrm{Random}}(\boldsymbol{u}^i) = \frac{1}{30}\sum_{j=1}^{30} \alpha(P^j_{\mathrm{Random}}(T, \boldsymbol{u}^i), X'(\boldsymbol{u}^i)).$$

Third, the test problem is optimized by using the interpolative initialization method and the mean value of 30 trials of $\alpha(P_{\mathrm{Interp}}(t, \boldsymbol{u}^i), X'(\boldsymbol{u}^i)), t = 0, \ldots, T, i = 1, \ldots, L$ is calculated as

$$\bar{\alpha}_{\mathrm{Interp}}(t, \boldsymbol{u}^i) = \frac{1}{30}\sum_{j=1}^{30} \alpha(P^j_{\mathrm{Interp}}(t, \boldsymbol{u}^i), X'(\boldsymbol{u}^i)).$$

In the scheduling list Z, all of the 30 trials are the same. Finally, the minimum generation $\tau(\boldsymbol{u})$ that satisfies $\bar{\alpha}_{\mathrm{Interp}}(t, \boldsymbol{u}) \leq \bar{\alpha}_{\mathrm{Random}}(\boldsymbol{u})$ and ratio

$$\hat{\alpha}(\boldsymbol{u}) = \frac{\bar{\alpha}_{\mathrm{Interp}}(T, \boldsymbol{u})}{\bar{\alpha}_{\mathrm{Random}}(\boldsymbol{u})}$$

are calculated.

8.5.3 Discussion of Results

Figures 8.11 and 8.12 are distribution of $\tau(\boldsymbol{u})$ and $\hat{\alpha}(\boldsymbol{u})$ in two-dimensional condition variable space. Furthermore, Figs. 8.13 and 8.14 are the sorted bar charts of $\tau(\boldsymbol{u})$ and $\hat{\alpha}(\boldsymbol{u})$, respectively. To clarify the effectiveness of proposed method such as the gray area in Fig. 8.6, set of pair $(\boldsymbol{u}, \tau(\boldsymbol{u}))$ was sorted in the descending order by $\tau(\boldsymbol{u})$ as the key, and a set of pair $(\boldsymbol{u}, \hat{\alpha}(\boldsymbol{u}))$ was sorted in the descending order by $\hat{\alpha}(\boldsymbol{u})$ as the key. These results are discussed as follows:

- First of all, when U are scheduled by Type A algorithm, four vectors $\boldsymbol{u} = [0\ 0]^{\mathrm{T}}, [0\ 1]^{\mathrm{T}}, [1\ 0]^{\mathrm{T}}, [1\ 1]^{\mathrm{T}}$ that form the convex hull of U are selected. Because the four MOPs defined by these vectors are initialized by the random initialization method, these achievement generations become 100. Similarly, If Type B algorithm is adopted, four MOPs corresponding $\boldsymbol{u} = [0\ 0]^{\mathrm{T}}, [0\ 0.1]^{\mathrm{T}}, [0.1\ 0]^{\mathrm{T}}, [0.1\ 0.1]^{\mathrm{T}}$ need 100 generations.

- In the results of TF-1 scheduled by Type A, the achievement generation of $[0.5\ 0.5]^{\mathrm{T}}$ becomes about 40, since the interpolative initialization method using the four vectors are applied. After that, the achievement generations of $[0.5\ 0]^{\mathrm{T}}, [1\ 0.5]^{\mathrm{T}}, [0.5\ 1]^{\mathrm{T}}$ become about 20. Finally, the achievement generations of others become 0. This result implies we can directly employ the interpolation initial populations $P(0, \boldsymbol{u})$ as $P(*, \boldsymbol{u})$ without optimization. The generation reduction ratio of the proposed method was about 96% in this case.

- In the results of TF-2 scheduled by Type B, the number of achievement generations became 40-90 at several conditions. Because the four conditions constructing convex hull have the same the Pareto optimal set, it is understood that the interpolative initialization method did not work well in the early stage of the schedule. For instance, the Pareto optimal set of $\boldsymbol{u} = [0.5\ 0.5]^{\mathrm{T}}$, that was evaluated next to four condition vectors, is located the positions furthest away from the four sets in the decision variable space. The generation reduction ratio of the proposed method was about 90% in this case.

- In the results of TF-1 and TF-2 scheduled Type B were almost same. It is understood that the results led to the continuousness of the condition variable space in each problem. The generation reduction ratios of the proposed method were about 93% in these cases.

- When comparing to the similarity measure of the final generation, it is understood that the interpolative initialization method is able to obtain a Pareto approximation set that is on average around 31–38% better than the random initialization method in Fig. 8.12.

Summary of condition variable scheduling is as follows:

- When Type A is used, the number of achievement generations can be reduced for simple problems such as TF-1. However, an effect sufficient enough might not be achieved for a problem such as TF-2.

- When Type B is used, a steady effect is achieved also in problems such as TF-2. However, extrapolation is always executed in this algorithm. Hence, it is difficult to use the extrapolated population of a condition vector far away from search history.

Therefore, the proposed method is able to improve the precision of Pareto approximation set under the same generation, as well as reduce the number of generations needed.

8.6 Summary

In this chapter, Parametric Multi-objective Optimization Problems (PMOPs) were proposed as a new problem class of MOPs with condition variables. Moreover, MOEAs for PMOPs were discussed. To solve the PMOPs efficiently, an interpolative initialization method was introduced. The proposed method interpolates plural Pareto approximation sets searched in the past and uses the interpolated population for an initial population of a MOP defined by new condition variable. As a result, it was understood that the proposed method has a possibility of evaluation cost reduction through numerical experimentation.

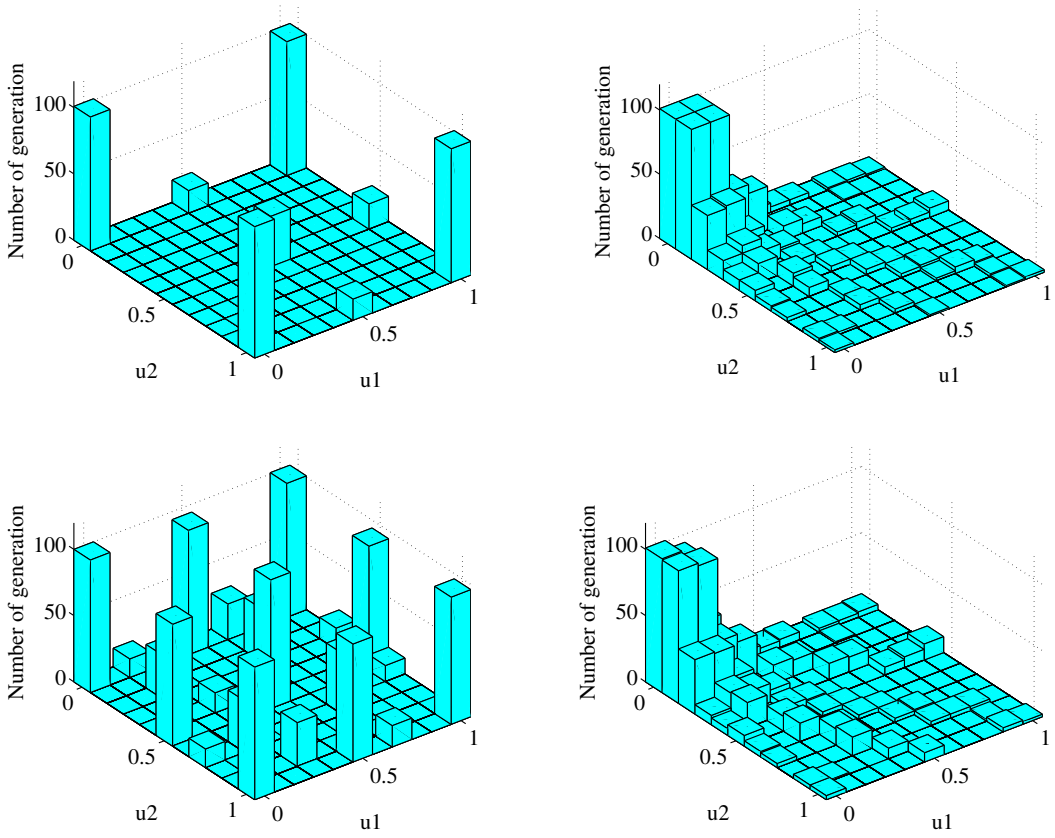


Figure 8.11: Distribution of the needed generation in condition variable space (upper left: TF-1, Type A, upper right: TF-1, Type B, lower left: TF-2, Type A, lower right: TF-2, Type B).

In the future, research on the following items will be conducted, and the practicality of the MOEAs for PMOPs will be improved.

- The test functions used in the numerical experiments were comparatively simple because of the need to confirm the basic behavior of the method. Therefore, it is necessary to evaluate problems whose interpolations are more difficult.
- In the numerical experiments, the similarity measure α was introduced as an evaluation measure to investigate the precision of Pareto approximation set in decision variable space. The relationship between α and other evaluation measures [18] will be examined.
- In this thesis, two scheduling algorithms called Type A and B were employed. However, because many scheduling algorithms can be considered, new scheduling algorithms will be developed and their effectiveness will be investigated.
- The proposed method will be adopted for the calibration of real engines to confirm

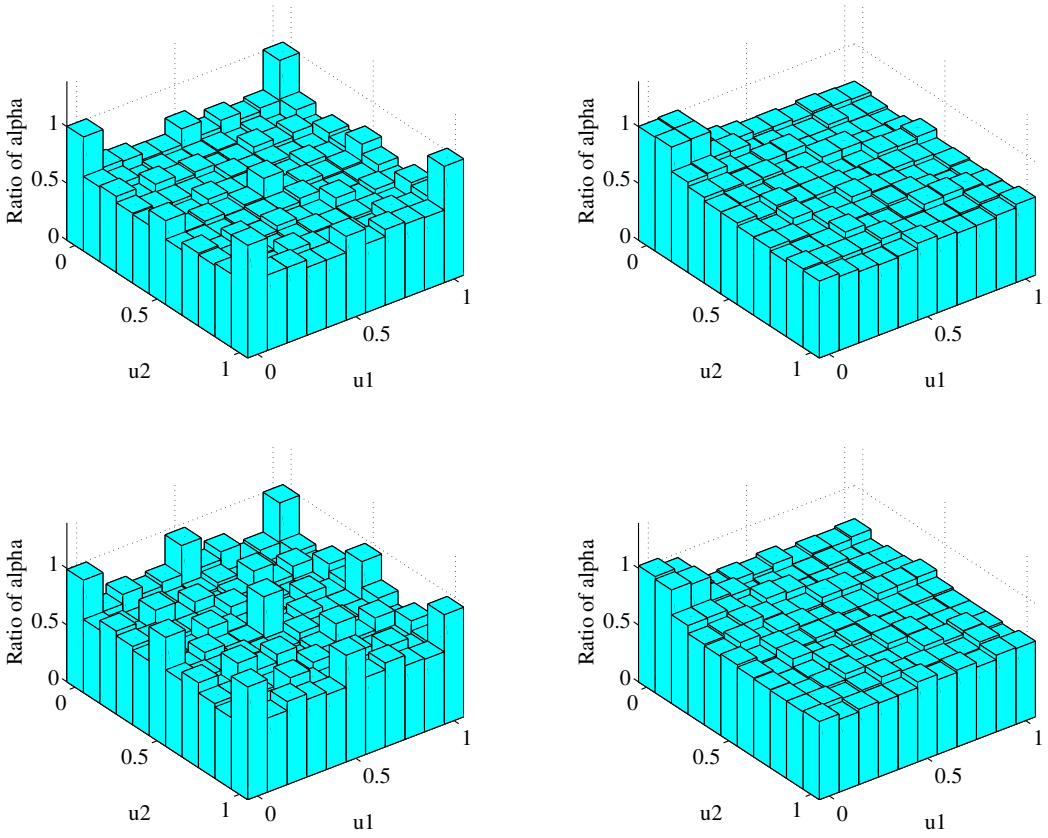


Figure 8.12: Distribution of the precision of solution in condition variable space (upper left: TF-1, Type A, upper right: TF-1, Type B, lower left: TF-2, Type A, lower right: TF-2, Type B).

its effectiveness.

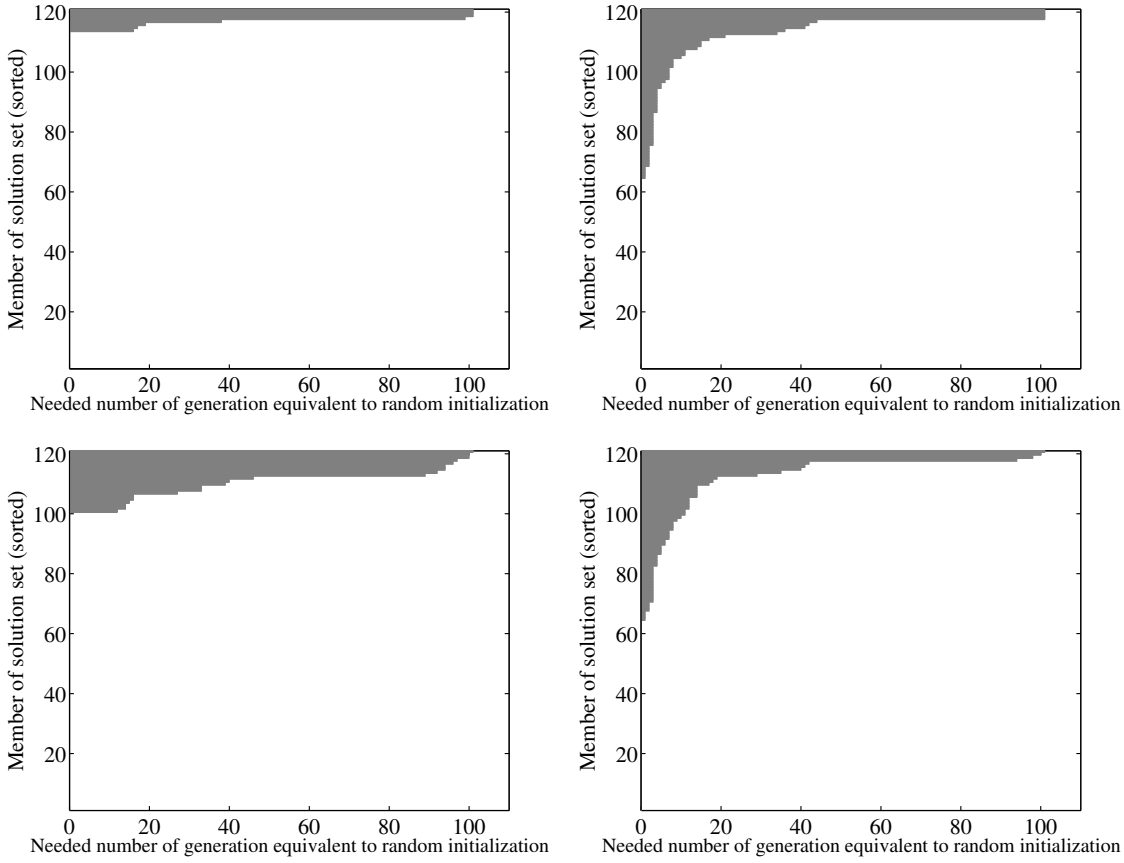


Figure 8.13: Comparison of the needed generation (upper left: TF-1, Type A, upper right: TF-1, Type B, lower left: TF-2, Type A, lower right: TF-2, Type B).

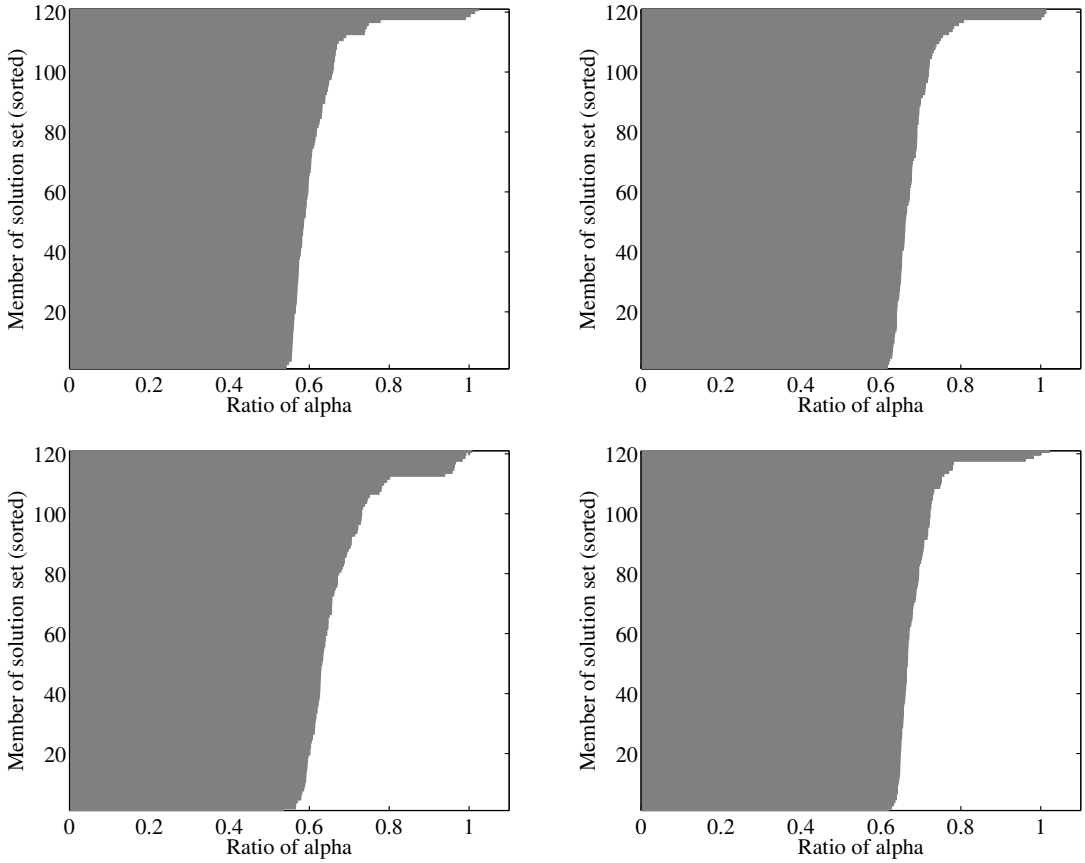


Figure 8.14: Comparison of the precision of solution (upper left: TF-1, Type A, upper right: TF-1, Type B, lower left: TF-2, Type A, lower right: TF-2, Type B).

Chapter 9

Conclusion

9.1 Summary

In this thesis, to cope with the drawbacks of conventional automatic engine calibration based on simulation-based optimization and Response Surface Methodology (RSM), Experiment-Based Evolutionary Multi-objective Optimization (EBEMO) have been studied as a novel approach. We are confident this is a promising calibration method against the background of the progress of engine Hardware In the Loop Simulation (HILS) environments.

In the following, the proposals in this thesis and the results described in aforementioned chapters are briefly summarized.

- In Chapter 4, we proposed Memory-based Fitness Estimation and Distribution Selection GA (MFE-DSGA) to handle the influence of observation noise. At first, we introduced the three problems of conventional MOEAs for noisy environment. These problems are caused by three features of MOEAs, i. e. , ranking algorithm, diversity maintenance, and elitism. Next, we proposed MFE-DSGA that has three features to overcome these problems. The first feature is fitness estimation method that is an extension of MFEGA proposed by Sano and Kita [91, 93] for stabilizing ranking algorithm. The second feature is distribution based selection method to maintain the diversity of the population. The third feature is the α-domination strategy proposed by Ikeda et al. [49] to eliminate 'fake' non-dominated solutions survived by elitism. The effectiveness of the proposed method was verified through numerical experiments and a real engine experiment.

- In Chapter 5, we proposed UNDX for Periodic function (UNDX-P) to overcome the periodicity of functions. At first, we introduced the difficulties of periodic function optimization by EAs. When we apply EAs for periodic function optimization, we face the sampling bias and the evolutionary stagnation. In order to solve these problems, we proposed the basic framework of a crossover on hy-

persphere called UNDX on Hypersphere (UNDX-H) and a crossover for periodic functions called UNDX-P as a special case of UNDX-H. Through investigation of statistical properties of UNDX-P, we found that UNDX-P has two suitable properties that roughly preserve statistics and change offspring distribution from bimodal to unimodal. Moreover, we demonstrated, through numerical experiments, that UNDX-P can stably optimize periodic functions so they do not depend on the domain of function. We also applied UNDX-P to non-periodic optimization problems because it does not have sampling bias in the early stage of the search. As a result, the search performance showed that it was comparatively robust for the optimum position.

- In Chapter 6, we proposed Individual Evaluation Scheduling (IES) to reduce the influence of transient response caused by parameter switching of dynamical systems. To cope with the influence of transient response, we proposed IES. IES is composed of Evaluation Order Scheduling (EOS) and Evaluation Time Scheduling (ETS). Through numerical experiment using a formal test problem and experiment using a HILS environment for real engines, it was shown that the proposed method was able to improve search accuracy and shorten search time of MOEAs for dynamical systems simultaneously.

- In Chapter 7, we discuss further Pre-selection to accelerate optimization by reducing the number of evaluation in experiment-based optimization. First, the performance of proposed method was examined through numerical experiments. As a result, it was confirmed that the Pre-selection was able to greatly reduce the number of evaluations greatly for four test functions regardless of the presence of the observation noise. Moreover, the search performance did not changed too much when the setting parameters of Pre-selection and UNDX were changed. These facts indicate that the proposed method has robustness for the setting parameters. Next, the Pre-selection was applied for EBEMO of real internal-combustion engines. To demonstrate that MOEAs can optimize real engines in practical time by applying Pre-selection, a real engine was optimized as a two-objective optimization problem. As a result, it was confirmed that NSGA-II with Pre-selection is able to reduce optimization time to about 13% as well as improve the search accuracy, and achieve the optimization time at a practical level.

- In Chapter 8, Parametric Multi-objective Optimization Problems (PMOPs) were formulated as a new class of MOPs. To solve PMOPs efficiently, an interpolative initialization method was introduced. This method interpolates plural Pareto approximation sets searched in the past and uses the interpolated population for an initial population of a MOP defined by new condition variable. The interpolative initialization method is based on optimal matching and linear regression. The proposed framework is composed by four modules, i. e. , condition variable

scheduler, initialization operator, an MOEA, and search history. The initialization operator selects between random and interpolative method depending on the number of condition variable vector stored in the search history. In the numerical experiments, two simple test functions were used to confirm effectiveness. In addition, two types of condition variable scheduling algorithms were adopted. As a result, it was understood that the proposed method has the potential to reduce evaluation costs.

9.2 Contribution for Engine Calibration

In this thesis, EBEMO is provided as another way to approach automatic engine calibration with HILS environment, instead of Response Surface Methodology (RSM) that is currently employed as the optimization method. In particular, it is consider that EBEMO is effective in the early stages of engine calibration. Because the engine specifications are often modified, the control parameters have to be optimized each time. The calibration can be done immediately by using EBEMO after the modifications, whereas the statistical models must be re-constructed if RSM is adopted. Hence, EBEMO will contribute to efficient engine calibrations, because some processes such as design of experiments and model selection that are necessary in RSM can be omitted.

EBEMO is developed for automatic engine calibration, it is found that the proposed methodology is useful for prototype engine evaluations. Currently, the proposed methods are integrated and used as a performance analysis tool for visualization of trade-offs and control parameter correlation. In fact, calibration engineers have commented that EBEMO is useful for the investigation of the engine specifications.

In implementation of EBEMO for engine calibration, we recommend following structure:

Basic Structure: To optimize at an operating condition efficiently, application of IES and Pre-selection to MOEAs such as NSGA-II are recommended. Because Pre-selection for EBEMO inherits fitness estimation, sampling in sparse area, and α-domination strategy from MFE-DSGA, it can optimize well in noisy environments. Moreover, IES can reduce the influence of transient response. The both of the algorithms do not interfere in each other.

In the Pre-selection algorithm introduced in Chapter 7, IES is used for the initial population $P(0)$ and the evaluated offspring population $Q(t)$ before the evaluation of the real environment.

Periodic Control Parameter: If the problem has periodic control parameter such as injection timing, UNDX-P should be adopted as the crossover operator.

Map Construction: In the engine calibration for map-based controller, the problem we should solve is a PMOP. Therefore, the framework of MOEA for PMOP

proposed in Chapter 8 should be employed. In the framework, the basic structure can be used as a MOEA, because the Interpolative Initialization Method requires only final population to generate initial populations.

Figure 9.1 shows the block diagram of EBEMO for engine calibration recommended above. MFE-DSGA, IES, Pre-selection and Interpolative Initialization Method use

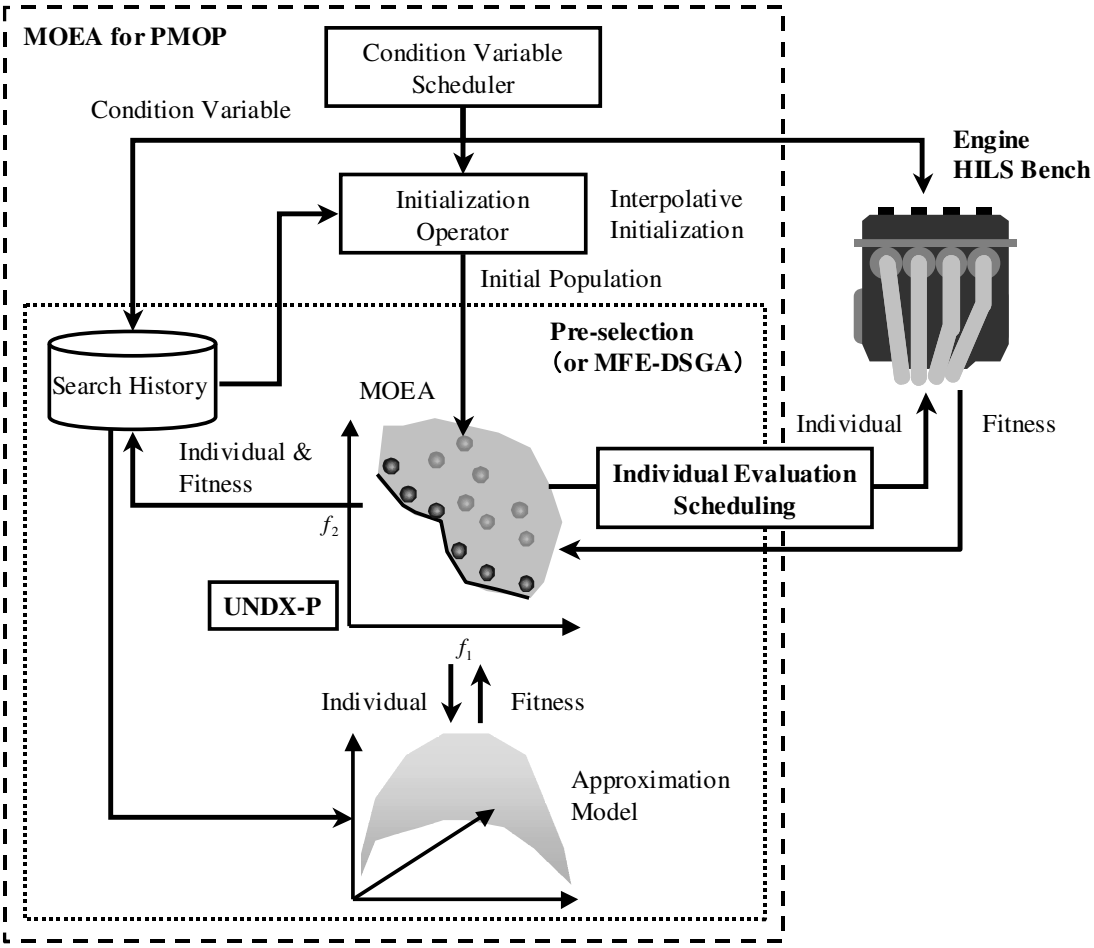


Figure 9.1: Block diagram of EBEMO for engine calibration.

Euclidean distance in decision variable and objective function spaces. Therefore, adequate normalization is important.

9.3 Future Work

In this thesis, a basic EBEMO framework for automotive engine calibration has been constructed. However, several topics exist to construct a more effective framework. Furthermore, expansion into other applications can be considered.

Constraint Handling: In this thesis, unconstrained multi-objective optimization problems were dealt with. However, lots of real world applications usually have several constraints. Hence, EBEMO with constraint handling techniques should be studied to extend the application area.

Multi Criterion Decision Making: Selection of preference solution from the Pareto frontier is the most important issue of post processing. For example, EBEMO can give plural Pareto optimal sets corresponding map grids. Hence, to integrate these solutions to maps, we have to select preference solutions one by one. This decision making requires considerable effort. To lighten the burden of the decision making, a support tool based on Multi Criterion Decision Making (MCDM) techniques [13] should be developed.

Extension to Other Applications: In this thesis, engine calibration focused on a specific engineering application. However, EBEMO framework is promising approach for other real world applications because each proposed technique has versatility.

For example, UNDX-P can be used as an universal crossover operator without sampling bias for many applications, regardless of periodicity. IES can improve the quality of optimization and protect optimized objects from extreme parameter switching of dynamical systems. Various mechanical and electrical systems such as hydraulic actuator and electric motor are suitable engineering applications. Pre-selection can be applied to noiseless environments with expensive evaluation cost such as CDF simulations, since we indicated the effectiveness of proposed method under noiseless environment. Moreover, Agent-Based Simulation (ABS) as a complex simulation with randomness [51] and Interactive Evolutionary Computation (IEC) as a 'Human In the Loop Simulation' are interesting applications. Furthermore, the idea of interpolative initialization method for PMOPs can readily be extended to Dynamic Multi-objective Optimization Problems (DMOPs), because the time parameter t of DMOPs can be considered as a kind of condition variable.

Appendix A

Real-Coded Genetic Algorithms

A.1 Unimodal Normal Distribution Crossover

Unimodal Normal Distribution Crossover (UNDX) proposed by Ono et al. [80] is described as follows:

1. Select three parents $\boldsymbol{x}^1$, $\boldsymbol{x}^2$, and $\boldsymbol{x}^3$.
2. Calculate $\boldsymbol{g} = (\boldsymbol{x}^1 + \boldsymbol{x}^2)/2$.
3. Calculate $\boldsymbol{d} = \boldsymbol{x}^1 - \boldsymbol{x}^2$.
4. Calculate the distance D from $\boldsymbol{x}^3$ to main search axis that is the line through $\boldsymbol{x}^1$ to $\boldsymbol{x}^2$.
5. Generate offspring by following equation.

$$\boldsymbol{x}^{\mathrm{c}} = \boldsymbol{g} + \xi \boldsymbol{d} + D \sum_{i=1}^{n-1} \eta_i \boldsymbol{e}_i,$$
$$\xi \sim N(0, \sigma_\xi^2),\ \eta_i \sim N(0, \sigma_\eta^2),$$

   where n is the number of dimension, $N(0, \sigma^2)$ is normal distribution random number defined by mean 0 and variance σ^2, and $\boldsymbol{e}_i$ is the normal orthogonal base vector of the subspace orthogonaled to the main search axis. $\sigma_\xi = 0.5$ and $\sigma_\eta = 0.35/\sqrt{n}$ are recommended in [80].

Figure A.1 shows schematic diagram of UNDX.

A.2 Minimal Generation Gap

Minimal Generation Gap (MGG) proposed by Sato et al. [95], which is a generation alternation model, is described as follows:

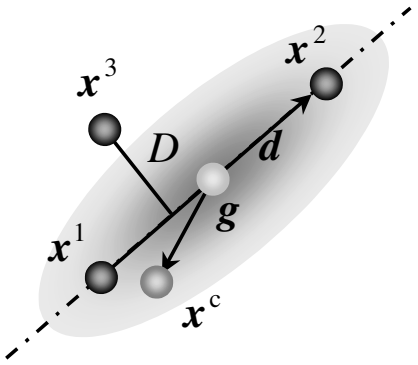


Figure A.1: Unimodal Normal Distribution Crossover (UNDX).

1. Randomly generate an initial population.
2. Randomly select two parents from the population, and generate k offspring by crossover operator.
3. Select the best individual and randomly select an individual from family that is constructed by parents and k offspring.
4. Replace the couple of selected individuals to parents in the population.

Conceptual diagram of MGG is shown in Fig. A.2.

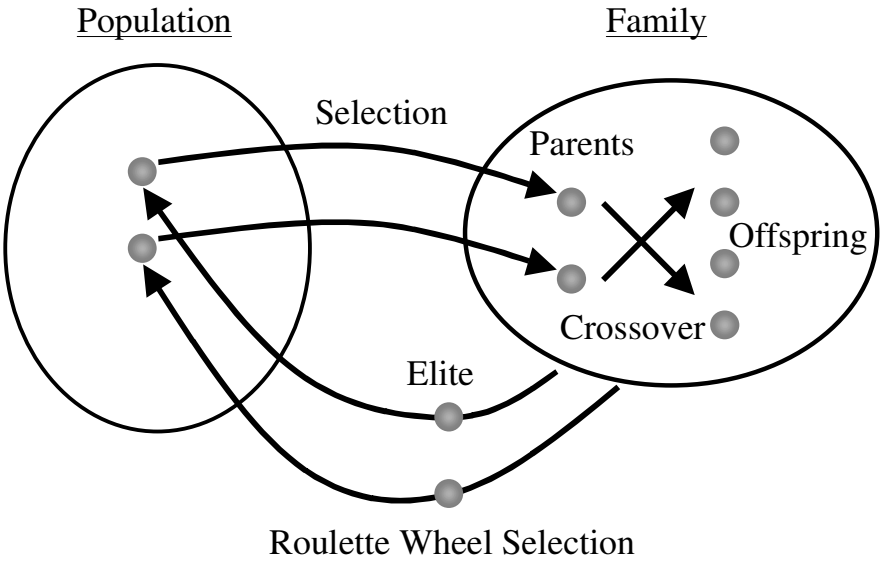


Figure A.2: Minimal Generation Gap (MGG).

Appendix B

NSGA-II

The non-dominated sorting, the crowding distance, and the crowded tournament selection used in NSGA-II proposed by Deb et al. [19] are introduced in the following.

B.1 Non-dominated Sorting

The concept of the ranking method used in non-dominated sorting was originally introduced in Goldberg' textbook [37]. Non-dominated sorting algorithm is described as follows:

1. Set rank $r := 1$.
2. Find the non-dominated set P_{nd} of population P.
3. Update $P_r = P_{\text{nd}}$ and $P := P \backslash P_{\text{nd}}$.
4. If $P \neq \emptyset$, set $r := r + 1$ and return to Step 2); otherwise, stop and declare all sets P_i, for $i = 1, 2, \ldots, r$.

Figure B.1 depict the rank number of population by non-dominated sorting.

B.2 Crowding Distance

Crowding distance of the set P_r is calculated as follows:

1. Call the number of solutions in P_r as $l = |P_r|$, and assign $d_j = 0$ for $j = 1, 2, \ldots, l$.
2. Sort P_r in worse order of f_i for each objective function $i = 1, 2, \ldots, m$, and find the sorted indices vector I^i.

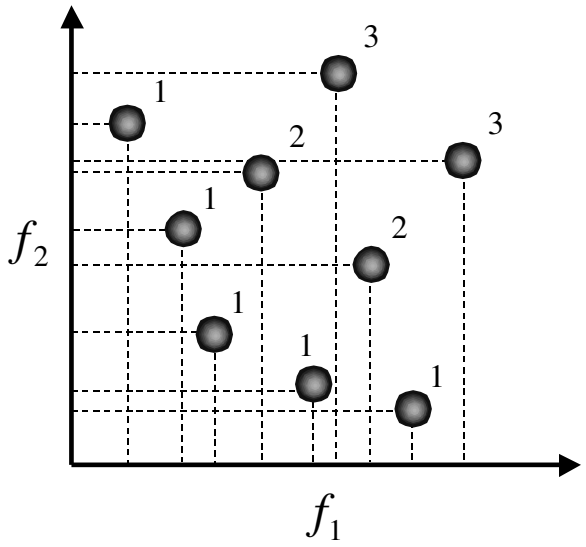


Figure B.1: Rank based on non-dominated sorting.

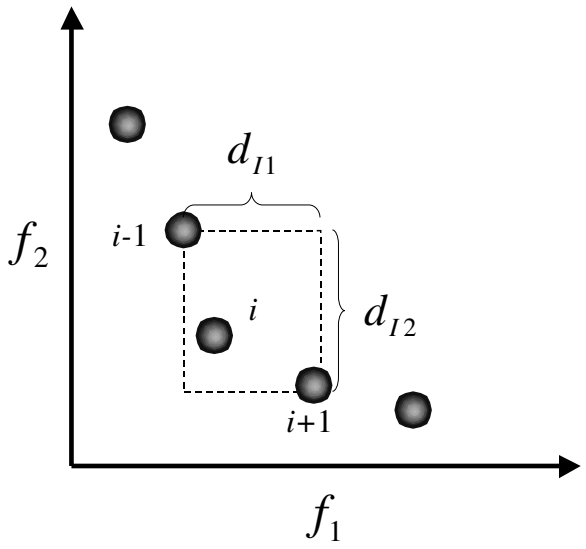


Figure B.2: Calculation of crowding distance.

3. For $i = 1, 2, \ldots, m$, assign a large distance to the boundary solutions, or $d_{I_1^i} = d_{I_l^i} = \infty$, and for all other solutions $j = 2$ to $l - 1$, assign:

$$d_{I_j^i} := d_{I_j^i} + \frac{f_i^{(I_{j+1}^i)} - f_i^{(I_{j-1}^i)}}{f_i^{\max} - f_i^{\min}},$$

where I_j^i denotes the solution index of the j th member in the sorted indices vector of i th objective function.

Conceptual diagram of crowding distance is shown in Fig. B.2.

B.3 Crowded Tournament Selection

A solution $\boldsymbol{x}^1$ wins a tournament with another solution $\boldsymbol{x}^j$ if any of the following conditions are true:

- If solution $\boldsymbol{x}^1$ has a better rank $r^1 < r^2$.
- If they have the same rank $r^1 = r^2$ but solution $\boldsymbol{x}^1$ has a better crowding distance $d^1 > d^2$.

Bibliography

[1] J. Alonso, F. Alvarruiz, J. Desantes, L. Hernández, V. Hernández, and G. Moltó "Combining Neural Networks and Genetic Algorithms to Predict and Reduce Diesel Engine Emissions," *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 1, pp. 46–55, 2007.

[2] C. Atkeson, A. Moore, and S. Schaal, "Locally Weighted Learning," *Artificial Intelligence Review*, Vol. 11, pp. 11–73, 1997.

[3] T. Bäck: *Evolutionary Algorithms in Theory and Practice*, Oxford, 1996.

[4] M. Berg, M. Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry Algorithms and Applications*, Springer, 1997.

[5] G. Böker, O. Magnor, and M. Schultalbers, "A Rapid Calibration Tool for Engine Control Software and its Application to Misfire Diagnosis Functions," *Proc. of CACSD*, pp. 2338–2342, 2006.

[6] C. Bohn, P. Stöber, and O. Magnor, "An Optimization-Based Approach for the Calibration of Lookup Tables in Electronic Engine Control," *Proc. of CACSD*, pp. 2315–2320, 2006.

[7] J. Branke, "Evolutionary Approaches to Dynamic Optimization Probrems – Update Survey –," *Proc. of GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*, pp. 27–30, 2001.

[8] J. Branke, C. Schmidt, and H. Schmeck, "Efficient Fitness Estimation in Noisy Environments," *Proc. of GECCO 2001*, pp. 243–250, 2001.

[9] J. Branke, and C. Schmidt, "Faster Convergence by means of Fitness Estimation," *Soft Computing*, Vol. 9, No. 1, pp. 13–20, 2005.

[10] D. Büche, P. Stoll, R. Dornberger and P. Koumout-sakos, "Multiobjective Evolutionary Algorithm for the Optimization of Noisy Combustion Processes," *IEEE Transactions on System, Man and Cybernetics*, Vol. 32, No. 4, pp. 460–473, 2002.

[11] D. Büche, N. Schraudolph, and P. Koumutsakos, "Accelerating Evolutionary Algorithms with Gaussian Process Fitness Function Models," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 35, No. 2, pp. 183–194, 2005.

[12] L. Bui, J. Branke, and H. Abbass, "Multiobjective Optimization for Dynamic Environments," *Proc. of CEC 2005*, pp. 2349–2356, 2005.

[13] V. Chankong, and Y. Haimes, *Multiobjective Decision Making Theory and Methodology*, Dover, 1983.

[14] C. Coello Coello, G. Lamont, D. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems 2nd Edition*, Springer, 2007.

[15] J. Cook, J. Sun, J. Buckland, I. Kolmanovsky, H. Peng, and J. Grizzle, "Automotive Powertrain Control: A Survey," *Asian Journal of Control*, Vol. 8, No. 3, pp. 237–260, 2005.

[16] D. Corne, J. Knowles, and M. Oates, "The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization," *Proc. of PPSN VI*, pp. 839–848, 2000.

[17] H. Chung, and J. Alonso, "Multiobjective Optimization using Approximation Model-Based Genetic Algorithms," *10th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, AIAA 2004–4325, 2004.

[18] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, 2001.

[19] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, pp. 182–197, 2002.

[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multi-Objective Optimization," *Proc. of CEC 2002*, pp. 825–830, 2002.

[21] K. Deb, and H. Gupta, "Searching for Robust Pareto-Optimal Solutions in Multi-objective Optimization," *Proc. of EMO 2005*, LNCS 3410, pp. 150–164, Springer, 2005.

[22] K. Deb, P. Zope, and A. Jain, "Distributed Computing of Pareto-Optimal Solutions using Multi-Objective Evolutionary Algorithms," KanGAL Report Number 2002008, 2002.

[23] T. Dvorak, L. Malone, and R. Hoekstra, "Statistical Process Control and Design of Experiment Process Improvement Methods for the Powertrain Laboratory," SAE Technical Paper, 2003–01–3208, 2003.

[24] T. Dvorak, R. Rohrer, P. Lamb, R. Hoekstra, and R. Meyer, "Non-Constant Variance – Emission MOdeling Methods for Offline Optimization and Calibration of Engine Management Systems," SAE Technical Paper, 2003–32–0010, 2003.

[25] T. Dvorak, R. Hoekstra, and J. Pet-Armacost, "Improving Exhaust Header Performance with Multiple Response Surface Methods," SAE Technical Paper, 2003–01–1389, 2003.

[26] M. El-Beltagy, and A. Keane, "Evolutionary Optimization for Computationally Expensive Problems using Gaussian Process," *Proc. of International Conference on Articfical Intelligence (ICAI 2001)*, pp. 708–714, 2001.

[27] M. Emmerich, A. Giotis, M. Özdenir, T. Bäck, and K.Giannakoglou, "Metamodel-Assisted Evolution Strategies," *Proc. of PPSN VII*, LNCS 2439, pp. 371–380, Springer, 2002.

[28] M. Emmerich, and B. Naujoks, "Metamodel Assisted Multiobjective Optimisation Strategies and their Application in Airfoil Design," *Proc. of ACDM VI*, pp. 249–260, 2004.

[29] M. Emmerich, and B. Naujoks, "Metamodel-Assisted Multiobjective Optimization with Implicit Constraints and Its Application in Airfoil Design," *Proc. of Design Optimization International Conference 2004*, pp. –, 2004.

[30] L. Eshelman, and J. Schaffer, "Real-Coded Genetic Algorithms and Interval-Schemata," *Foundations of Genetic Algorithms 2*, pp. 187–202, 1993.

[31] L. Eshelman, K. Mathias, and J. Schaffer, "Crossover Operator Biases : Exploiting the Population Distribution," *Proc. of 7th ICGA*, pp. 354–361, 1997.

[32] M. Farina, K. Deb, and P. Amato, "Dynamic Multiobjective Optimization Problems: Test Cases, Approximation, and Applications," *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 5, pp. 425–442, 2004.

[33] P. J. Fleming, and R. Purshouse, "Evolutionary Algorithms in Control Systems Engineering: a Survey," *Control Engineering Practice*, Vol. 10, No. 11, pp. 1223–1241, 2002.

[34] C. Fonseca, and P. J. Fleming, "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization," *Proc. of 5th ICGA*, pp. 416–423, 1993.

[35] A. Giotis, M. Emmerich, B. Naujoks, K. Giannakoglou, and T. B ack, "Low-Cost Stochastic Optimization For Engineering Applications," *Proc. of Evolutionary Methods for Design, Optimization and Control*, 2002.

[36] C. Goh, and K .Tan, "An Investigation on Noisy Environments in Evolutionary Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 11, No. 3, pp. 354–381, 2007.

[37] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989

[38] L. Gräning, Y. Jin, and B. Sendhoff, "Efficient Evolutionary Optimization using Individual-Based Evolution Control and Neural Networks: A Comparative Study," *Proc. of ESANN 2005*, pp. 273–278, 2005.

[39] L. Guzzella, and C. Onder, *Introduction to Modeling and Control of Internal Combustion Engine Systems*, Springer, 2004.

[40] L. Guzzella, and C. Onder, "Past, Present and Future of Automotive Control," *Control of Uncertain Systems*, LNCIS 329, pp. 163–182, Springer, 2006.

[41] M. Hafner, and R. Isermann, "Multiobjective Optimization of Feedforward Control Maps in Engine Management Systems towards Low Consumption and Low Emissions," *Transactions of the Institute of Measurement and Control*, Vol. 25, No. 1, pp. 57–74, 2003.

[42] K. Harada, J. Sakuma, and S. Kobayashi, "Local Search for Multiobjective Function Optimization: Pareto Descent Method," *Proc. of GECCO 2006*, pp. 659–666, 2006.

[43] T. Higuchi, S. Tsutsui, and M. Yamamura, "Theoretical Analysis of Simplex Crossover for Real-Coded Genetic Algorithms," *Proc. of PPSN VI*, LNCS 1917, pp. 365–374, Springer, 2000.

[44] T. Hiroyasu, M. Miki, and S. Watanabe: "Divided Range Genetic Algorithms in Multiobjective Optimization Problems", *Proc. of IWES 99*, pp. 57–65, 1999.

[45] T. Hiroyasu, M. Miki, J. Kamiura, S. Watanabe, and H. Hiroyasu, "Multi-Objective Optimization of Diesel Engine Emissions and Fuel Economy using Genetic Algorithms and Phenomenological Model," SAE Technical Paper, 2001–01–2778, 2002.

[46] A. Hoerl and R. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics* Vol. 12, pp. 55–67, 1970.

[47] J. H. Holland, *Adaptation in natural and artificial systems*, The University of Michigan Press, 1975.

[48] E. J. Hughes, "Evolutionary Multi-Objective Ranking with Uncertainty and Noise," *1st International Conference on Evolutionary Multi-Criterion Optimization 2001*, LNCS 1993, pp. 329–343, Springer, 2001.

[49] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-Based MOEAs, Does Non-Dominated Really Mean Near to Optimal?," *Proceedings of 2001 IEEE Congress on Evolutionary Computation*, pp. 957–962, 2001.

[50] K. Ikeda, *Global Optimization using Genetic Algorithms for Practical Uses*, Ph. D Thesis, Tokyo Institute of Technology, 2002. (in Japanese)

[51] K. Ikeda, H. Suzuki, S. Markon, and H. Kita, "Designing Traffic-Sensitive Controllers for Multi-Car Elevetors Through Evolutionary Multi-objective Optimization," *4th Int'l Conf. on EMO 2007*, LNCS 4403, pp. 673–686, Springer, 2007.

[52] S. Jeong, K. Yamamoto, and S. Obayashi, *Kriging-Based Probabilistic Method for Constrained Multi-Objective Optimization Problem*, AIAA Paper 2004–6437, 2004.

[53] Y. Jin, M. Olhofer, and B. Sendhoff, "On Evolutionary Optimization with Approximate Fitness Functions," *Proc. of GECCO 2000*, pp. 786–793, 2000.

[54] Y. Jin, M. Olhofer, and B. Sendhoff, "Managing Approximate Models in Evolutionary Aerodynamic Design Optimization," *Proc. of CEC 2001*, pp. 592–599, 2001.

[55] Y. Jin, M. Olhofer, and B. Sendhoff, "A Framework for Evolutionary Optimization with Approximate Fitness Function," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 5, pp. 481–494, 2002.

[56] Y. Jin, and B. Sendhoff, "Reducing Fitness Evaluations using Clustering Techniques and Neural Network Ensembles," *Proc. of GECCO 2004*, pp. 688-699, 2004.

[57] Y. Jin, and B. Sendohoff, "Constructing Dynamic Test Problems using the Multiobjective Optimization Concept," *Applications of Evolutionary Computing*, LNCS 3005, pp. 525–536, Springer, 2004.

[58] Y. Jin, and J. Branke, "Evolutionary Optimization in Uncertain Environments – a Survey," *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 3, pp. 303–317, 2005.

[59] Y. Jin, "Comprehensive Survey of Fitness Approximation in Evolutionary Computation," *Soft Computing*, Vol. 9, pp. 3–12, 2005.

[60] Y. Jin (Ed.), *Mult-Objective Machine Learning*, Springer, 2006.

[61] I. Kamihira, M. Yamaguchi, and H. Kita, "On-line Adaptation of Vehicles by Means of an Evolutionary Control System," *Proc. of SMC 1999*, pp. 553–558, 1999.

[62] M. Kanazaki, M. Morikawa, S. Obayashi, and K. Nakahashi, "Multiobjective Design Optimization of Merging Configuration for an Exhaust Manifold of a Car Engine," *Proc. of PPSN VII*, LNCS 2439, pp. 281–287, Springer, 2002.

[63] U. Kiencke, and L. Nielsen, *Automotive Control Systems For Engine, Driveline, and Vehicle*, Springer, 2000.

[64] M. Kim, M. Liechty, and R. Reitz, "Application of Micro-Genetic Algorithms for the Optimization of Injection Strategies in a Heavy-Duty Diesel Engine," SAE Technical Paper, 2005–01–0219, 2005.

[65] M. Kim, and C. Lee, "An Experimental Study on the Optimization of Controller Gains for an Electro-Hydraulic Servo System using Evolution Strategies," *Control Engineering Practice*, Vol. 14, pp. 137–147, 2006.

[66] H. Kita, I. Ono, S. Kobayashi, "Theoretical Analysis of the Unimodal Normal Distribution Crossover for Real-coded Genetic Algorithms," *Proc. of CEC 1998*, pp. 529-534, 1998.

[67] H. Kita, I. Ono, and S. Kobayashi, "Multi-parental Extension of the Unimodal Normal Distribution Crossover for Real-Coded Genetic Algorithms," *Proc. of CEC 1999*, pp. 1581–1587, 1999.

[68] H. Kita, and M. Yamamura, "A Functional Specialization Hypothesis for Designing Genetic Algorithms," *Proc. of SMC 1999*, p. 250, 1999.

[69] J. Knowles, and E. Hughes, "Multiobjective Optimization on a Budget of 250 Evaluations," *Proc. of EMO 2005*, LNCS 3410, pp. 176–190, Springer, 2005.

[70] J. Knowles, "ParEGO: A Hybrid Algorithm with On-line Landscape Approximation for Expensive Multiobjective Optimization Problems," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 1, pp. 50–66, 2006.

[71] M. Laumanns, and N. Laumanns, "Evolutionary Multiobjective Design in Automotive Development," *Applied Intelligence*, Vol. 23, pp. 55–70, 2005.

[72] K. Liang, X. Yao, and C. Newton, "A Preliminary Study Into Evolutionary Search of an Approximated N-Dimensional Landscape," *Proc. of the Third Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, pp. 201–208, 1999.

[73] J. Mehnen, T. Wagner, and G. Rudolph, "Evolutionary Optimization of Dynamic Multi-objective Test Functions," *Proc. of Second Italian Workshop on Evolutionary Computation (GSICE2)*, 2006.

[74] R. Myers, and D. Montgomery, *Response Surface Methodology 2nd edition*, John Wiley & Sons, 2002.

[75] P. Nain, and K. Deb, "A Multi-Objective Optimization Procedure with Successive Approximate Models," KanGAL Report Number 2005002, 2005.

[76] H. Nakayama, M. Arakawa, and R. Sasaki, "A Computational Intelligence Approach to Optimization with Unknown Objective Functions," *Proc. of ICANN 2001*, LNCS 2130, pp. 73–80, 2001.

[77] F. Neri, G. Cascella, N. Salvatore, and S. Stasi, "An Adaptive Prudent-Daring Evolutionary Algorithm for Noise Handling in On-line PMSM Drive Design," *Proc. of CEC 2007*, pp. 584–591, 2007.

[78] A. Nessler, C. Haukap, and K. Röpke, "Global Evaluation of the Drivability of Calibrated Diesel Engine Maps," *Proc. of CACSD*, pp. 2683–2688, 2006.

[79] Y. Ong, P. Nair, A. Keane, K. Wong, "Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems," *Knowledge Incorporation in Evolutionary Computation*, Studies in Fuzziness and Soft Computing, pp. 307–332, Springer, 2004.

[80] I. Ono, and S. Kobayashi, "A Real-coded Genetic Algorithm for Function Optimization using Unimodal Normal Distribution Crossover," *Proc. of 7th ICGA*, pp. 246–253, 1997.

[81] I. Ono, S. Kobayashi, and K. Yoshida, "Global and Multi-objective Optimization for Lens Design by Real-coded Genetic Algorithms," *Proc. of International Optical Design Conference*, pp. 110–121, 1998.

[82] R. Omran, R. Younes, J. Champoussin, D. Fedeli, F. Masson, and N. Guerrassi, "Genetic Algorithm for Dynamic Calibration of Engine's Actuators," SAE Technical Paper, 2007–01–1079, 2007.

[83] W. Ponweiser, and M. Vincze, "The Multiple Multi Objective Problem – Definition, Solution and Evaluation," *4th Int'l Conf. on EMO 2007*, LNCS 4403, pp. 877–892, Springer, 2007.

[84] T. Ray, and W. Smith, "A surrogate Assisted Parallel Multiobjective Evolutionary Algorithm for Robust Engineering Design," *Engineering Optimization*, Vol. 38, No. 8, pp. 997–1011, 2006.

[85] A. Ratle, "Accelerating the Convergence of Evolutionary Algorithms by Fitness Landscape Approximation," *Proc. of PPSN V*, LNCS1498, pp. 87–69, Springer, 1998.

[86] A. Ratle, "Optimal Sampling Strategies for Learning a fitness Model," *Proc. of CEC 1999*, pp. 2078–2085, 1999.

[87] I. Rechenberg, "Evolution Strategy," *Computational Intelligence Imitating Life*, pp. 147–159, IEEE Press, 1995.

[88] K. Röpke (Ed.), *Design of Experiments (DoE) in Engine Development II*, Expert Verlag, 2005.

[89] K. Röpke (Ed.), *Design of Experiments (DoE) in Engine Development III*, Expert Verlag, 2007.

[90] J. Sacks, W. Welch, T. Michell, and H. Wynn, "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, pp. 409–435, 1989.

[91] Y. Sano, and H. Kita, "Optimization of Noisy Fitness Functions by means of Genetic Algorithms using History of Search," *Proc. of PPSN VI*, LNCS1917, pp. 571–580, Springer, 2000.

[92] Y. Sano, H. Kita, I. Kamihira, and M. Yamaguchi, "Online Optimization of an Engine Controller by means of a Genetic Algorithm using History of Search," *Proc. of SEAL 2000*, pp. 2929–2934, 2000.

[93] Y. Sano, and H. Kita, "Optimization of Noisy Fitness Functions by Means of Genetic Algorithms using History of Search with Test of Estimation," *Proc. of CEC 2002*, pp. 360–365, 2002.

[94] Y. Sano, H. Kita, H. Kaji, and M. Yamaguchi, "Optimization of Dynamic Fitness Function By Means Of Genetic Algorithm using Sub-Populations," *Proc. of SEAL 2002*, 2002.

[95] H. Satoh, M. Yamamura, and S .Kobayashi, "Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation," *Proc. of IIZUKA*, pp. 494–497, 1996.

[96] Y. Sawaragi, H. Nakayama, and T. Tanino, *Theory of Multi-Objective Optimization*, Academic Press, 1985.

[97] H. -P. Schwefel, *Evolution and Optimum Seeking*, John Wiley & Sons, 1995.

[98] K. Shi, T. Chan, Y. Wong, S. Ho, "Speed Estimation of an Induction Motor Drive using an Optimized Extended Kalman Filter," *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 1, pp. 124–133, 2002.

[99] A. Singh, B. Minsker, and D. E. Goldberg, "Combining Reliability and Pareto Optimality - An Approach using Stochastic Multi-Objective Genetic Algorithms," *ASCE EWRI World Water and Environmental Resources Congress 2003 and Related Symposia*, 2003.

[100] A. Singh, *Uncertainty Based Multi-Objective Optimization of Groundwater Remediation Design*, M. S. Thesis, University of Illinois, 2003

[101] H. Someya, and M. Yamamura, "Robust Evolutionary Algorithms with Toroidal Search Space Conversion for Function Optimization," *Proc. of GECCO 2002*, pp. 553–560, 2002.

[102] H. Someya, *A Genetic Algorithm for Optimization Considering Functional Specialization of Genetic Operators*, Ph. D Thesis, Tokyo Institute of Technology, 2001. (in Japanese)

[103] S. Srinivasan, and F. Tanner, "An Investigation of the Gradient Determination Strategies for the Optimization of Diesel Engines," *Proc. of 16th International Multidimensional Engine Modeling Meeting at the SAE Congress*, 2006.

[104] O. Takahashi, S. Kimura, and S. Kobayashi, "An Adaptive Neighboring Search using Crossover-like Mutation for Deceptive Multimodal Function Optimization," *Trans. of JSAI*, Vol. 16, No. 2, pp. 175–184, 2001. (in Japanese)

[105] O. Tomobe, I. Ono, and S. Kobayashi, "Experimental Study on Determination of Protein Three Dimensional Structure using Genetic Algorithm," *Proc. of 25th SICE Symposium on Intelligent Systems*, pp. 35–40, 1998. (in Japanese)

[106] S. Tsutsui, "Multi-Parent Recombination in Genetic Algorithms with Search Space Boundary Extension by Mirroring," *Proc. of PPSN V*, LNCS1498, pp. 428–437, Springer, 1998.

[107] H. Ulmer, F. Streichert, and A. Zell, "Evolution Strategies with Controlled Model Assistance," *Proc. of CEC 2004*, pp. 1569–1576, 2004.

[108] D. Van Veldhuizen, J. Zydallis, G. Lamont, "Considerations in Engineering Parallel Multiobjective Evolutionary Algorithms," *IEEE Transactions of Evolutionary Computation*, Vol. 7, No. 2, pp. 144–173, 2003.

[109] K. Watanabe, and M. Tümer, "An Automotive Engine Calibration System using Microcomputer," *IEEE Transactions on Vehicular Technology*, Vol. 33, No. 2, pp. 45–50, 1984.

[110] K. Won, and T. Ray, "Performance of Kriging and Cokriging based Surrogate Models within the Unified Framework for Surrogate Assisted Optimization," *Proc. of CEC 2004*, pp. 1577–1585, 2004.

[111] S. Yang, Y. Ong, and Y. Jin (Eds.), *Evolutionary Computation in Dynamic and Uncertain Environments*, Springer, 2007.

[112] A. Zhou, Y. Jin, O. Zhang, B. Sendhoff, and E. Tsang, "Prediction-Based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization," *4th Int'l Conf. on EMO 2007*, LNCS 4403, pp. 832–846, Springer, 2007.

[113] Z. Zhou, Y. Ong, and P. Nair, "Hierarchial Surrogate-Assisted Evolutionary Optimization Framework," *Proc. of CEC 2004*, pp. 1586–1593, 2004.

[114] E. Zitzler, and L. Thiele, "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach," *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, 1999.

[115] E. Zitzler, K. Deb, and L. Thiele, "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation Journal*, Vol. 8, No. 2, pp. 173–195, 2000.

[116] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm," Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, 2001.