

Study on applying interactive
multi-objective optimization method
for power plant system design

Masakazu Shirakawa

Student ID number: 17D452

Dissertation submitted to the Graduate School of Engineering

Kagawa University

Thesis supervisor: Masao Arakawa

March 2019

Copyright © 2019 Masakazu Shirakawa All Rights Reserved.

Abstract

There is a blind spot that should be considered when applying optimization techniques to engineering design. In recent years, optimization techniques are being used in many procedures for product research and development, design, manufacturing, and repair service. The widespread use of optimization techniques for conventional design problems is probably as a result of improvements in computer performance. Therefore, many consider that optimized calculations involve “finding optimal solutions using the capability of a computer.”

Although the capabilities of computers and optimization tools have improved, we are often confronted with the difficulty of solving actual design problems. One of the reasons for this difficulty is mathematical modeling of the design objects. However, the problems faced by designers during decision-making are equally important. For example, in the engineering design tasks of a power plant, design engineers from different fields such as mechanical, electrical, chemical, control, information, architectural, and civil engineering work together. Then, cooperative design of the entire plant system is carried out, although emphasis is placed on a certain objective (evaluation index). However, a competitive relationship (trade-off relationship) exists such that it is necessary to achieve a reasonable compromise in other objectives. In other words, multiple-criteria decision-making based on appropriate value judgment by design engineers is required for optimum design with multiple evaluation indices.

To support the design problem involving multiple-criteria decision-making as described above, this doctoral research focused on an interactive multi-objective optimization method in which a designer and a computer cooperate to seek an optimal solution to the design problem with trade-off relationships among the evaluation indices. We developed a novel multi-objective optimization system, which was successfully applied to the system designs of a power plant, particularly plant operation design and plant layout design. We verified the usefulness of the proposed system by evaluating some design problems faced in actual power plants. The proposed method does not require the effort of registering design knowledge and rules on a computer as in the conventional method, and it has a feature that easily reflects the thinking and value judgment of designers in the resulting optimum design. Moreover, there is no

need to comprehensively calculate Pareto-optimal solutions beforehand as in the conventional method; the proposed method is superior in terms of its responsiveness to changes in the market environment and plant characteristics.

We believe that next-generation plant engineering design and operations will utilize virtual plants through digitization, as shown in Fig. A.1. That is, design databases such as piping and instrumentation diagram (P&ID), skeleton diagram, three-dimensional (3D) computer-aided design (CAD), control block diagram, and plant dynamic simulator cooperate with one another and automatically construct a virtual plant on a computer while interpreting design specifications. Therefore, many detailed design tasks will become automated with artificial intelligence (AI) and mechanization. Design engineers should concentrate on the optimum design of the entire system in the basic planning and design stage from a broader perspective. The interactive multi-objective optimization method can be expected to maximize the capabilities of design engineers in such advanced engineering environments.

Keywords

Control design, Dynamic simulation, Interactive optimization, Layout design, Multi-objective optimization, Multiple-criteria decision-making, Plant operation, Power plant, Satisficing trade-off method, Systems engineering

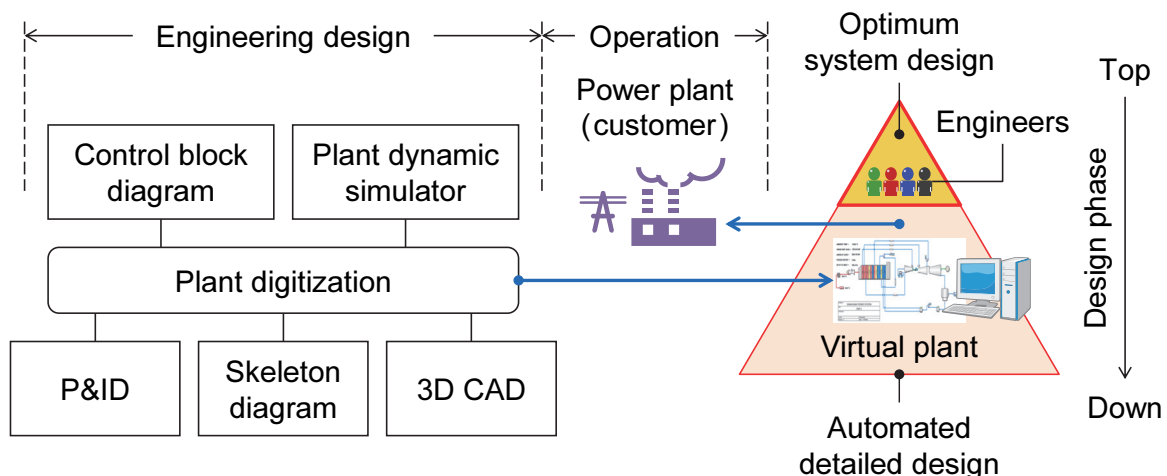


Fig. A.1 Concept of the next-generation plant system design. The next-generation plant engineering design and operations will utilize virtual plants through digitization.

Acknowledgment

First, I would like to thank Dr. Masao Arakawa (Professor, Kagawa University) who supervised my doctoral research; he gave me the opportunity to carry out research on this topic and offered many valuable suggestions. I would also like to thank the referees for this doctoral thesis, Dr. Tetsuro Butsuen (Professor, Kagawa University), Dr. Naka Gotoda (Associate Professor, Kagawa University), and Dr. Rihito Yaegashi (Associate Professor, Kagawa University), for detailed review of this thesis and for providing valuable comments to improve the thesis.

I would like to thank Dr. Hirotaka Nakayama (Professor Emeritus, Konan University) and Dr. Yeboon Yun (Professor, Kansai University) for the valuable discussions we had, and the brilliant ideas provided during my doctoral research.

I am grateful to my managers at Toshiba Energy Systems & Solutions Corporation (Toshiba ESS), Mr. Hideki Yoshikami (Senior Manager, Toshiba ESS) and Mr. Yuusuke Fushimi (Group Manager, Toshiba ESS), for supporting my doctoral study. I am also grateful to Mr. Kensuke Kawai (former Technology Executive, retired from Toshiba) for giving me the opportunity to carry out this study, and my co-workers, Mr. Shigeru Matsumoto (Senior Expert, Toshiba ESS) and Mr. Yuuki Okada (Chief Specialist, Toshiba ESS), for their help in the preparation of this study.

I also would like to thank my co-authors in previous articles, Dr. Masashi Nakamoto (former Chief Specialist, retired from Toshiba) and Mr. Shunji Hosaka (Chief Specialist, Toshiba), for their support.

Finally, I would like to thank my wife, Takako, for her dedication and many years of supporting me, and our daughter, Ruri, for everything she has done.

Takamatsu
December 2018

Masakazu Shirakawa

Contents

Abstract	3
Acknowledgment	5
List of figures	17
List of tables	21
Chapter 1 Introduction	23
1.1 Background	23
1.2 Interactive multi-objective optimization method	25
1.3 Power plant system design	27
1.4 Roadmap	29
1.4.1 Part I: System dynamics modeling and simulation of power plant	29
1.4.2 Part II: Multi-objective optimization system for plant operation design	30
1.4.3 Part III: Multi-objective optimization system for plant layout design	30
References	31
Part I System dynamics modeling and simulation of power plant	33

Chapter 2 Dynamic simulation and optimization of start-up processes in combined cycle power plants	35
Abstract	35
2.1 Introduction	35
2.2 Combined cycle power plant	36
2.3 Simulation model descriptions	37
2.3.1 Gas turbine model	37
2.3.2 Drum and evaporator model	37
2.3.3 Heat exchanger model	40
2.3.4 Steam turbine model	40
2.3.5 Pipe model	41
2.4 Verification of the simulation model	41
2.5 Optimization of the plant operation	44
2.5.1 Optimization method	44
2.5.1.1 Parameters	45
2.5.1.2 Objective functions	45
2.5.1.3 Constraints	46
2.5.1.4 Numerical algorithm	46
2.5.2 Application example	47
2.6 Conclusions	48
Appendices	51
Notation	51
References	53
 Chapter 3 Development of a thermal power plant simulation tool based on object orientation	 55
Abstract	55

3.1	Introduction	55
3.2	Development environment	56
3.3	Modeling method	57
3.3.1	Object-oriented model and internal state quantity calculation	57
3.3.2	Automatic parameter setting	60
3.3.3	Model library	65
3.3.4	Flow vector	66
3.4	Features in the tool use	67
3.4.1	GUI	67
3.4.2	Real-time calculation function	68
3.5	Application results	69
3.5.1	Verification of the dynamic simulation tool	69
3.5.2	Simulation example	69
3.5.3	Controller test facility	71
3.6	Virtual design environment with the hardware-less simulation system	74
3.7	Conclusions	76
	Appendices	77
	A1. Notation	77
	A2. Glossary of object-oriented terms	77
	References	78

Part II	Multi-objective optimization system for plant operation design	81
---------	-------------------------------------------------------------------------	----

Chapter 4	Intelligent start-up schedule optimization system for a thermal power plant	83
-----------	--------------------------------------------------------------------------------------	----

Abstract	83
4.1 Introduction	83
4.2 Combined cycle power plant	84
4.2.1 Plant configuration	84
4.2.2 Start-up scheduling method	85
4.3 Start-up scheduling problem	87
4.3.1 Schedule variables	87
4.3.2 Objective functions	87
4.3.3 Operational constraints	88
4.4 Concept of the system	88
4.4.1 Plant simulator part	89
4.4.2 Human interface part	89
4.4.3 Optimization calculation part	90
4.5 Intelligent optimization method	90
4.5.1 Satisficing trade-off method	90
4.5.1.1 On operation P	91
4.5.1.2 On operation T	92
4.5.2 Radial basis function network	93
4.5.3 Genetic range genetic algorithm	94
4.5.3.1 Definition of searching range	95
4.5.3.2 Inheritance of searching range	96
4.5.3.3 Initial searching range	97
4.5.3.4 New searching range	98
4.5.4 Numerical algorithm	99
4.6 Application results	100
4.6.1 Formulation	100
4.6.2 Training of RBFN	101

4.6.3 Multi-objective optimization	102
4.6.4 Computing time	105
4.7 Discussions	105
4.8 Economic evaluation	106
4.9 Conclusions	107
References	108

Chapter 5 Intelligent multi-objective model predictive control applied to steam turbine start-up	111
Abstract	111
5.1 Introduction	112
5.2 Proposed multi-objective model predictive control	113
5.3 Intelligent optimization method	118
5.3.1 Satisficing trade-off method	118
5.3.2 GA with blend alpha crossover	119
5.3.3 Model prediction using an RBFN	120
5.3.3.1 Proposed RRBFN prediction model	121
5.3.3.2 Cluster analysis for RRBFN training data	123
5.3.3.3 Verification of the proposed RRBFN prediction model	124
5.4 Application results	126
5.4.1 Plant system configuration	127
5.4.2 Problem definitions for the proposed multi-objective model predictive control	129
5.4.2.1 Optimizer	129
5.4.2.1.1 Design variables	129
5.4.2.1.2 Objective functions	130
5.4.2.1.3 Constraining conditions	131

5.4.2.2 Prediction model	132
5.4.3 Results of the dynamic simulation study	133
5.4.4 Computing time	137
5.5 Conclusions	138
Appendices	139
References	139

Chapter 6 Multi-objective evaluation method for combined cycle power plant start-up 143

Abstract	143
6.1 Introduction	143
6.2 Combined cycle power plant start-up	145
6.3 Multi-objective evaluation methods	146
6.3.1 SOM	146
6.3.2 DEA	148
6.4 Simulation evaluation	150
6.4.1 Analysis of start-up solutions by design of experiments	150
6.4.2 Evaluation of start-up solutions by SOM	152
6.4.3 Evaluation of start-up solutions by DEA	153
6.5 Conclusions	156
References	157

Part III Multi-objective optimization system for plant layout design 159

Chapter 7 Genetic algorithm approach for designing

layout of plot plan for power plans	161
Abstract	161
7.1 Introduction	161
7.2 Plant layout design and its evaluation indicators	163
7.3 Multi-objective optimization system for plant layout design	164
7.3.1 Interactive layout solution method	165
7.3.2 Algorithm	167
7.4 Layout optimization method developed using GA	168
7.4.1 Plant data	168
7.4.1.1 Outer shape dimension	168
7.4.1.2 Connections of pipes, ducts, and cables	170
7.4.1.3 Group arrangement	170
7.4.1.4 Site condition	171
7.4.2 Coding method	171
7.4.3 Considering the arrangement constraints	174
7.4.4 Evaluation functions	177
7.4.5 Genetic computation method	179
7.4.5.1 Evaluation and selection	179
7.4.5.2 Crossover	180
7.4.5.3 Mutation	181
7.4.5.4 Mutation of foreign species	181
7.5 Application results	181
7.5.1 Application results for trial design problem	181
7.5.2 Application results for thermal power plants	183
7.6 Conclusions	186
References	187

Chapter 8 Application of particle swarm optimization to automatic placement and adjustment of plant systems .. 189

Abstract	189
8.1 Introduction	189
8.2 Multi-objective optimization system for plant layout design	191
8.2.1 Overview of the previous study	191
8.2.2 Problems in the previous study	194
8.3 Summary of PSO	195
8.3.1 Standard PSO	195
8.3.2 Agent-type PSO	196
8.4 Proposed method	198
8.4.1 Design variables	198
8.4.2 Algorithm	200
8.5 Application results	204
8.5.1 Evaluation functions	204
8.5.2 Design conditions	205
8.5.3 Optimization results	207
8.5.4 Discussions	212
8.6 Conclusions	212
References	213

Chapter 9 Interactive multi-objective optimization technique for pipe routing design 215

Abstract	215
9.1 Introduction	216
9.2 Plot plan design and multi-objective optimization system	218
9.3 Proposed method	222

9.3.1	Temporary pipe rack layout according to a design rule	223
9.3.2	Extraction of pipe routing solutions according to ACO	226
9.3.3	Combinatorial optimization of pipe routing according to GA	229
9.4	Application results	231
9.4.1	Application results for trial design problem	231
9.4.1.1	ACO search results	231
9.4.1.2	Multi-objective optimization	234
9.4.1.3	Computing time	238
9.4.2	Application results for thermal power plants	239
9.4.2.1	Optimization result	239
9.4.2.2	Discussions	239
9.5	Conclusions	241
	Appendices	242
	References	245
	Summary and conclusions	249
	Chapter 10 Conclusions and future directions	251
10.1	Conclusions	251
10.2	Future directions	254
	List of publications by the author	255
	Publications included as part of this thesis	257

Journal papers	257
Book chapters	258
Expository articles	258
Peer-reviewed international conference papers	259
Awards related to this thesis	259

List of figures

A.1	Concept of the next-generation plant system design.	4
1.1	Concept of the interactive multi-objective optimization system.	25
1.2	Basic principles of the interactive multi-objective optimization method.	26
2.1	Schematic diagram of combined cycle power plant (one-shaft type).	36
2.2	Schematic picture of the drum and evaporator model.	38
2.3	Comparative results of simulation and actual plant operation data for hot start-up.	42
2.4	Comparative results of simulation and actual plant operation data for cold start-up.	43
2.5	Calculation flowchart which combined the dynamic simulation and the nonlinear programming, SQP.	47
2.6	Operational parameters (loading rates of the gas turbine).	48
2.7	Dynamic simulation results of conventional heuristic approach.	49
2.8	Dynamic simulation results of the proposed method (optimal solution).	50
3.1	Concept of object-oriented model.	58
3.2	Notation at an arbitrary segment surface (superheater model).	59
3.3	Concept of the heat exchanger (superheater) object.	64
3.4	Part of the development model library.	65
3.5	Flow vector linking of the object models.	67
3.6	Comparative results of simulation and actual plant operation data for warm start-up.	70
3.7	Schematic diagram of combined-cycle power generation system.	71

3.8	Part of the screen display of the whole plant model.	72
3.9	Example of the simulation result.	73
3.10	Schematic diagram of the controller test facility.	73
3.11	Photograph of the controller test facility.	74
3.12	Overview of the virtual design environment with the hardware-less simulation system for thermal power plants applying virtualization technologies.	75
4.1	Schematic diagram of combined cycle power plant (multi-shaft type).	85
4.2	Start-up curve for a multi-shaft type combined cycle power plant.	86
4.3	Plant operation optimization system.	89
4.4	Concept of the satisficing trade-off method.	92
4.5	Concept of the GRGA.	95
4.6	Crossover in the GRGA.	97
4.7	Training results of the RBFN models.	101
4.8	Results of the multi-objective optimization processes as a relation between three objective functions in two dimensions.	102
4.9	Initial start-up schedule corresponding to the first aspiration level (the symbol \square in Fig. 4.8).	103
4.10	Optimal start-up schedule obtained for the second aspiration level (the symbol \bullet in Fig. 4.8).	104
4.11	Results of the Monte Carlo simulations.	107
5.1	Structure of the proposed multi-objective model predictive control system.	115
5.2	Discrete model predictive control scheme.	116
5.3	Proposed RRBFN prediction model.	122
5.4	Cluster analysis for a two-dimensional variable space.	124
5.5	Configuration of NO _x decomposition in a thermal power plant.	125

5.6	Comparison of predicted and actual plant operation values of the chimney inlet NO _x flow rate.	126
5.7	Configuration of a typical steam power plant system.	128
5.8	Results of multi-objective optimization process.	134
5.9	Dynamic simulation results obtained for time-optimal model predictive control corresponding to point C in Fig. 5.8a.	135
5.10	Dynamic simulation results obtained for the first aspiration level.	136
5.11	Dynamic simulation results obtained for the second aspiration level. ..	137
6.1	Concept underlying the multiple-criteria decision-making system based on tradeoff analytics for plant operation.	144
6.2	Basic structure of a two-dimensional SOM known as a Kohonen network.	147
6.3	DEA as a dual-objective problem.	150
6.4	Correlation diagram between each pair of evaluation functions in two dimensions.	151
6.5	Contour line visualizations by the SOM.	153
6.6	Results of the start-up dynamic simulation for solution B (the symbol ▲ in Fig. 6.4).	156
6.7	Results of the start-up dynamic simulation for solution D (the symbol ■ in Fig. 6.4).	156
7.1	Flowchart showing the functional structure of the proposed equipment layout design system.	166
7.2	Configuration of equipment/buildings.	170
7.3	Directions of equipment/building.	172
7.4	Maintenance spaces around equipment/building.	172
7.5	Installation points with respect to equipment/building.	173
7.6	Types of arrangement constraints.	175
7.7	Design data of the equipment, building structures, and the site condition, referring to the plant data in Table 7.1.	176

7.8	Cording method and phenotype of plant layout.	177
7.9	Crossover.	180
7.10	Mutations.	181
7.11	Results for each aspiration level.	182
7.12	Convergence history of the proposed GA.	183
7.13	Design equipment layout of the plot plan for a gas-turbine combined-cycle power plant.	185
8.1	GA coding method.	192
8.2	Design variables.	199
8.3	Layout by a skilled designer.	206
8.4	GA layout.	209
8.5	Layout by the proposed PSO.	210
8.6	Convergence history of the proposed PSO.	211
9.1	Flowchart showing the functional structure of the proposed pipe routing design system.	221
9.2	Equipment layout of the site in the trial design problem.	224
9.3	Temporary placement of the pipe rack for the trial design problem in Fig. 9.2 based on the design rule.	225
9.4	Search results for each connection of pipes and cables by the proposed ACO.	232–233
9.5	Pipe rack placement results of the trial design problem in Fig. 9.3 obtained for each aspiration level together with an initial route solution.	236
9.6	Design pipe routing of the plot plan for a gas-turbine combined-cycle power plant.	240

List of tables

2.1	Comparison results of the start-up time	44
4.1	Approximation errors of the RBFN models	101
4.2	Simulation results of the objective functions	104
5.1	Computing times for training an RRBFN in each clustering case	126
5.A	Functional table of the speed-up rate $\Delta u_S(t + p_1)$ and loading rate $\Delta u_{L, \text{tbl}}$ in Eq. (5.23)	139
5.B	Functional table of the steam turbine speed and load u , and the boiler outlet steam pressure P_s and temperature T_s	139
6.1	Representative calculation results of DEA	155
7.1	Samples of the plant data	169
7.2	Example of the chromosome of GA	171
7.3	Application results through six trial calculations by the proposed GA	184
8.1	Application results through six trial calculations by the proposed PSO	208
9.1	Application result for the trial design problem in Fig. 9.3	235
9.2	Application result for a gas-turbine combined-cycle power plant	241
9.A	Piping and cabling line list for the trial design problem	242
9.B	Pipe and cable routing results of the trial design problem in Fig. 9.3 obtained for each aspiration level together with an initial route solution	243

Chapter 1

Introduction

1.1 Background

In projects of social infrastructure systems such as energy and environment, water supply and sewerage, roads, railways, airports, urban and building, and information communication where experts from different fields participate, various opinions and ideas are given at the upstream design phase, and the system design often becomes increasingly complicated. This study was conducted to overcome such problems; that is, an optimization method that can be considered for the overall system design was established to improve the system performance and achieve appropriate balance among various evaluation indices while satisfying safety requirements and avoiding over-engineered of the entire system. Moreover, in this study, we considered that it is possible to create a completely new technical innovation by accepting diverse system requirements.

For example, thermal and nuclear power plants can be described as large-scale systems having a complicated hierarchical structure as they are composed of many systems, machines, and components. In the system design of such a power plant, naturally, the technical development of individual machine and component is important. Furthermore, a system design that captures the overall characteristics of the plant system and its design optimization is essential. The evaluation indices in the system design of power plants are diverse. Specifically, on the premise of ensuring safety of life and property, the evaluation indices have multiple objectives taking many factors into consideration such as maintainability, environmental feasibility, economic efficiency, construction workability, operational controllability, and reliability. These evaluation indices are treated as multi-objective optimization problems. Although the capabilities of computers and optimization tools have improved, we are often confronted with the difficulty of solving problems in actual plant system design. The reasons for the difficulty include the mathematical modeling of the design object; however, the problem of decision-making by the designer is equally important. In

engineering design tasks for a power plant, design engineers with a variety of expertise such as mechanical, electrical, chemical, control, information, architectural, and civil work together in the project. In addition, engineering design, from basic planning to construction, has many processes and takes several years to complete. Design engineers are required to make appropriate value judgment according to design processes taking the design progress and market changes into consideration.

The system design of a power plant must satisfy safety requirements and take into consideration a plurality of evaluation indices. In addition, features such as the need for flexible decision-making according to design progress and market changes should also be considered. Such advanced engineering designs are mostly dependent on the expertise and intuition of skilled designers, which may increase the project cost and risk. Consequently, the problem of how to carry out the overall system optimal design has not been addressed.

Therefore, the aim of this doctoral research is to develop an interactive multi-objective optimization system that supports multiple-criteria decision-making for the system design of a power plant through interaction between a designer and a computer, as shown in Fig. 1.1. This system consists of a simulation model that calculates various evaluation indices of the plant system design and an interactive multiple-criteria decision-making system that implements a multi-objective optimization algorithm. The basic planning and design of the plant system can be categorized into three major design problems: process design, where heat balance calculations of the process flowsheet are carried out; operation design, where a control strategy is developed; and layout design, where a plot plan (general arrangement drawing) is created. Since the process design can be mostly evaluated objectively (non-subjectively), we focused on the remaining two design problems in this research, namely operation design and layout design, which are likely to reflect the designers' (or operators') subjective decision. Further, since we intend to use this system in the upstream design phase (for instance, basic planning and design), which has high degree of design freedom for decision-making, the system should be able to examine numerous design candidates in a limited time frame using simple design data.

The structure of this chapter is as follows. Section 1.2 explains the concept of interactive multi-objective optimization method, which is the framework of the system shown in Fig. 1.1, and Section 1.3 explains the power plant system design, while Section 1.4 presents the roadmap for this thesis.

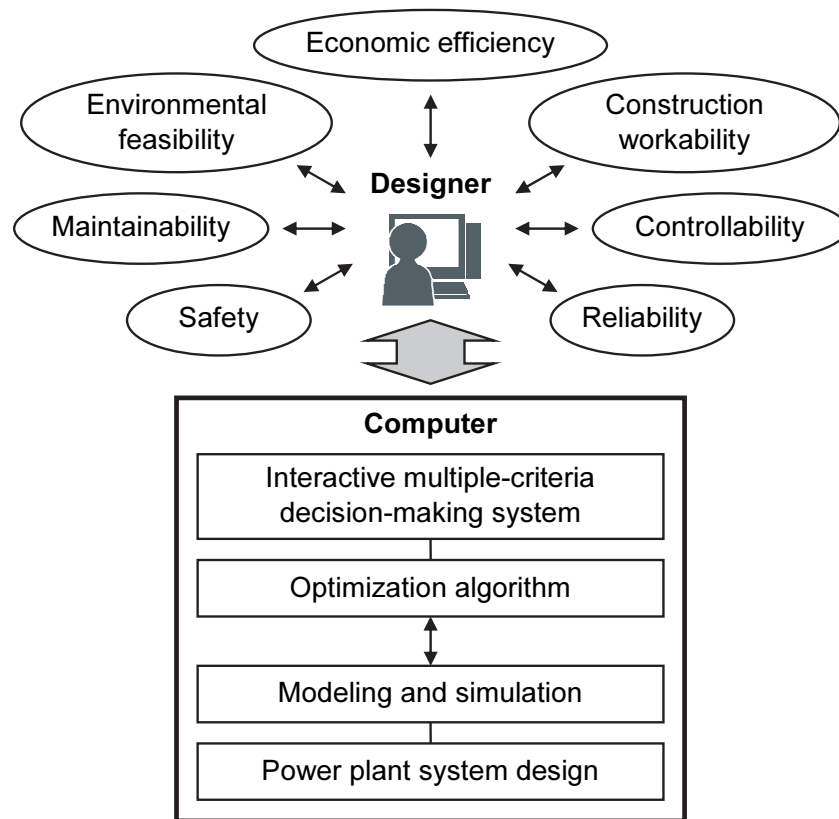


Fig. 1.1 Concept of the interactive multi-objective optimization system. This system supports multiple-criteria decision-making for system design of a power plant through interaction between a designer and a computer.

1.2 Interactive multi-objective optimization method

In multi-objective optimization problems, a trade-off relationship exists among objectives in general; thus, it is not possible to optimize all the objectives at the same time. Therefore, the limiting state where all objectives cannot be improved any more can be called the Pareto-optimal solution. It is not determined to be a single solution, rather it is a solution set. The solution set can be determined using the method of multi-point simultaneous search typified by a multi-objective genetic algorithm (MOGA) and a multi-objective evolutionary algorithm (MOEA) (Horn et al., 1994; Fonseca and Fleming, 1995; Deb, 2001; Coello Coello et al., 2007); however, the disadvantage of this method is that the required human and computation resources are high. Moreover, a great deal of labor is needed to extract the solution intended by the

designer from the Pareto-optimal solution set. On the other hand, in an interactive solution method, the solution intended by the designer is sequentially obtained through interaction between the designer (decision-maker) and a computer. In this method, it is possible to support the multiple-criteria decision-making of the designer while maintaining a low calculation load, as the designer can understand the step-by-step process of the design problem. In the satisficing trade-off method (Nakayama, 1995), which is one of the interactive multi-objective optimization methods, the Pareto-optimal solution satisfied by the designer is found while alternately repeating the following interactive steps.

[Computer] Search for the Pareto-optimal solution closest to the given aspiration level and present it to the designer as a solution candidate.

[Designer] If the designer is not satisfied with the presented solution candidates, modify the aspiration level based on the trade-off relationship.

The basic principles of the satisficing trade-off method are described below using a simple example of two objective functions as shown in Fig. 1.2. The aspiration level, \bar{f}_i (the symbol \bigcirc in Fig. 1.2), is the target value of the designer that is preferable under the present condition for the objective function (evaluation function), f_i ($i = 1, 2, \dots, r$). The ideal point, f_i^* (the symbol $*$ in Fig. 1.2), is the origin; for instance, the lower limit value for each objective function. The broken line in Fig. 1.2 indicates the Pareto frontier (a curved surface mapping the Pareto-optimal solution set

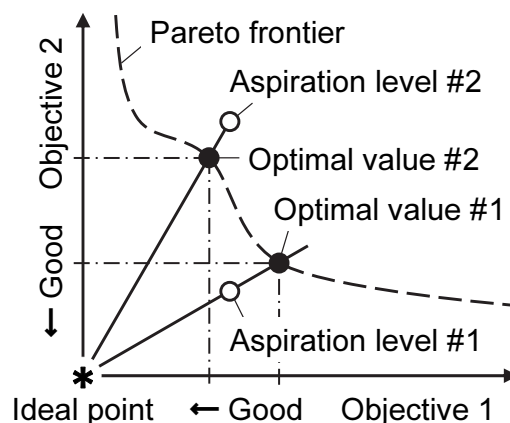


Fig. 1.2 Basic principles of the interactive multi-objective optimization method. This figure shows a simple example of two objective functions using the satisficing trade-off method.

in the objective space). The obtained solution is one of the Pareto-optimal solutions, which corresponds to the optimum value (the symbol ● in Fig. 1.2) at the intersection of the Pareto frontier and the line (the solid line in Fig. 1.2) passing through the aspiration level and the ideal point. Since this intersection can be represented by the contact point with the contour line of the weighted-uniform norm (Chebyshev norm) (one-dot chain line in Fig. 1.2), the solution search can be reduced to the problem of obtaining a design variable vector, \mathbf{x} , that minimizes the scalarization function, F , of the following expression:

$$F(\mathbf{x}) = \max_{1 \leq i \leq r} \left[\frac{f_i(\mathbf{x}) - \overline{f_i}}{f_i - f_i^*} \right] \rightarrow \min_{\mathbf{x}} \quad (1.1)$$

In Fig. 1.2, the operation is shown in the order of aspiration level #1 to optimal value #1 (convex surface), aspiration level #2 to optimal value #2 (non-convex surface). In this approach, the property that the setting of the aspiration level and the behavior of the solution substantially agree with the intention of the designer has some aspects that aid tradeoff analytics.

Now, in conventional optimization techniques, it is necessary to set the objective functions in advance, so that it is difficult to deal with issues affected by human subjectivity. On the other hand, the interactive solution method is suitable for design problems that cannot relinquish value judgment to computers alone. For example, there are design problems that require the use of subjective value and skill of the designer. In the plant operation and plant layout design problems discussed in this thesis, it is difficult to determine the accuracy of the result only by objective (non-subjective) data; thus, the accuracy can be improved through the designer's subjective decision.

1.3 Power plant system design

There has been a progress in the efforts to achieve high thermal efficiency and increased power output by the raising the steam temperature and pressure for thermal power plants, while simultaneously being increasingly environment friendly, such as the generation of a stable supply of the electrical energy by diversification of fuel and reduction of carbon dioxide (CO₂) emission. Furthermore, with regard to the practical use of electric power systems, the functional extension of the load adjusting

performance over the thermal power plant has become strong, while frequent start-up and shutdown, and rapid and large load changes are obliged. It is thought that this tendency could become stronger in future with an expansion of renewable energy sources such as wind and solar radiation, wherein production of electricity tends to be influenced by the climatic conditions. Taking this into account, functional requirements of the system design of the thermal power plant for a customer is further developed. Moreover, in emerging countries market, as there is a huge difference in electric power supply and demand as a result of remarkable economic growth, quick completion and delivery of plants are desired and the time that can be spent on system design is naturally less.

The system design of power plants requires adaptability and flexibility for rational decision-making for the following three reasons:

1. Because it is a build-to-order product, the amount of design data, which is huge and the standardization of which is difficult, needs to be taken into account.
2. Because it is a long project from the time an order is received until its delivery, a system design needs to correspond to progress and changes in design, allowing a smooth exchange of design data among plural departments.
3. Because the life cycle of a plant is long, the system design should be able to take into consideration changes in the market environment.

Therefore, a system design method in which an interactive multi-objective optimization method, which has the adaptability and flexibility for rational decision-making through human-computer interactions, is applied to such high-level design problems.

Power plants are large-scale indent products (build-to-order products) in which the requirement specifications of the system differ for every site. For example, although the construction duration necessary for completion of a thermal power plant changes with geographical conditions or power outputs, it takes about three to four years from when an order is received until its delivery. In the construction project of a power plant, there are roughly three processes—engineering (E), procurement (P), and construction and commissioning (C)—together called the EPC project, and each process is carried out in close cooperation. In particular, the system design in the upstream design phase (for instance, basic planning and design) of engineering process heavily influences the execution of projects, such as cost, quality, and delivery date. As the preceding section described, the basic planning and design of the plant system can be categorized into three major design problems: process design, operation design, and layout design.

Although this doctoral dissertation focuses on operation design and the layout design, the physical configuration of the system, which is decided by process design, and the layout design that embodies it, are closely related. Moreover, the layout design and the operation design for the safety systems in place for the eventuality of a severe accident, state of emergency, etc. are developed in cooperation. Furthermore, the use of concurrent engineering, which has been helping advance the plant system in recent years and which takes into consideration simultaneously a process design (equipment hardware design) and an operation design (control software design), is recommended. Hence, while the process design, the operation design, and the layout design are related to one another, a system-wide optimal design is desired.

1.4 Roadmap

This thesis is organized into three main parts, conclusions, and future directions. In the first part, the dynamic simulation used in plant operation design is presented. In the second part, the multi-objective optimization system of the plant operation design is presented. In the third part, the multi-objective optimization system of the plant layout design is described.

1.4.1 Part I: System dynamics modeling and simulation of power plant

In Part I, the dynamic simulation used to evaluate the operating characteristics of the power plant is described. Dynamic simulation is used to calculate multiple evaluation function values in the multi-objective optimization system of the plant operation design presented in Part II. This part comprises two chapters, namely Chapters 2 and 3.

Chapter 2 presents the mathematical model (dynamic simulation model) that represents the system dynamics of the power plant. Moreover, the accuracy of the dynamic simulation model is verified using actual operation data of the power plant. This chapter also presents the mathematical formulas of various evaluation functions for the plant operation optimization presented in Part II.

Chapter 3 describes the dynamic simulation tool developed by applying the mathematical model presented in Chapter 2, which enables a flexible plant configuration by modeling various equipment components based on object orientation. This tool is applied to the plant simulator that constitutes the plant operation optimization system described in Chapter 4 of Part II. Furthermore, the developed

controller simulation–testing environment is utilized to verify the performance of the optimization and evaluation methods proposed in Part II.

1.4.2 Part II: Multi-objective optimization system for plant operation design

In Part II, the application of the interactive multi-objective optimization method to the operational design problem of the power plant is described. This part comprises three chapters, namely Chapters 4–6.

Chapter 4 describes an intelligent operation optimization system for the power plant developed using the plant simulator presented in Chapter 3 of Part I. In this study, a novel intelligent optimization algorithm, which combines surrogate-based optimization with radial basis function network and the satisficing trade-off method, is proposed and introduced into the system. The system can determine an optimal and flexible operation schedule from multiple objectives at a reasonable computing time and calculation accuracy through human–computer interactions.

In Chapter 5, we extended the optimization algorithm discussed in Chapter 4 for real-time (online) optimal control. In this study, a novel intelligent multi-objective model predictive control method is proposed for a dynamic environment with multiple objectives in which a recurrent radial basis function network is applied for model prediction, and the satisficing trade-off method is used to construct the multi-objective optimization. This control method can perform optimal and flexible plant control from multiple objectives within a limited practical computing time in real-time through human–computer interactions.

Chapter 6 describes an analysis method to evaluate the obtained optimal solution in order to adapt the multi-objective optimization method to the plant operation discussed in Chapters 4 and 5 for practical use. In this study, multi-objective evaluation methods are proposed for plant operation using self-organizing maps and data envelopment analysis. The analysis method can support multiple-criteria decision-making based on tradeoff analytics for extracting the optimal solution desired by designers (or operators).

1.4.3 Part III: Multi-objective optimization system for plant layout design

In Part III, the application of the interactive multi-objective optimization method to the layout design problem of the power plant is described. This part comprises three chapters, namely Chapters 7–9.

Chapter 7 describes the development of a multi-objective optimization system for plant layout design, which was constructed using the satisficing trade-off method. In this study, a layout optimization method that uses a genetic algorithm is proposed, which can generate various layout solutions at high speed, and is introduced into the system. The system can determine an optimal layout solution of equipment and building structures from multiple objectives through human–computer interactions, with low computation load and simple design data.

Chapter 8 describes a more practical design support method that compensates for the limitations of solution search performance observed in the layout optimization method discussed in Chapter 7. In this study, an automatic layout adjustment method is proposed, which employs a modified particle swarm optimization algorithm to find the true global optimal solution near the layout solution obtained by the genetic algorithm described in Chapter 7. This method can improve the overall layout through automatic layout adjustment taking multiple objectives into consideration.

In addition to equipment and building structures layout discussed in Chapters 7 and 8, Chapter 9 describes a multi-objective optimization system for pipe and cable routing design, which was constructed using the satisficing trade-off method. In this study, a route optimization method that uses a multi-ant colony optimization algorithm to generate various route solutions at high speed is proposed and introduced into the system. The system can simultaneously optimize the pipe and cable routes, as well as pipe rack layouts from multiple objectives through human–computer interactions, with low computation load and simple design data.

References

- Coello Coello, C. A., Lamont, G. B. and Van Veldhuizen, D. A., *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd Edition, (2007), Springer Science+Business Media, LLC, New York, NY, the United States.
- Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, (2001), John Wiley & Sons, Ltd., Chichester, the United Kingdom.
- Fonseca, C. M. and Fleming, P. J., An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary Computation*, Vol. 3, Issue 1, (1995), pp. 1–16.
- Horn, J., Nafpliotis, N. and Goldberg, D. E., A niched Pareto genetic algorithm for

multiobjective optimization, Proceedings of the First IEEE Conference on Evolutionary Computation (ICEC'94), IEEE World Congress on Computational Intelligence, Vol. 1, (1994), pp. 82–87, Orlando, FL, the United States.

Nakayama, H., Aspiration level approach to interactive multi-objective programming and its applications, In: Advances in Multicriteria Analysis (Pardalos, P. M., Siskos, Y. and Zopounidis, C., eds.), (1995), pp. 147–174, Kluwer Academic Publishers, Dordrecht, the Netherlands.

Part I

System dynamics modeling and simulation of power plant

Chapter 2

Dynamic simulation and optimization of start-up processes in combined cycle power plants

Abstract

This study treats the development of a dynamic simulation model and its application to the optimization of a start-up process for a combined cycle power plant. Generally, the plant system design is complicated, and the control design is difficult to establish without dynamic simulation. The comparison of the simulation results and the actual plant operation data is reported. The obtained results demonstrate that the simulation is reliable to evaluate the plant dynamic phenomenon and available to predict the operational processes. However, not only an analytical method but also the development of a design method is important to determine the optimal operational procedure. In this study, the dynamic simulation and the nonlinear programming are combined, and it proposes the design method that optimizes the operational parameters.

Keywords

Combined cycle, Dynamic simulation, Modeling, Nonlinear programming, Optimal operation, Optimization, Power plant, Start-up, Steam turbine, Thermal stress

2.1 Introduction

The demand for flexible operation of a combined cycle power plant is becoming increasingly important because this plant is particularly suited for load control. Its operational characteristics have been intensively studied by the dynamic simulation (Akiyama et al., 1993; Mastronarde and Shiue, 1995), but it is not easy to determine the optimal operational parameters because it is necessary to iterate the dynamic simulation based on trial and error by the engineer's intuition and experience.

Several proposals to improve start-up scheduling for a power plant have been made but each has its limitations. For instance, an adaptive-edge search operation used with

genetic algorithms has been proposed (Kamiya et al., 1999). This method considers only thermal stress in the steam turbine rotor as an operational restriction. The combined cycle power plant, however, has many operational restrictions, not only thermal stress in the steam turbine rotor but also loading rates of the gas turbine, temperature gradients of the heat recovery steam generator, NO_x emission rate from the plant, etc. A fuzzy expert system that uses an engineer's experiences in fuzzy rules has been proposed (Matsumoto et al., 1996, 1997). It requires additional cost to obtain this engineering knowledge.

This study aims at proposing a design method to determine the practical optimal operational parameters using a dynamic simulation and nonlinear programming. This method does not require a great deal of labor in order to prepare the knowledge base.

2.2 Combined cycle power plant

The configuration of the combined cycle power plant is shown in Fig. 2.1. This plant consists of a gas turbine, a heat recovery steam generator (HRSG), a steam turbine, and a generator. The gas turbine and the steam turbine drive the generator. Also, the HRSG generates steam for the steam turbine using waste heat from the gas turbine. The HRSG

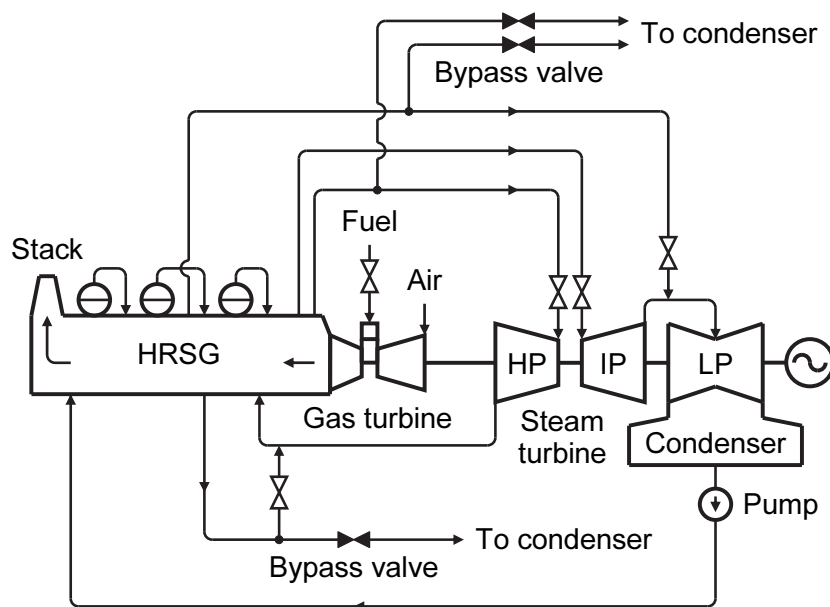


Fig. 2.1 Schematic diagram of combined cycle power plant (one-shaft type). This plant consists of a gas turbine, an HRSG, a steam turbine, and a generator.

is the triple pressure and reheat type. The steam turbine has a high-pressure (HP) turbine, an intermediate-pressure (IP) turbine, and a low-pressure (LP) turbine.

2.3 Simulation model descriptions

The dynamic simulation model consists of the process equipment model and the control system model. The control system is modeled to be equivalent to the actual control system. The first principle is adapted to model the process equipment. The following are brief descriptions of major models.

2.3.1 Gas turbine model

The gas turbine responds more quickly than the other equipment. The flow rate and the temperature of the exhaust gas are represented by functions of the fuel flow rate. The gas turbine duct has a time delay because of the large heat capacity. Therefore, the gas turbine model considers the heat capacity of the duct. The rotor dynamics can be disregarded in the start-up process, though it is considered in a rapid response as the load rejection.

2.3.2 Drum and evaporator model

The HRSG is composed of heat exchangers that have a long time delay when compared to the gas turbine and the steam turbine. It has a significant effect on the starting characteristics of the entire plant. The model of the drum and evaporator is shown in Fig. 2.2, and the basic equations are described in Eqs. (2.1) to (2.12). The metal of the drum and evaporator has a time delay because of the large heat capacity. Therefore, the drum and evaporator model considers the heat capacity of the metal in Eqs. (2.3), (2.8), and (2.9).

Drum mass balance (liquid phase):

$$\frac{d(M_{dw})}{dt} = G_w - G_{bo} - \sum G_c + \sum (1-x)G_e + G_{cd} - G_{dv} . \quad (2.1)$$

Drum mass balance (steam phase):

$$\frac{d(M_s)}{dt} = \sum x G_e - G_s - G_{cd} + G_{dv} . \quad (2.2)$$

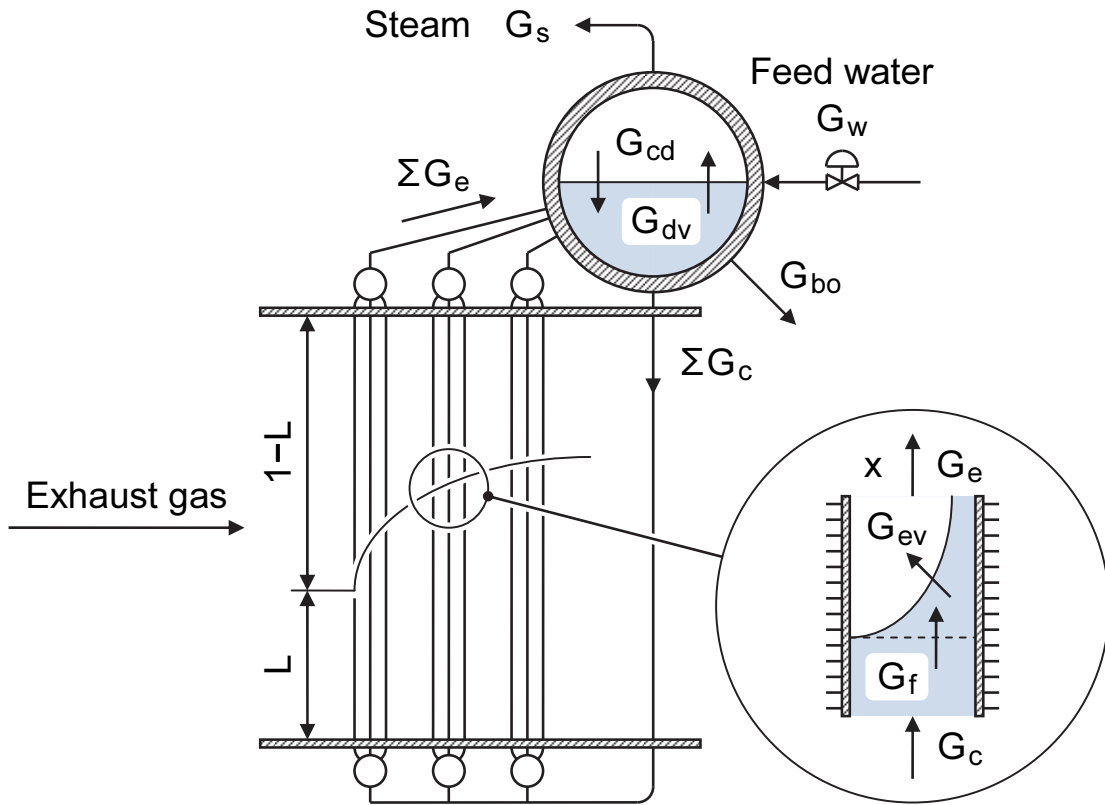


Fig. 2.2 Schematic picture of the drum and evaporator model. A nonlinear dynamic model for natural circulation drum-evaporators is presented.

Drum energy balance (liquid phase):

$$\begin{aligned} \frac{d(M_{dw} h_{dw} + M_{dm} C_{dm} T_{dm})}{dt} &= G_w h_w - G_{bo} h_{dw} - \sum G_c h_{dw} \\ &+ \sum (1-x) G_e h_{sw} + G_{cd} h_{dw} - G_{dv} h_s + M_{dw} v_{dw} \left(\frac{dP}{dt} \right). \end{aligned} \quad (2.3)$$

Drum energy balance (steam phase):

$$\frac{d(M_s h_s)}{dt} = \sum x G_e h_s - G_s h_s - G_{cd} h_{dw} + G_{dv} h_s + M_s v_s \left(\frac{dP}{dt} \right). \quad (2.4)$$

Evaporator mass balance (subcooled water phase):

$$\frac{d(M_{wu})}{dt} = G_c - G_f. \quad (2.5)$$

Evaporator mass balance (saturated water phase):

$$\frac{d(M_{sw})}{dt} = G_f - (1-x)G_e - G_{ev} . \quad (2.6)$$

Evaporator mass balance (saturated steam phase):

$$\frac{d(M_s)}{dt} = G_{ev} - xG_e . \quad (2.7)$$

Evaporator energy balance (subcooled water phase):

$$\begin{aligned} \frac{d(M_{wu} h_{wu} + M_{eum} C_{eum} T_{eum})}{dt} &= QL + G_c h_{dw} - G_f h_{sw} \\ &+ M_{wu} v_{wu} \left(\frac{dP}{dt} \right) . \end{aligned} \quad (2.8)$$

Evaporator energy balance (saturated water phase):

$$\begin{aligned} \frac{d(M_{sw} h_{sw} + M_{esm} C_{esm} T_{esm})}{dt} &= Q(1-L) + G_f h_{sw} - (1-x)G_e h_{sw} \\ &- G_{ev} h_s + M_{sw} v_{sw} \left(\frac{dP}{dt} \right) . \end{aligned} \quad (2.9)$$

Evaporator energy balance (saturated steam phase):

$$\frac{d(M_s h_s)}{dt} = Q(1-L) + G_{ev} h_s - xG_e h_s + M_s v_s \left(\frac{dP}{dt} \right) . \quad (2.10)$$

Total volume constant (drum):

$$\frac{d(M_{dw} v_{dw} + M_s v_s)}{dt} = 0 . \quad (2.11)$$

Total volume constant (evaporator):

$$\frac{d(M_{wu} v_{wu} + M_{sw} v_{sw} + M_s v_s)}{dt} = 0 . \quad (2.12)$$

In the two-phase flow section of the evaporator, the precise estimation of the void fraction and the circulating flow rate is important. Though the dynamic modeling was

referred to the past study (Akiyama et al., 1993), the heat transfer rate and the pressure drop in the evaporator are evaluated from our experimental data of the test plant (Shirakawa, 2003).

2.3.3 Heat exchanger model

The basic equations of the heat exchanger are described in Eqs. (2.13) to (2.15). The heat transfer coefficients are defined by the general formula of turbulent flow.

$$Q_g = G_g C_g (T_{g, in} - T_{g, out}) = A_g k_g (T_g - T_m), \quad (2.13)$$

$$Q_{stm} = G_{stm} (h_{stm, out} - h_{stm, in}) = A_{stm} k_{stm} (T_m - T_{stm}), \quad (2.14)$$

$$\frac{d(T_m)}{dt} = \frac{Q_g - Q_{stm}}{M_m C_m}. \quad (2.15)$$

2.3.4 Steam turbine model

The basic equations of the steam turbine are described in Eqs. (2.16) to (2.18). The adiabatic efficiency of the steam turbine is defined as the turbine pressure ratio. The steam flow rate through the steam turbine is calculated using the constant flow coefficient. The turbine power is calculated using the enthalpy difference between the inlet and outlet of the steam turbine. No dynamic effect is evaluated in the flow and pressure relationship.

$$h_{tb, out} = h_{tb, in} - \eta (h_{tb, in} - h_{tb, or}), \quad (2.16)$$

$$G_{tb} = \zeta S \sqrt{\frac{(P_{tb, in} - P_{tb, out}) \times 10^3}{v_{tb, in}}}, \quad (2.17)$$

$$W_{tb} = G_{tb} (h_{tb, in} - h_{tb, out}). \quad (2.18)$$

The basic equations of the thermal stress are described in Eqs. (2.19) and (2.20). The thermal stress of the steam turbine rotor is obtained by the temperature distribution,

which is calculated by the dynamic model of the thermal conduction that divides the rotor into several vertical cylinders.

$$\frac{\partial \theta}{\partial t} = \frac{\lambda}{C \rho} \left(\frac{\partial^2 \theta}{\partial r^2} + \frac{1}{r} \frac{\partial \theta}{\partial r} \right), \quad (2.19)$$

$$\sigma_{zz} = \frac{E \alpha_{\ell}}{1 - \mu} \left(\frac{2}{R_o^2 - R_i^2} \int_{R_i}^{R_o} \theta r \, dr - \theta \right). \quad (2.20)$$

2.3.5 Pipe model

Since the steam pipe is long and has a large volume, the pressure in the pipe is modeled by the multiple parts and volumes in which the steam is stored. The basic equations of the pipe are described in Eqs. (2.21) to (2.23).

$$\frac{d(M)}{dt} = G_{in} - G_{out}, \quad (2.21)$$

$$\frac{d(Mh)}{dt} = G_{in} h_{in} - G_{out} h_{out}, \quad (2.22)$$

$$G = \zeta(z) S \sqrt{\frac{(P_{in} - P_{out}) \times 10^3}{v_{in}}}. \quad (2.23)$$

2.4 Verification of the simulation model

The actual plant operation data were measured from one-shaft type and triple pressure with reheat cycle 238 MW rated capacity, as shown in Fig. 2.1. The firing temperature of the gas turbine was 1300°C class. The comparison with hot start-up (standby period is 8 hours) data is shown in Fig. 2.3. The comparison with cold start-up data is shown in Fig. 2.4. Figures 2.3a and 2.4a show the gas turbine exhaust gas temperature and flow rate response. Figures 2.3b and 2.4b show the HP steam temperature, pressure, and flow rate response. Figures 2.3c and 2.4c show the IP steam temperature, pressure, and flow rate response. Figures 2.3d and 2.4d show the LP steam

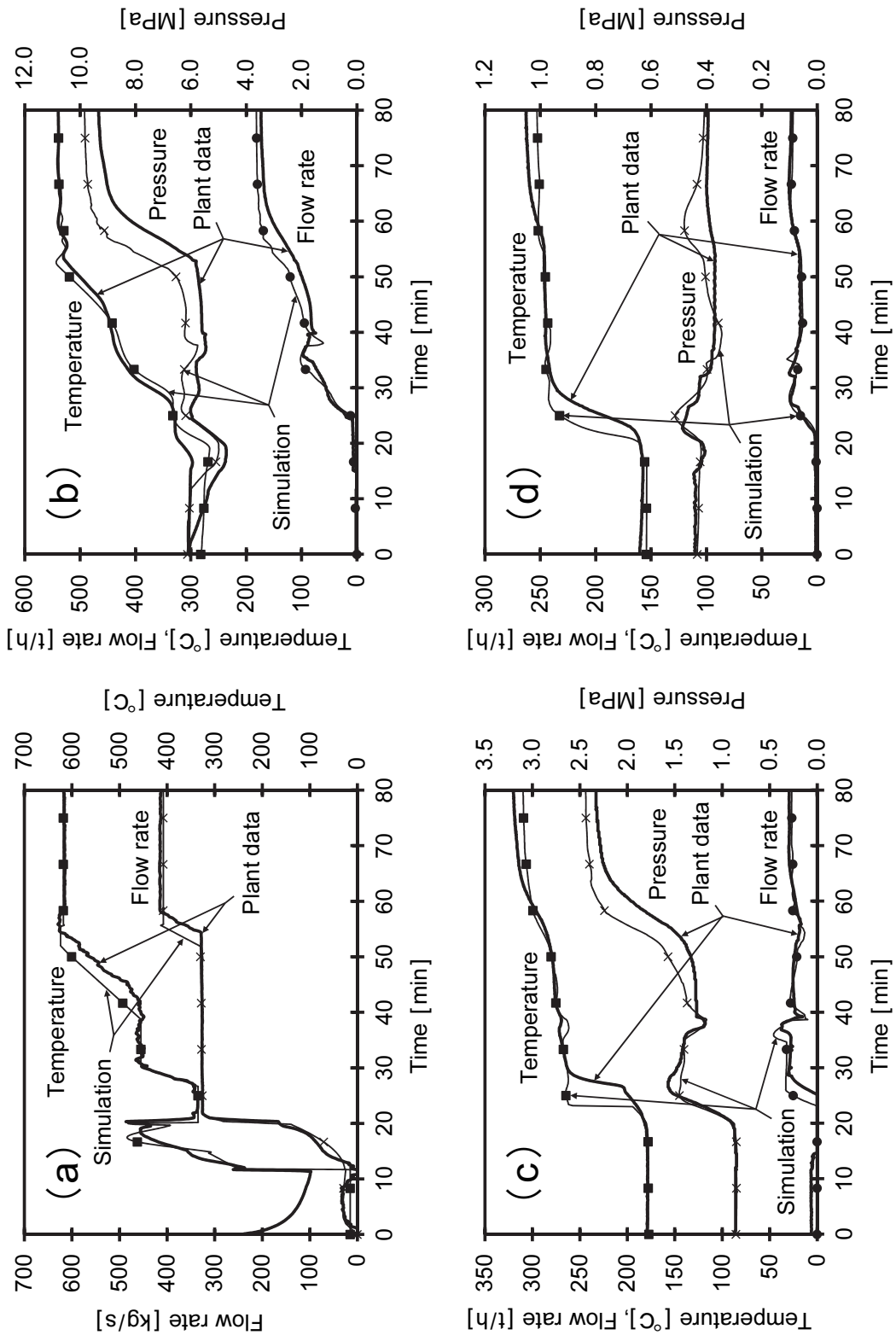


Fig. 2.3 Comparative results of simulation and actual plant operation data for hot start-up. The symbols lines are the simulation results and the solid lines are the plant operation data. (a) Gas turbine exhaust gas response. (b) HP steam response. (c) IP steam response. (d) LP steam response.

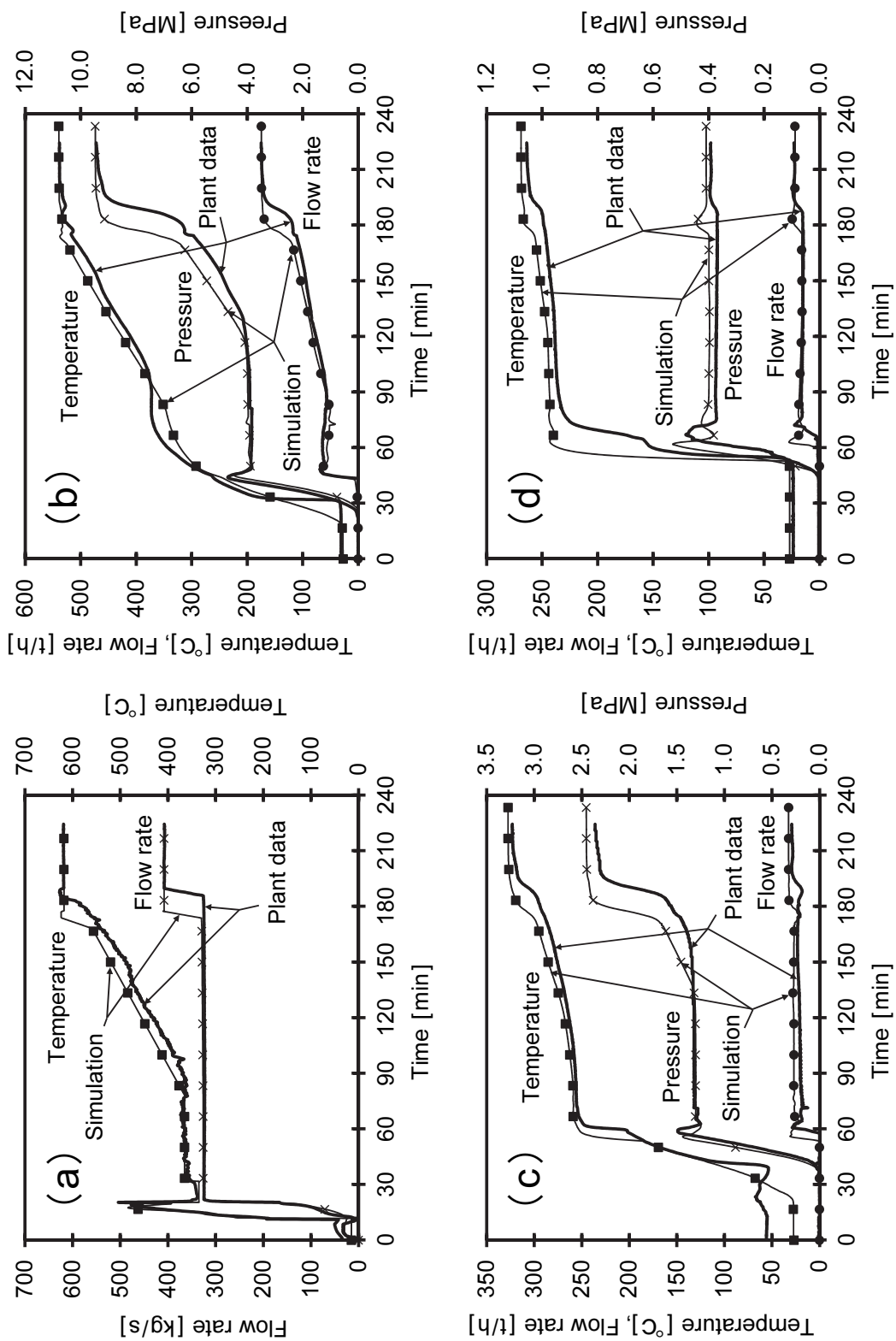


Fig. 2.4 Comparative results of simulation and actual plant operation data for cold start-up. The symbols lines are the simulation results and the solid lines are the plant operation data. (a) Gas turbine exhaust gas response. (b) HP steam response. (c) IP steam response. (d) LP steam response.

Table 2.1 Comparison results of the start-up time

	Simulation	Plant data
Hot start-up	56 [min]	58 [min]
Cold start-up	178 [min]	189 [min]

temperature, pressure, and flow rate response. The symbols lines are the simulation results and the solid lines are the plant operation data in Figs. 2.3 and 2.4. The simulation results are in good correspondence with the plant operation data. Accordingly, the simulation model will be available to use in the plant system and control design.

The comparison results of the start-up time from the gas turbine start to the base-load operation are shown in Table 2.1. The differences of the start-up time are only 2 minutes (3.4%) for hot start-up and 11 minutes (5.8%) for cold start-up. The simulation results are a little faster than the actual plant operation data, because the simulation model neglected the sequence control of some drain and vent valves to reduce the computation time.

2.5 Optimization of the plant operation

2.5.1 Optimization method

Nonlinear optimization problem is generally written as follows:

$$\min_{\mathbf{x}} f(\mathbf{x}), \quad (2.24)$$

$$\text{subject to } \begin{cases} g_{\ell}(\mathbf{x}) = 0, & \ell = 1, 2, \dots, s_e, \\ g_{\ell}(\mathbf{x}) \geq 0, & \ell = s_e + 1, s_e + 2, \dots, s, \end{cases} \quad (2.25)$$

where objective function $f(\mathbf{x})$ and constraints $g_{\ell}(\mathbf{x})$ depend on various values of vector parameter \mathbf{x} , s_e is number of equality constraints, and s is number of equality and inequality constraints.

The vector parameter \mathbf{x} is represented by transpose of a matrix of all the parameters x_1, x_2, \dots, x_n such as

$$\mathbf{x} = \left[x_1, x_2, \dots, x_n \right]^T, \quad (2.26)$$

where n is number of parameters.

The operational procedures of the combined cycle power plant can be formulated on the optimization problem which searches minimum starting time and/or fuel consumption rate during the start-up/shutdown under the various operational constraints (Shirakawa and Nakamoto, 2002).

2.5.1.1 Parameters

The operational parameters are control input during the plant operation (e.g., loading rate of the gas turbine).

2.5.1.2 Objective functions

The operational performance of the power plant can be evaluated in two ways as described here:

1. Using a function representing the plant state (e.g., starting time of the plant).

$$f_i(\mathbf{x}), \quad i = 1, 2, \dots, r_e. \quad (2.27)$$

2. Using a function expressing the transition from one state to another (e.g., fuel consumption rate during the start-up/shutdown).

$$f_i(\mathbf{x}) = \int_{t_1}^{t_2} f_{pv,i}(\mathbf{x}, t) dt, \quad i = r_e + 1, r_e + 2, \dots, r, \quad (2.28)$$

where r_e is number of functions f_i in Eq. (2.27), and r is number of all functions f_i in Eqs. (2.27) and (2.28). For example, t_1 is time of the ignition, and t_2 is time of reached at the base-load operation.

The objective function can be represented by a function such as

$$f(\mathbf{x}) = f(f_1, f_2, \dots, f_r). \quad (2.29)$$

Functions f_i (and the process values $f_{pv,i}$) in Eqs. (2.27) and (2.28) are determined by the dynamic simulation carried out under various values of vector parameter \mathbf{x} .

2.5.1.3 Constraints

Under the operational constraints, there are design limitation values of the equipment and environmental regulation values (e.g., the former is thermal stress in the steam turbine rotor and the HP drum temperature gradient, the latter is NO_x emission rate from the plant). The constraints are written as follows:

1. Using a function representing the maximum process value (e.g., maximum value of thermal stress in the steam turbine rotor and the HP drum temperature gradient).

$$g_{\ell}(\mathbf{x}) = 1 - \frac{\max_{t_1 \leq t \leq t_2} |g_{pv, \ell}(\mathbf{x}, t)|}{g_{c, \ell}} \geq 0, \quad \ell = 1, 2, \dots, s_e. \quad (2.30)$$

2. Using a function expressing the transition from one state to another (e.g., quantity of NO_x emission rate from the plant during the start-up/shutdown).

$$g_{\ell}(\mathbf{x}) = 1 - \frac{\int_{t_1}^{t_2} g_{pv, \ell}(\mathbf{x}, t) dt}{g_{c, \ell}} \geq 0, \quad \ell = s_e + 1, s_e + 2, \dots, s, \quad (2.31)$$

where $g_{c, \ell}$ are the limitations, s_e is number of functions g_{ℓ} in Eq. (2.30), and s is number of all functions g_{ℓ} in Eqs. (2.30) and (2.31). The process values $g_{pv, \ell}$ in Eqs. (2.30) and (2.31) are determined by the dynamic simulation carried out under various values of vector parameter \mathbf{x} .

In order to obtain a feasible solution, each parameter is limited as follows:

$$x_{L, j} \leq x_j \leq x_{U, j}, \quad j = 1, 2, \dots, n, \quad (2.32)$$

where x_L is the lower bound on x , and x_U is the upper bound on x .

2.5.1.4 Numerical algorithm

A calculation flowchart is shown in Fig. 2.5. Sequential quadratic programming (SQP) is introduced to the optimization procedure. This method is a type of nonlinear programming using the value of the gradient. The gradient of the objective function and the operational constraints are approximated by finite differences. The objective function, the operational constraints, and the gradients are calculated from the dynamic simulation.

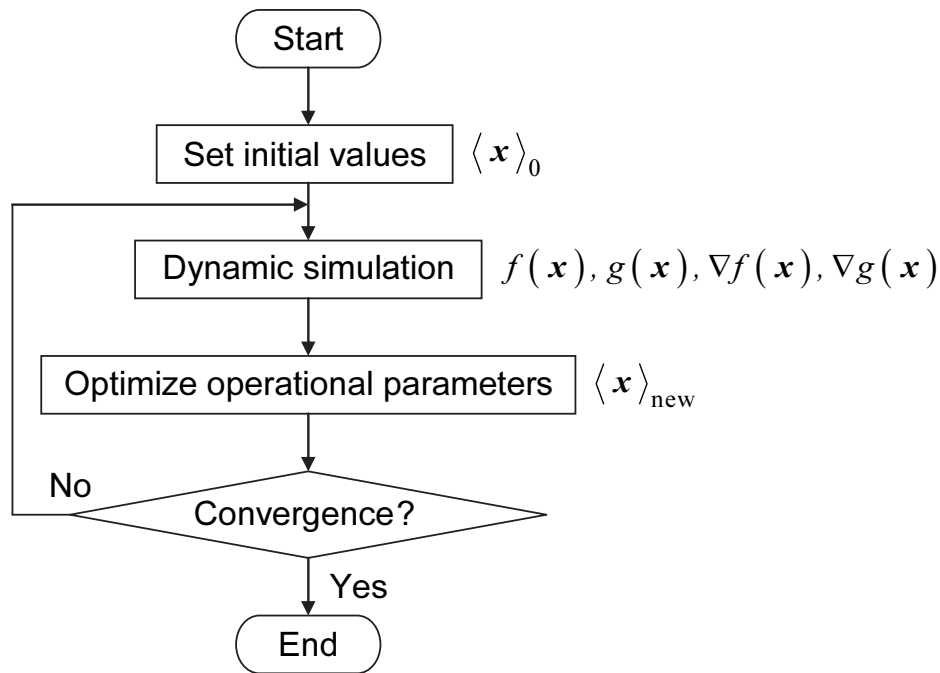


Fig. 2.5 Calculation flowchart which combined the dynamic simulation and the nonlinear programming, SQP.

2.5.2 Application example

The proposed method is applied to the optimization of the hot start-up process for a multi-shaft type combined cycle power plant (see Fig. 4.1). The firing temperature of the gas turbine is 1300°C class and total plant output is 670 MW.

The thermal stresses are generated by the rise of the exhaust gas temperature during the gas turbine loading. The operational parameters are shown in Fig. 2.6, that is, the control input of eight pieces of the gas turbine ramp rate is optimized. The limitation value of each ramp rate is decided in consideration of the gas turbine's design criteria based on the surge margin and stress limits.

$$0.1 \text{ \%/min} \leq x_j \leq 9.0 \text{ \%/min}, \quad j = 1, 2, \dots, 8. \quad (2.33)$$

The objective function combines the starting time with fuel consumption rate. The operational constraints are the thermal stress in the steam turbine rotor, the HP drum temperature gradient, and the NO_x emission rate from the plant. In this case, the NO_x emission rate had an enough margin for the limitation value.

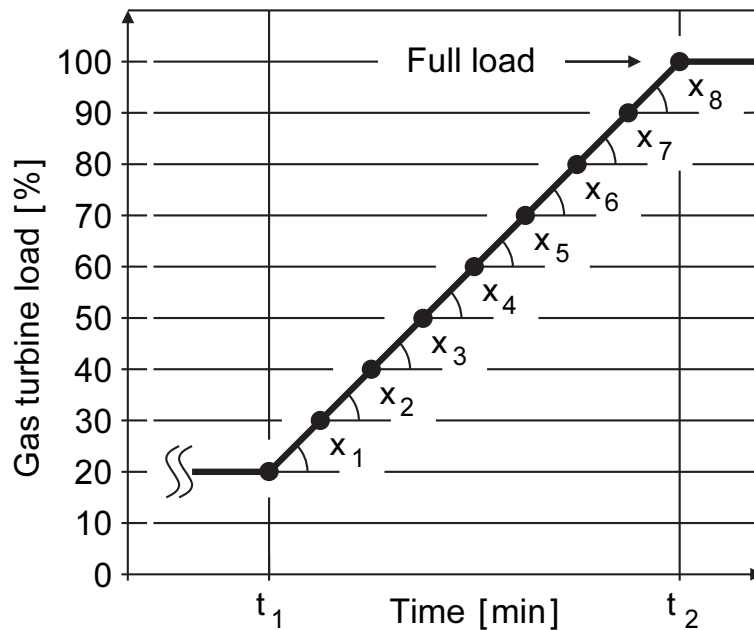


Fig. 2.6 Operational parameters (loading rates of the gas turbine). The control input of eight pieces of the gas turbine ramp rate is optimized.

The results of conventional heuristic approach are shown in Fig. 2.7. This approach has been decided using only the dynamic simulation by the engineer's trial and error. The results of this proposed method are shown in Fig. 2.8. This method can be automatically obtained by optimal operational parameters. In these figures, GT is the gas turbine and ST is the steam turbine. The proposed method reduces the start-up time by 6.0 minutes (21.7%) compared to the conventional method and satisfies the operational constraints.

2.6 Conclusions

An optimal design method for the plant system and a control design for the combined cycle power plant are proposed. It is possible to determine the optimal operational parameters by taking into account the dynamic characteristics of the plant.

The results of this study are summarized as follows:

1. The mathematical models of the combined cycle power plant were derived. These models were verified using actual plant operation data. The start-up simulation results were in good agreement with the plant operation data.

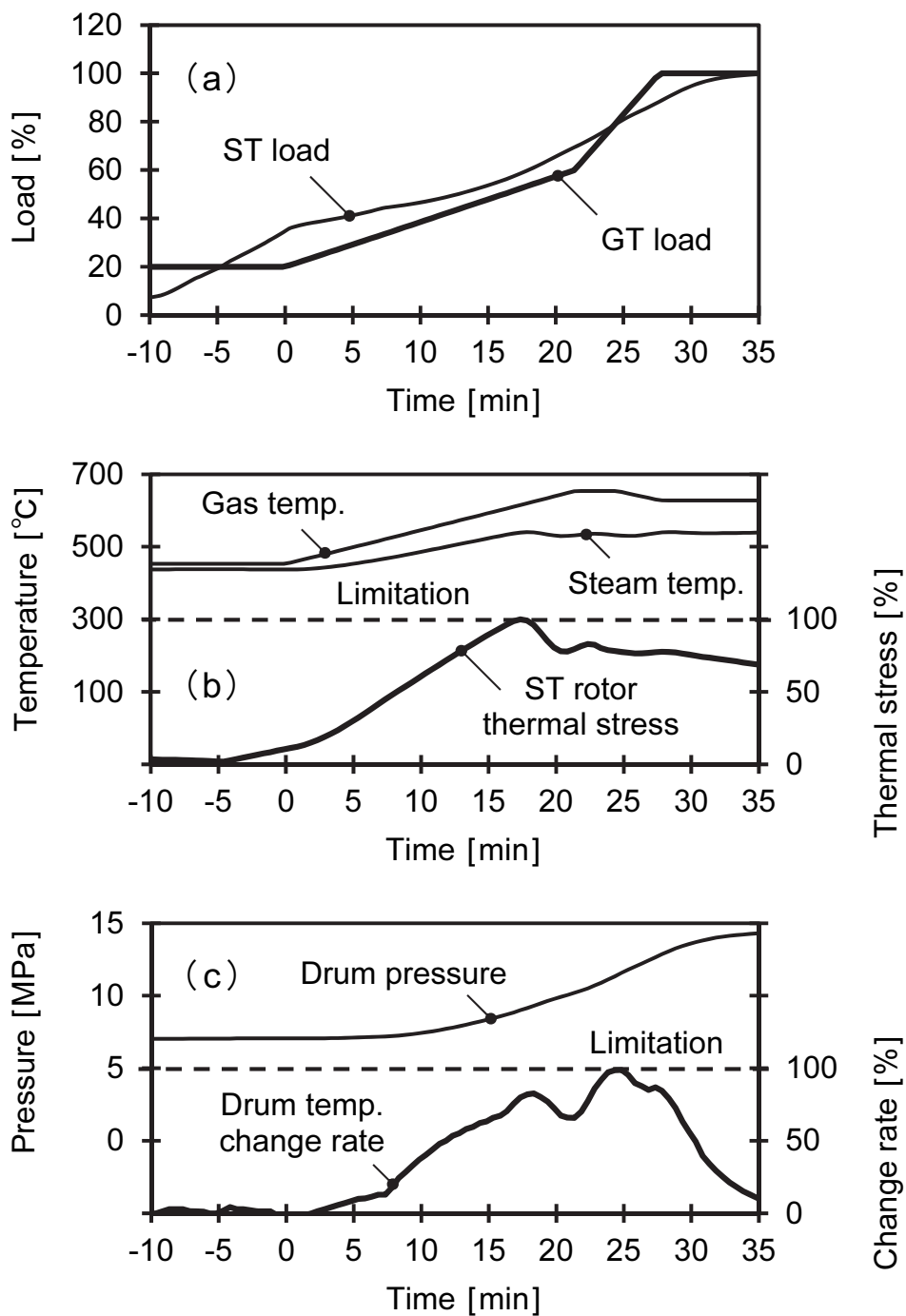


Fig. 2.7 Dynamic simulation results of conventional heuristic approach. (a) Gas turbine and steam turbine loads. (b) Gas turbine exhaust gas temperature, HP steam temperature, and thermal stress of the steam turbine rotor. (c) HP drum pressure and HP drum temperature gradient.

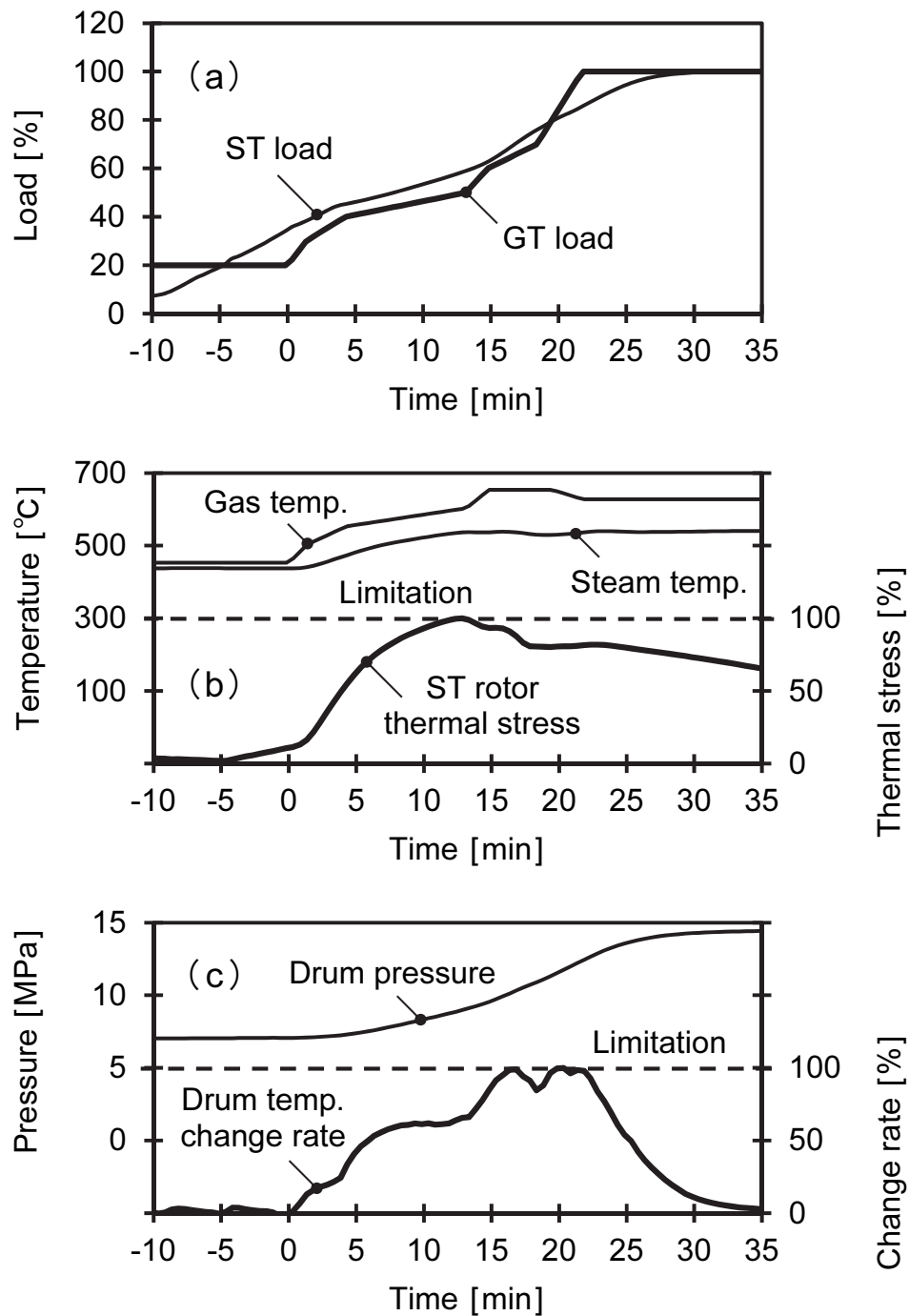


Fig. 2.8 Dynamic simulation results of the proposed method (optimal solution). (a) Gas turbine and steam turbine loads. (b) Gas turbine exhaust gas temperature, HP steam temperature, and thermal stress of the steam turbine rotor. (c) HP drum pressure and HP drum temperature gradient.

2. We formulized the plant operation to the optimization problem, and performed start-up optimization of a validated combined cycle power plant model using the nonlinear programming, SQP (sequential quadratic programming).

Finally, some issues that will be considered in the following chapters are as follows:

1. In order to optimize the plant operation and control system design of power plants, it is desirable for design engineers or operators, rather than analytical specialists, to perform dynamic simulations themselves. Therefore, easy-to-use dynamic simulation tools need to be developed for them.
2. Since the evaluation functions (objective and constraint functions) in the mathematical models cannot be expressed explicitly in terms of design variables (operational parameters), the values of the evaluation functions are determined by performing dynamic simulations. Usually, obtaining the values of evaluation functions by performing dynamic simulations is considerably time consuming. In addition, the optimization of the operation of a thermal power plant should be a multi-objective optimization problem that takes into account various evaluation indicators. In cases with multi-objective functions, it is extremely difficult to adjust the weights of each objective function. Moreover, the computing time increases drastically, making applications impossible. Therefore, the devised multi-objective optimization method is required to reduce the number of dynamic simulations as much as possible.

The above two issues are addressed in Chapter 3 of Part I and Chapter 4 of Part II, respectively.

Appendices

Notation

<i>A</i>	heat transfer area [m^2]
<i>C</i>	specific heat [$\text{kJ}/(\text{kg} \cdot ^\circ\text{C})$]
<i>E</i>	Young's modulus [N/mm^2]
<i>G</i>	mass flow rate [kg/s]
<i>h</i>	specific enthalpy [kJ/kg]
<i>k</i>	heat transfer coefficient [$\text{kW}/(\text{m}^2 \cdot ^\circ\text{C})$]
<i>L</i>	evaporating point [pu]

M	mass [kg]
P	pressure [kPa]
Q	heat flow [kW]
r	radial axis [m]
R_i	inner radius [m]
R_o	outer radius [m]
S	section area [m ²]
t	time [s]
T	temperature [°C]
v	specific volume [m ³ /kg]
W	power [kW]
x	quality [pu]
α_ℓ	coefficient of linear expansion [1/°C]
ζ	flow coefficient [-]
$\zeta(z)$	flow coefficient depending on valve position z [-]
η	adiabatic efficiency of steam turbine [pu]
θ	rotor temperature [°C]
λ	thermal conductivity [kW/(m · °C)]
μ	Poisson's ratio [-]
ρ	density [kg/m ³]
σ_{zz}	thermal stress [N/mm ²]

Subscripts

bo	drum blow
c	circulation
cd	condensation in drum
dm	drum metal
dv	evaporation in drum
dw	drum holding water
e	evaporator exit
esm	evaporator saturated section metal
eum	evaporator subcooled section metal
ev	evaporation in evaporator
f	interface

<i>g</i>	gas side
<i>in</i>	inlet
<i>m</i>	heat exchanger tube metal
<i>out</i>	outlet
<i>s</i>	saturated steam
<i>stm</i>	steam side
<i>sw</i>	saturated water
<i>tb</i>	turbine
<i>tbor</i>	ideal turbine exit at 100% efficiency
<i>w</i>	feed water
<i>wu</i>	subcooled water

References

- Akiyama, T., Matsushima, T., Nagafuchi, N., Nakajima, A., Yoshizaki, H. and Matsumura, J., Dynamic simulation of an advanced combined cycle plant with three pressure and reheat cycle, Proceedings of the JSME-ASME International Conference on Power Engineering-93 (ICOPE-93), The Japan Society of Mechanical Engineers, Vol. 1, (1993), pp. 221–226, Tokyo, Japan.
- Kamiya, A., Kawai, K., Ono, I. and Kobayashi, S., Adaptive-edge search for power plant start-up scheduling, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 29, No. 4, (1999), pp. 518–530.
- Mastronarde, T. P. and Shiue, Y. L., Transient behavior of forced circulation and natural circulation heat recovery steam generators, Proceedings of the 1995 ASME Cogen-Turbo Power Conference, Paper number 95-CTP-98, (1995), Vienna, Austria.
- Matsumoto, H., Ohsawa, Y., Takahashi, S., Akiyama, T. and Ishiguro, O., An expert system for start-up optimization of combined cycle power plants under NO_x emission regulation and machine life management, IEEE Transactions on Energy Conversion, Vol. 11, No. 2, (1996), pp. 414–422.
- Matsumoto, H., Ohsawa, Y., Takahashi, S., Akiyama, T., Hanaoka, H. and Ishiguro, O., Startup optimization of a combined cycle power plant based on cooperative fuzzy reasoning and a neural network, IEEE Transactions on Energy Conversion, Vol. 12, No. 1, (1997), pp. 51–59.

Shirakawa, M. and Nakamoto, M., Dynamic optimization method for start-up scheduling of a combined cycle power plant, Transactions of the Japan Society of Mechanical Engineers, Series B, Vol. 68, No. 672, (2002), pp. 2377–2384 (in Japanese).

Shirakawa, M., Dynamic simulation of heat recovery steam generations for combined cycle power plants (Dynamic modeling and characterization tests of evaporators), Transactions of the Japan Society of Mechanical Engineers, Series B, Vol. 69, No. 686, (2003), pp. 2355–2362 (in Japanese).

Chapter 3

Development of a thermal power plant simulation tool based on object orientation

Abstract

This study presents a new dynamic simulation tool that facilitates plant and control system design of thermal power plants. In order to provide flexibility in building plant and control systems, object-oriented modeling techniques are used to model individual plant components and each component object model is based on the verified first principles model. Further, simulation models can be easily built by connecting individual component object models, similar to the procedure adopted in an actual plant, by using a user-friendly graphical user interface. This tool enables the design engineer to build a simulation model, execute the simulation during the design process, test an actual plant control system, and adjust the control parameters. It is a useful design tool that accelerates and enhances the plant and control system design process.

Keywords

Combined cycle, Control design, Dynamic simulation, Hardware-in-the-loop simulation, Model-based design, Modeling, Object orientation, Object-oriented modeling, Power plant, Systems engineering

3.1 Introduction

In the past, dynamic simulations have been used to evaluate the operating characteristics of a thermal power plant (Maffezzoni, 1992; Ordys et al., 1994; Leva and Maffezzoni, 2003). Now, as described in Chapter 2, after carrying out repeated verifications and improvements of existing plant simulation models by using actual plant operation data, a simulation technology that can quantitatively evaluate most of the dynamic phenomena has been established.

This simulation method can be used for two main purposes—compatibility

verification of the design specifications and phenomenal investigation of the design issues. From the viewpoint of accelerating and enhancing the product design process, it is considered that it is more effective if the former operation is performed by a product design engineer and the latter is essentially performed by an analytical specialist. However, a dynamic simulation tool that provides simplified operation conditions to design engineers is required in order to incorporate simulation together with phenomenal investigation into the design process.

Until now, there have been many reports with regard to dynamic modeling (Ahluwalia and Domenichini, 1990; Akiyama et al., 1993; Mansour et al., 2003) and simulation tools (Shor, 1987; Hashemi et al., 1988; Bartolini et al., 1996; Akiyama et al., 1997; Leva et al., 1998; Elmegaard, 1999; Faille et al., 2003; Casella and Leva, 2006) for thermal power plants. However, the general usage of these tools requires the design engineers, who may not be analytical specialists of either dynamic modeling or simulation tools, to possess extensive expertise. Moreover, many of these tools are based on modularized programming, and they require various special rules in order to simulate a large-scale system such as a thermal power plant; this could create problems with regard to the general versatility, function expansion, and maintainability. Recently, object-oriented techniques have drawn considerable attention in the field of software development (Taylor, 1991). Lu has proposed a thermal power plant model based on object orientation (Lu et al., 1995, 1996; Lu and Hogg, 1996; Lu, 1999). The proposed method facilitates a natural method of modeling in which the mechanical information on an actual plant is replicated, thereby providing simplified operation conditions for design engineers and serving as an excellent design tool in terms of the general versatility, function expansion, and maintainability.

Hence, in this study, the author has extended Lu's method by detailing and expanding the functions and developed a thermal power plant simulation tool on the basis of object orientation, which can be easily operated by design engineers. This study describes the development environment, the modeling method and the features in the tool use, the application results, and the virtual design environment with the hardware-less simulation system.

3.2 Development environment

The dynamic simulation tool works on a personal computer (PC), thereby providing

simplified operation conditions to the design engineer, and it was implemented in MATLAB/Simulink—a widely used package software created by The MathWorks, Inc.

The reasons for using MATLAB/Simulink are as follows.

1. Unlike programming languages, it provides a modeling and simulation environment that is equipped with a graphical user interface (GUI), thereby allowing the development of a dynamic simulation tool that can be easily operated by design engineers.
2. It can create control models necessary for plant simulation. It can also develop a control logic diagram model equivalent to an actual plant control system, thereby enhancing the analytical accuracy and simulation performance.
3. It offers abundant mathematical function libraries, numerical analyses, and graphic functions, thus reducing the development load in all areas other than modeling.
4. The layering function provides easy view ability and efficient usability, which are appropriate for modeling large-scale systems such as thermal power plants.
5. Owing to the strong connectivity to an actual plant control system, each phase from the plant system and control design phases to the phase of testing an actual plant control system (called pre-delivery factory test) and trial run adjustments can be performed in a consistent simulation environment.

3.3 Modeling method

The developed dynamic simulation tool has the ability to analyze conventional (simple-steam-cycle) thermal power generation systems and combined-cycle power generation systems. The author has developed a high-accuracy plant simulation model, as described in Chapter 2. The model is derived from first principles (mass, energy, and momentum conservation equations) and verified by using actual plant operation data. In this study, it is applied to the plant model of a dynamic simulation tool; thus, it can achieve an analytical accuracy and simulation performance equivalent to or greater than that of a model created using conventional programming languages (e.g., Fortran and C/C++). The features of the modeling method are described subsequently.

3.3.1 Object-oriented model and internal state quantity calculation

In order to provide a flexible plant configuration, the object-oriented modeling approach is adopted (Shirakawa et al., 2005). Figure 3.1 shows the concept of an

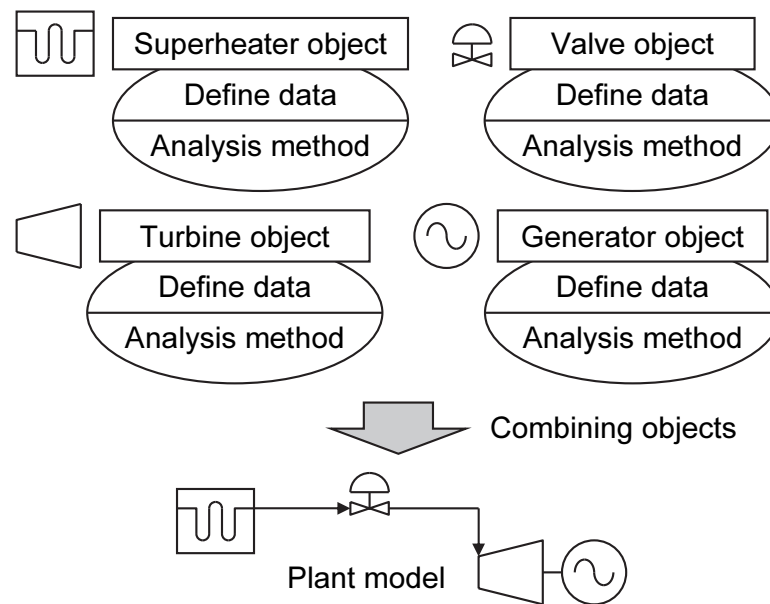


Fig. 3.1 Concept of object-oriented model. Dynamic simulation models can be easily built by connecting individual component object models.

object-oriented model. In this model, each component equipment model of the plant (heat exchanger, valve, steam turbine, and generator) forms an independent object, and the whole plant model is built by combining these independent objects. The equipment model for each object is defined as a combination of the physical model, which represents its individual behavior and related data (equipment data such as size and weight, and state quantities such as pressure, temperature, and flow rate).

Given the passage state quantity at a certain time, each object time integrates its governing equation and performs a procedure (called “method” in object orientation) to obtain the subsequent internal passage state quantity. Then, the information on the outlet state quantity (the internal state quantity) is successively transferred as a parameter to the downstream object, and this downstream object is requested to execute its method; this request is termed as a “message” in object orientation. By executing this operation sequentially, the time response of the whole plant model can be calculated. Each equipment model is formed self-sufficiently, allowing the object-oriented model to flexibly correspond to the plant configuration. Moreover, equipment models can also be changed and added independent of others, thus making this method superior in terms of the general versatility, function expansion, and maintainability.

An example of the governing equation of the heat exchanger (superheater) object (in the cases of counter flow and convective heat transfer) and the related data is shown subsequently. Here, the heat transfer surface is multiply segmented, and is converted into the multiple segment lumped-parameter model centralizing to each segment. Figure 3.2 depicts the notation at an arbitrary segment surface.

Heat transfer outside the tube:

$$Q_{g,j} = G_g C_g (T_{g,j} - T_{g,j+1}) \cong \alpha_{g,j} G_g^X \Delta\theta_{g,j} . \quad (3.1)$$

Logarithmic mean temperature difference (gas side):

$$\Delta\theta_{g,j} = \frac{(T_{g,j} - T_{m,j}) - (T_{g,j+1} - T_{m,j})}{\ln\left(\frac{T_{g,j} - T_{m,j}}{T_{g,j+1} - T_{m,j}}\right)} . \quad (3.2)$$

Heat transfer inside the tube:

$$Q_{stm,i} = G_{stm} C_{stm} (T_{stm,i+1} - T_{stm,i}) \cong \alpha_{stm,i} G_{stm}^Y \Delta\theta_{stm,i} . \quad (3.3)$$

Logarithmic mean temperature difference (steam side):

$$\Delta\theta_{stm,i} = \frac{(T_{m,j} - T_{stm,i}) - (T_{m,j} - T_{stm,i+1})}{\ln\left(\frac{T_{m,j} - T_{stm,i}}{T_{m,j} - T_{stm,i+1}}\right)} . \quad (3.4)$$

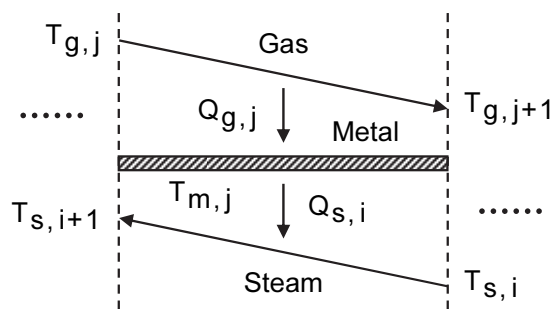


Fig. 3.2 Notation at an arbitrary segment surface (superheater model). This figure shows the cases of counter flow and convective heat transfer.

Energy balance for the heat exchanger tube metal:

$$\frac{d}{dt} T_{m,j} = \frac{Q_{g,j} - Q_{stm,i}}{M_{m,j} C_m}, \quad \text{where } M_{m,j} = \frac{M_m}{n}. \quad (3.5)$$

The outlet gas temperature can be obtained from Eqs. (3.1) and (3.2):

$$T_{g,j+1} = T_{m,j} - (T_{m,j} - T_{g,j}) \exp\left(-\frac{\alpha_{g,j} G_g^X}{G_g C_g}\right). \quad (3.6)$$

The outlet steam temperature can be obtained from Eqs. (3.3) and (3.4):

$$T_{stm,i+1} = T_{m,j} - (T_{m,j} - T_{stm,i}) \exp\left(-\frac{\alpha_{stm,i} G_{stm}^Y}{G_{stm} C_{stm}}\right). \quad (3.7)$$

Steam flow rate:

$$G_{stm} = \zeta S \sqrt{\frac{(P_{stm,in} - P_{stm,out}) \times 10^3}{v_{stm,in}}}. \quad (3.8)$$

This model requires equipment data such as the mass of the heat exchanger tube metal, the heat transfer coefficient outside and inside of the tube, and the flow coefficient. It also requires the initial value of the state quantity (initial temperature distribution of the heat exchanger tube metal) for the time integration of Eq. (3.5). When the inlet gas temperature, gas flow rate, inlet steam temperature, and inlet and outlet steam pressures are given, the outlet gas temperature, outlet steam temperature, and steam flow rate can be obtained from Eqs. (3.5) to (3.8).

3.3.2 Automatic parameter setting

In the dynamic simulation of a large-scale system such as a thermal power plant, it is necessary to set several equipment data and the initial values of the state quantities, which vary for each plant. Many of these parameters such as the heat transfer coefficient, the flow coefficient, and the steam mass inside the drum are not readily available; therefore, they require a format so that they can be fitted in the plant model. Therefore, in the past, analytical specialists well informed of the plant model have had

to separately calculate the parameters for the simulation and set them manually, thus reducing work efficiency and causing obstructions to communication with the design engineers.

This plant simulation method calculates the dynamic characteristics from the equations of mass, energy, and momentum conservation, including their time differentiations, on the basis of the assumption that the simulated plant is in accordance with the static characteristic represented by the results of heat balance calculation. Two types of equipment data exist—dynamic data (e.g., weight and volume), which depend on the time response of the plant, and static data (e.g., heat transfer coefficient and flow coefficient), which depend on the steady-state characteristics of the plant. The dynamic data are assigned estimated values derived from the previous plant results obtained at the beginning of design process; subsequently, these data are replaced with accurate data in accordance with the design progress. On the other hand, static data can be obtained by reverse calculating from the results of the heat balance calculation (pressure, temperature, flow rate, etc.).

The following example shows the application of the reverse calculation method to the governing equation of the heat exchanger (superheater) object described in Section 3.3.1 (in the cases of counter flow and convective heat transfer).

Heat flow:

$$Q_g = Q_{stm} = G_{stm} (h_{stm, out} - h_{stm, in}), \quad (3.9)$$

where the specific enthalpy can be calculated from the pressure and temperature using steam tables

$$\begin{cases} h_{stm, in} = h(P_{stm, in}, T_{stm, in}), \\ h_{stm, out} = h(P_{stm, out}, T_{stm, out}). \end{cases} \quad (3.10)$$

Logarithmic mean temperature difference (gas side):

$$\Delta\theta_g = \frac{(T_{g, in} - T_{m, in}) - (T_{g, out} - T_{m, out})}{\ln\left(\frac{T_{g, in} - T_{m, in}}{T_{g, out} - T_{m, out}}\right)}. \quad (3.11)$$

Logarithmic mean temperature difference (steam side):

$$\Delta\theta_{stm} = \frac{(T_{m, out} - T_{stm, in}) - (T_{m, in} - T_{stm, out})}{\ln\left(\frac{T_{m, out} - T_{stm, in}}{T_{m, in} - T_{stm, out}}\right)}, \quad (3.12)$$

where the temperature of the heat exchanger tube metal is assumed as follows

$$\begin{cases} T_{m, in} = \frac{T_{g, in} + T_{stm, out}}{2}, \\ T_{m, out} = \frac{T_{g, out} + T_{stm, in}}{2}. \end{cases} \quad (3.13)$$

Heat transfer coefficient outside the tube:

$$\alpha_g = \frac{Q_g}{G_g^X \Delta\theta_g}, \quad \text{where } \alpha_{g, j} = \frac{\alpha_g}{n}. \quad (3.14)$$

Heat transfer coefficient inside the tube:

$$\alpha_{stm} = \frac{Q_{stm}}{G_{stm}^Y \Delta\theta_{stm}}, \quad \text{where } \alpha_{stm, i} = \frac{\alpha_{stm}}{n}. \quad (3.15)$$

Flow coefficient:

$$\zeta = \frac{\sqrt{\frac{G_{stm}^2 v_{stm, in}}{(P_{stm, in} - P_{stm, out}) \times 10^3}}}{S}, \quad (3.16)$$

where the specific volume can be calculated from the pressure and temperature using steam tables

$$v_{stm, in} = v(P_{stm, in}, T_{stm, in}). \quad (3.17)$$

Initial temperature distribution of the heat exchanger tube metal:

$$\begin{cases} T_{m,1} = T_{m,in} - \frac{\Delta T_m}{2}, & j = 1, \\ T_{m,j} = T_{m,j-1} - \Delta T_m, & j = 2, 3, \dots, n, \end{cases} \quad (3.18)$$

$$\text{where } \Delta T_m = \frac{T_{m,in} - T_{m,out}}{n}. \quad (3.19)$$

Thus, given the inlet gas temperature, outlet gas temperature, gas flow rate, inlet steam pressure and temperature, outlet steam pressure and temperature, and steam flow rate from the results of the heat balance calculation, the equipment data such as the heat transfer coefficient outside and inside the tube, the flow coefficient, and the initial temperature distribution of the heat exchanger tube metal can be automatically obtained by reverse calculation.

This tool has a function of automatic parameter setting (equipment data and initial value of the state quantity) as the method of each object. When the simulation begins, these methods are executed. Therefore, each object only requires the results of the heat balance calculation (pressure, temperature, flow rate, etc.) corresponding to each object. From these results, each object calculates the data related to it and is able to automatically set the data for the whole plant model on the basis of the information on the outlet state quantity (internal state quantity) that is successively transferred as a parameter to the downstream object. This downstream object is requested to execute its method, and this request is called a “message” in object orientation. Hence, communication with the design engineers is enhanced because the method can substantially reduce the parameter setting operations for simulations that require enormous effort. Figure 3.3 shows an example of the concept of the heat exchanger (superheater) object. In this object, three methods for performing the calculations of the internal state quantity explained in Section 3.3.1, equipment data setting, and initial value setting of the state quantity are packaged along with the related data; this is termed as “encapsulation” in object orientation. Each method receives the necessary passage state quantities (inlet gas temperature, gas flow rate, inlet steam temperature, and inlet and outlet steam pressures) as a message from other objects. Then, the equipment data setting method calculates the equipment data (heat transfer coefficient outside and inside the tube, and flow coefficient) by applying the input values (outlet gas temperature, outlet steam temperature, and steam flow rate) of the heat balance calculation to its own object and from the value of the received message. Similarly, the

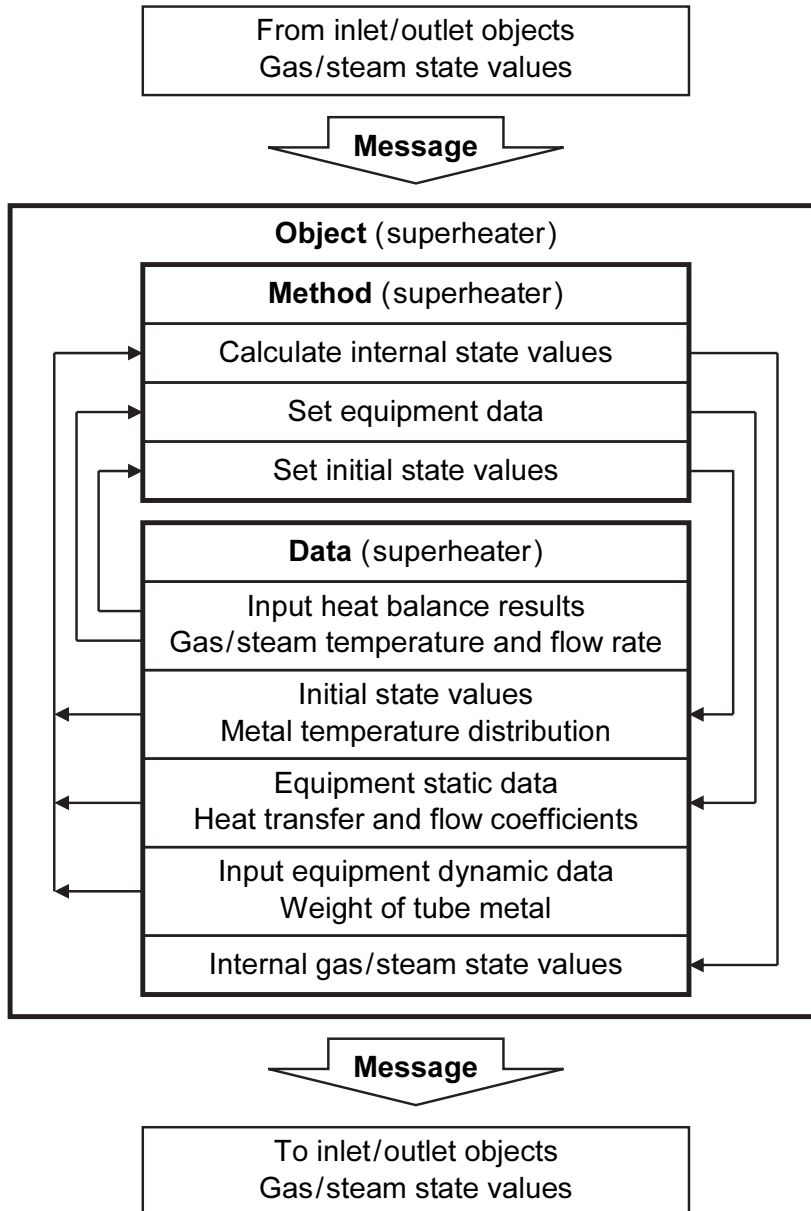


Fig. 3.3 Concept of the heat exchanger (superheater) object. The equipment data and the initial state values can be automatically obtained by reverse calculation.

method for setting the initial value of the state quantity calculates the initial temperature distribution of the heat exchanger tube metal. The method for the calculation of the internal state quantity (see Section 3.3.1) obtains the internal state quantity from the governing equation based on the component equipment data and the initial value of the state quantity.

3.3.3 Model library

In order to efficiently create an equipment model that is functionally similar to and can simulate other plants, a template with common governing equations and data variables, called a “class” in object orientation, is defined. Then, a concrete equipment model becomes an object that is provided with a specific data value (termed as an “instance” in object orientation) for the common governing equation. The thermal power plant is a compound object that comprises many equipment objects; this implies that a larger equipment model can be established by creating individual equipment objects and layering and combining these objects. In object orientation, information can be expressed in a commonly used natural form; this allows us to use the model without requiring the determination of the internal structure complexity.

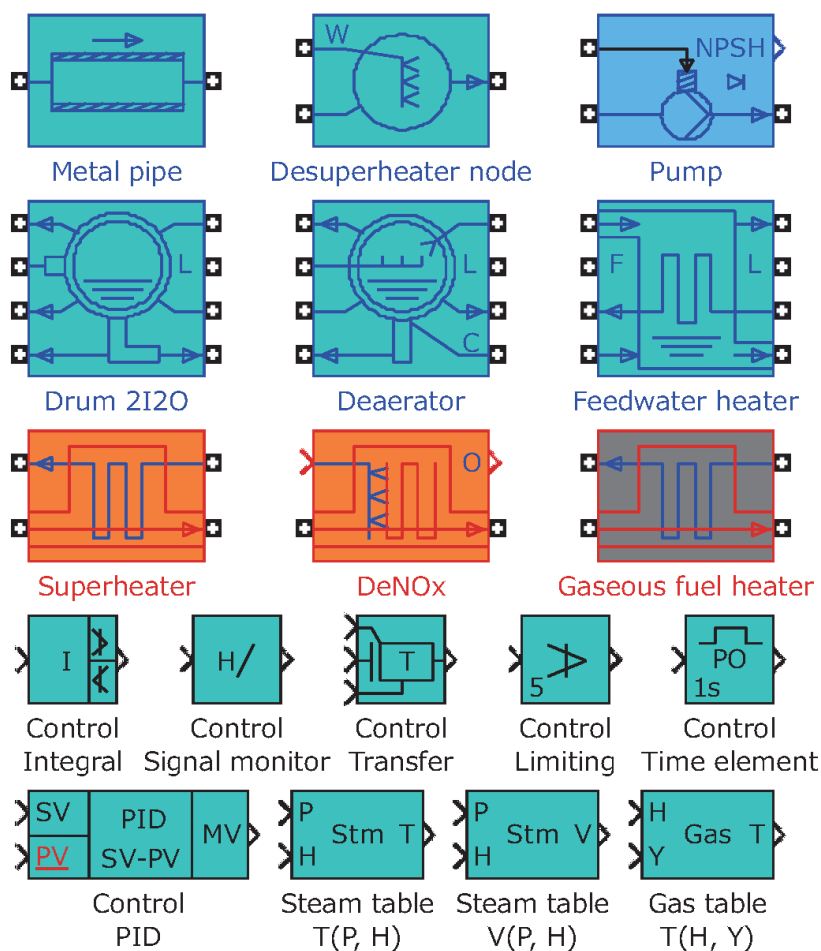


Fig. 3.4 Part of the development model library. The model library consists of equipment models, control models, steam tables, and gas tables.

The models as respective classes require to be registered in a model library so that they can be managed in an integrated fashion and recycled. The model library consists of 120 types of equipment models (gas turbine, steam turbine, burner, furnace, superheater, evaporator, economizer, drum, air preheater, mill, valve, pipe, pump, fan, sensor, etc.), 130 types of control models (PID controller, transfer, timer, etc.), and 100 types of steam tables and gas tables. In these models, a complex calculation is performed using a built-in function (called S-function) of MATLAB/Simulink by programming in C/C++, combining with the equipped mathematical models (mathematical function, integrator, regulator, upper and lower limiter, etc.) of Simulink. Figure 3.4 shows a part of the developed model library. Each model is represented by a unique icon so as to increase visibility and enlarge the data-setting screen (double-click each icon to enlarge the display).

A class consists of its own component equipment class, a concrete equipment class and a class hierarchy that comprises the abstractions (generalizations) of these equipment. (In object orientation, each component equipment class is called a “subclass” of an equipment class.) Each component equipment class is permitted to use the method and the variable of the equipment class (referred to as “inheritance” in object orientation). In short, by only sending a message corresponding to the methods for the internal state quantity calculation, equipment data setting, and initial setting of the state quantity, each unique method can be executed by applying it to each component equipment class (called “polymorphism” in object orientation).

3.3.4 Flow vector

Each object is connected by using the method of “flow vectors”. In this method, the state quantity of the operating fluid is bundled in the flow direction. The water and steam systems are connected by a flow vector in which the pressure, specific enthalpy, and flow rate are bundled. When the temperature, specific volume, specific entropy, etc. are required for calculation in an object, these values are calculated from the pressure and the specific enthalpy by using steam tables. Similarly, the air and gas systems are connected by a flow vector in which the pressure, specific enthalpy, flow rate, and gas composition are bundled. When the temperature, specific heat, etc. are required for calculation in an object, these values are calculated from the specific enthalpy and the gas composition by using gas tables.

The flow vector can connect different objects, similar to the function of process flow

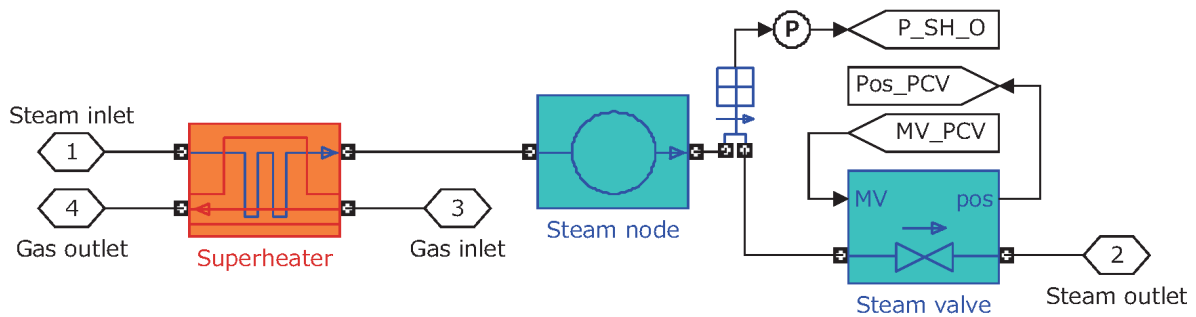


Fig. 3.5 Flow vector linking of the object models. The flow vector can connect different objects, similar to the function of process flow in an actual plant.

in an actual plant. Therefore, the plant model can be easily created by connecting each equipment model in accordance with the plant system diagram. The flow vector is constructed in the form of connecting lines with Two-Way Connection block of Simscape in Simulink's library. This block has a two-way connector port, which transports Simulink signals both ways. Figure 3.5 shows an example of a plant model in which the steam pressure node, pressure sensor, and pressure control valve (PCV) are connected by the flow vector on the downstream side of the heat exchanger (superheater). The signal tags P_SH_O, MV_PCV, and Pos_PCV represent the pressure sensor value, PCV opening command value, and PCV real opening value, respectively, and they also transfer the signals to each controller. Thus, the flow vector enables the creation of a comprehensive plant model with excellent visibility. Without the flow vector, each variable would require to be connected with numerous arrows, which would make the model confusing.

3.4 Features in the tool use

The GUI and real-time calculation are two major features of this tool. Their descriptions are provided subsequently.

3.4.1 GUI

The tool facilitates the creation and execution of simulation models and the display and printing of results through a GUI on Simulink.

An object-oriented model (see Section 3.3.1) maintains its own behavior for each equipment model and data; this enables the establishment of plant models with flexible plant compositions that can be obtained by only changing the connecting relationships among the equipment models. This operation is similar to that of a computer-aided design (CAD) tool. On the PC screen, the necessary equipment models can be pasted from the model library by using the mouse (see Section 3.3.3); this allows for the individual models to be connected in accordance with the plant system diagram of an actual plant. Different equipment models can be connected by using the flow vector in a manner similar to that of process flow in an actual plant (see Section 3.3.4), thus allowing the whole plant model to be easily and efficiently constructed.

After completing the construction of the whole plant model, the results of the heat balance calculation for each equipment model are entered in the data-setting screen. Then, the simulation is executed and the equipment data and the initial value of the state quantity are set by automatic parameter setting (see Section 3.3.2). The simulation execution only requires the initial value of the state quantity to be reconfigured because the equipment data remains unchanged.

Even while executing the simulation, the results can be displayed and printed, and the parameters can be changed. Additionally, the calculation can be aborted, paused, and resumed. Therefore, the adjustment of the control parameters and the equipment sizing are efficiently promoted.

3.4.2 Real-time calculation function

The calculation speed of this tool depends on the plant scale (the scale of modeling) and the PC performance. However, in recent years, the substantial improvement in the calculation performance offers a high-speed calculation that is significantly faster than real-time calculation, thereby allowing us to efficiently promote the design study during the design phase. Moreover, this tool can easily switch to a real-time plant simulation by using GUI. The plant model simulator with a good accuracy, which is used in the dynamic simulation for the design study, can be used as it is—as a real-time plant simulator connected to an actual plant control system. Hence, the dynamic simulation tool is unified from the plant system and control design phase to the testing hardware-in-the-loop simulation of the actual plant control system and the trial run adjustment process.

3.5 Application results

This study considers a combined-cycle power generation system. This system comprises a gas turbine, a heat recovery steam generator (HRSG), a steam turbine, and a generator. The gas and steam turbines drive the generator. Further, the HRSG generates steam for the steam turbine by using the waste heat generated from the gas turbine.

3.5.1 Verification of the dynamic simulation tool

The actual plant operation data were measured from a one-shaft type and triple pressure with reheat cycle (see Fig. 2.1). This plant generates a total plant output of 340 MW. The firing temperature of the gas turbine was 1300°C class. The comparison of warm start-up is shown in Fig. 3.6. Figure 3.6a shows the generator speed and output response. Figure 3.6b shows the gas turbine exhaust gas temperature and flow rate response. Figure 3.6c shows the high-pressure steam temperature, pressure, and flow rate response. Figure 3.6d shows the first-stage shell metal temperature and the rotor thermal stress of the steam turbine response. In Fig. 3.6, the lines traced by symbols are the plant operation data and the solid lines are the simulation results. The simulation results are in good agreement with the plant operation data. These results indicate that this tool can be used in plant and control system design.

3.5.2 Simulation example

The combined-cycle power generation system with a total plant output of 265 MW was designed. The firing temperature of the gas turbine was 1300°C class. Figure 3.7 shows the plant system diagram. In Fig. 3.7, SH denotes the superheater; EVA, the evaporator; ECO, the economizer; and DEA, the deaerator. The whole plant model and the controller model were built using this tool.

Figure 3.8 shows a part of the screen display of the whole plant model. Here, the object models represented by the icons are the main component equipment. The flow vectors shown by the form of connecting lines represent the operating fluid in the water/steam piping and the gas duct. Thus, this tool allows a comprehensive simulation model to be built in accordance with the plant system diagram. If the plant composition is changed, only the connecting relationships and the data on the related object models need to be reconfigured.

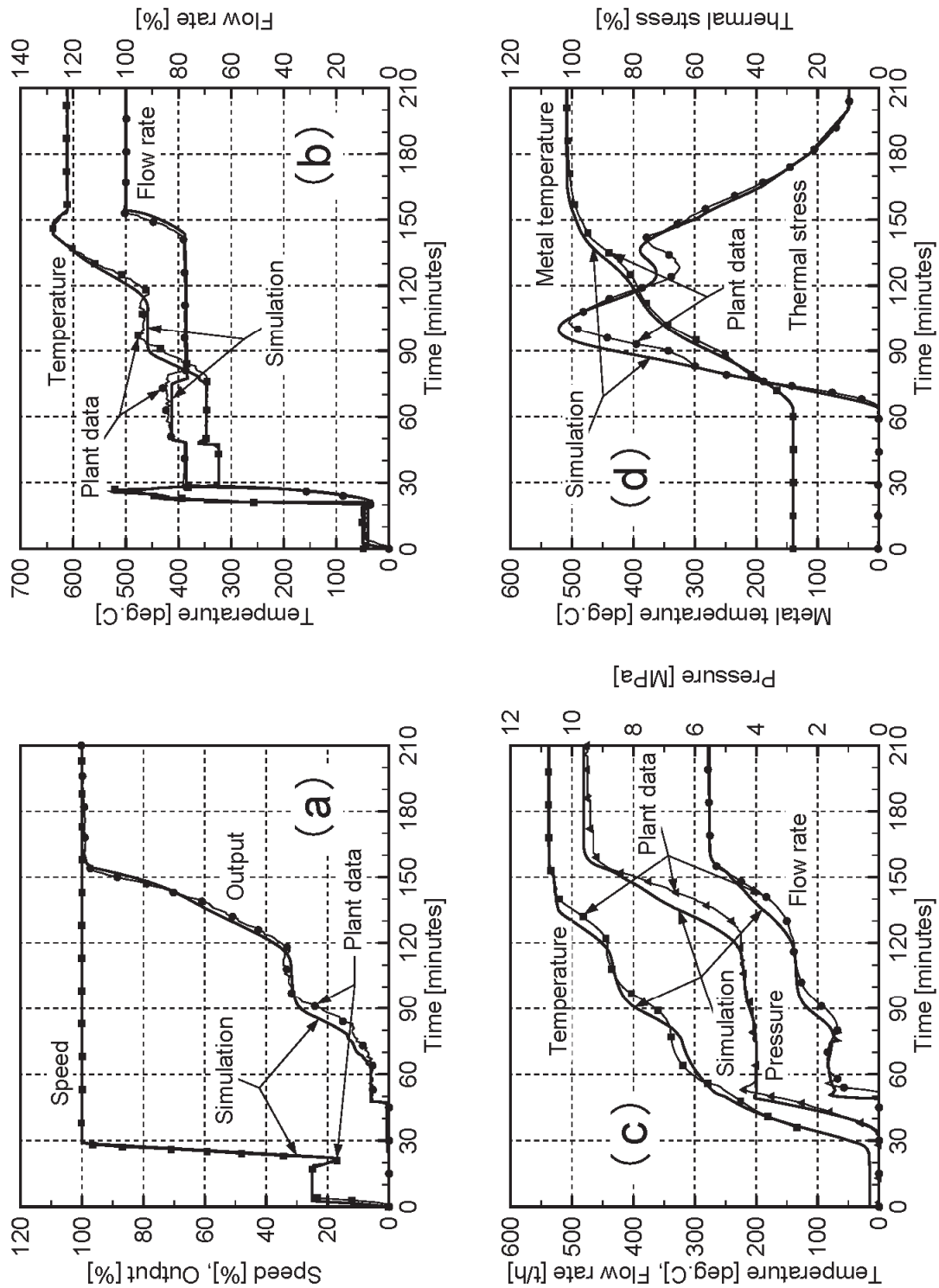


Fig. 3.6 Comparative results of simulation and actual plant operation data for warm start-up. The symbols lines are the plant operation data and the solid lines are the simulation results. (a) Generator response. (b) Gas turbine exhaust gas response. (c) High-pressure steam response. (d) Thermal stress of the steam turbine response.

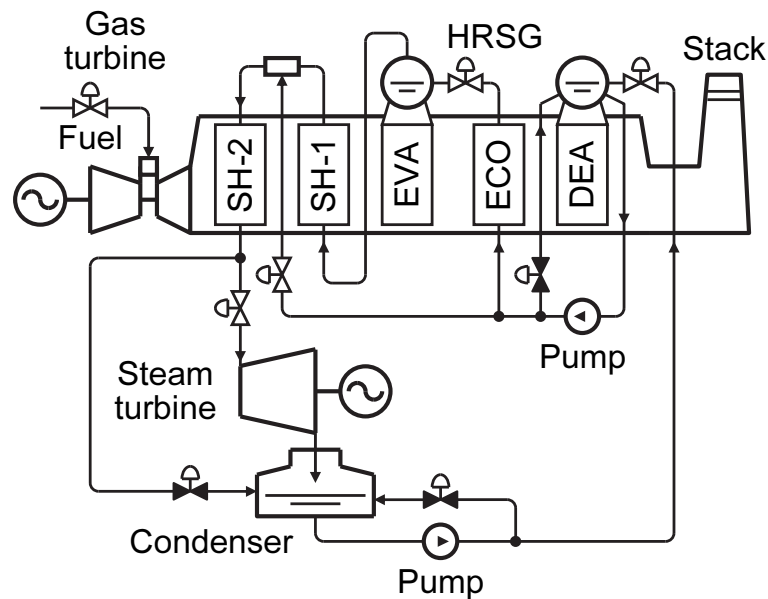


Fig. 3.7 Schematic diagram of combined-cycle power generation system. The system configuration is double shaft with separate generators for gas turbine and steam turbine.

The controller model accurately simulates the control logic diagram of an actual plant control system. Various operating statuses can be simulated by transferring the sensor value and the valve-opening command value signals between the whole plant model and the controller model. As an example, Fig. 3.9 shows the simulation result for the load change between 100 and 60 per cent. The gas turbine runs faster in order to compensate for the delay of the steam cycle system, thereby indicating the basic dynamic characteristics of the combined-cycle power generation system.

3.5.3 Controller test facility

A real-time plant simulator for controller testing (called “Virtual Design SystemTM”) has been developed. It provides a demonstrator and test bed for the controllers and the operator interface and facilitates the development of a prototype controller without the risk of causing damage to the actual plant. Figure 3.10 shows the schematic diagram of the controller test facility. Further, Fig. 3.11 shows the photograph of the controller test facility. A plant simulator (SIM) in which the dynamic simulation tool is installed, an operator station (OPS), an integrated engineering station (IES), and a controller are linked to each other through the Ethernet network.

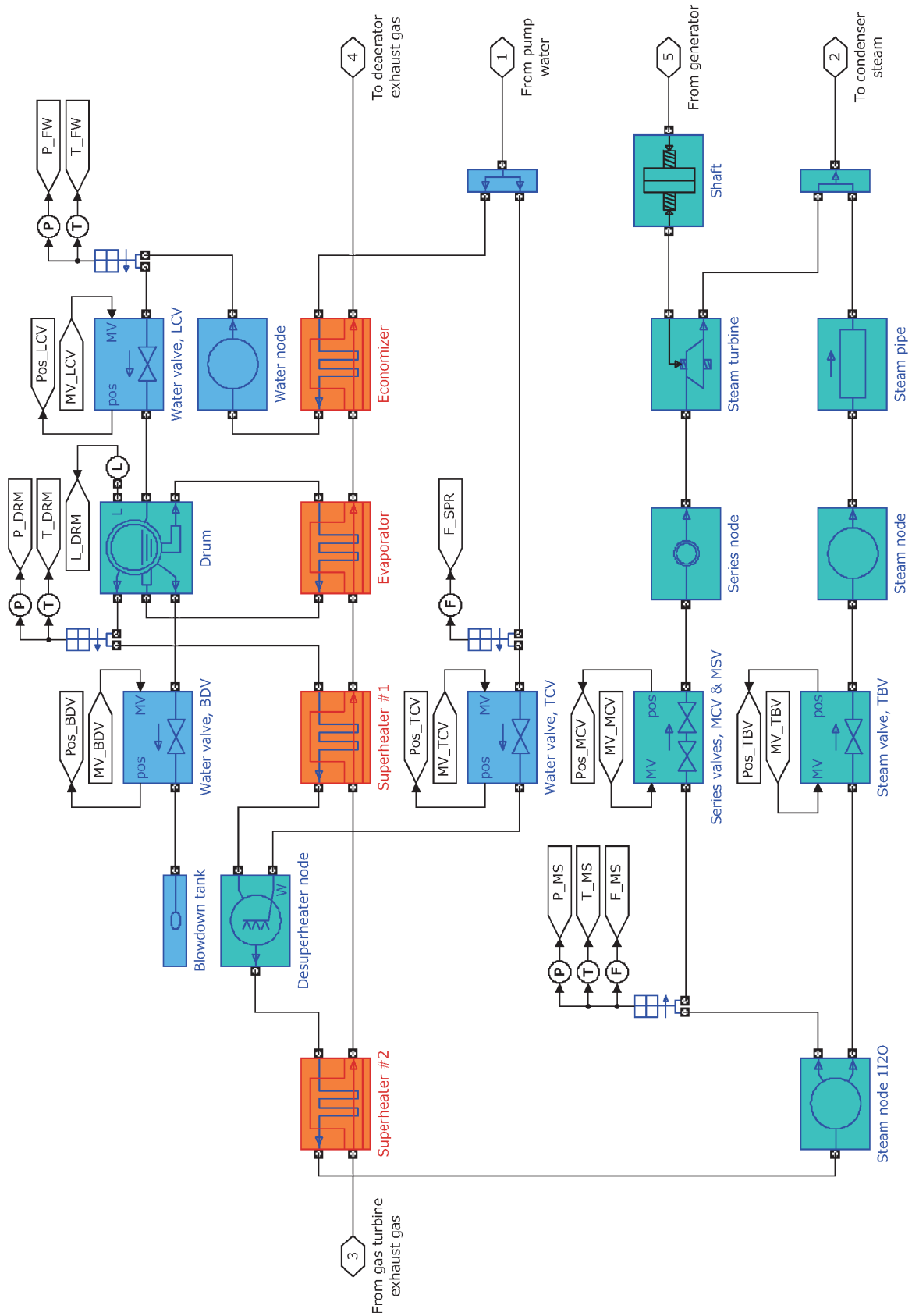


Fig. 3.8 Part of the screen display of the whole plant model.

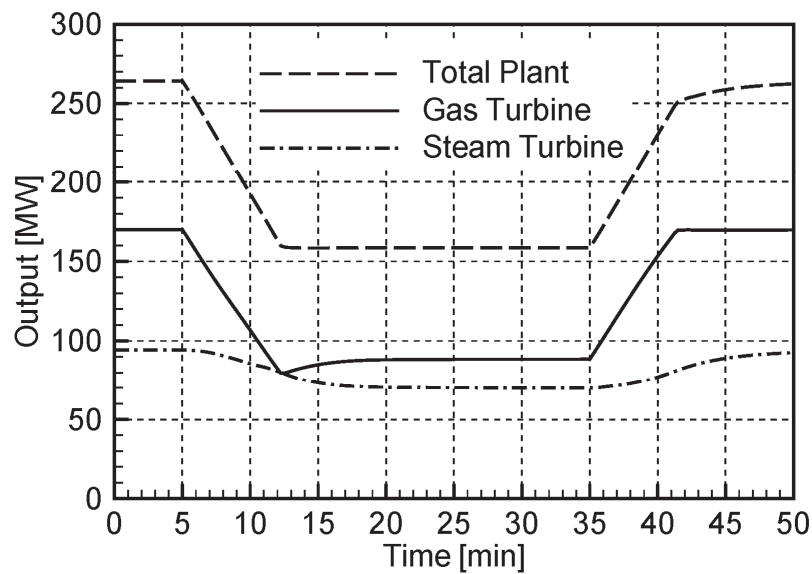


Fig. 3.9 Example of the simulation result. This figure shows the generator output response for the load change between 100 and 60 per cent.

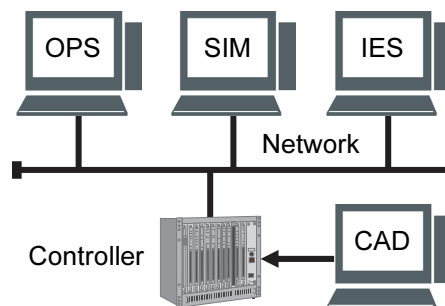


Fig. 3.10 Schematic diagram of the controller test facility. OPS denotes the operator station; SIM, the plant simulator; IES, the integrated engineering station; and CAD, the computer-aided design system for drawing the control logic diagram.

The control logic diagram is drawn by design engineers using a CAD tool. The software for plant control is automatically converted and directly installed into the controller from the CAD tool. IES is an engineering tool that adjusts and maintains the control parameters. The role of the controller is to control component equipment of the plant and generate the control signals for moving actuators such as valves and vanes.

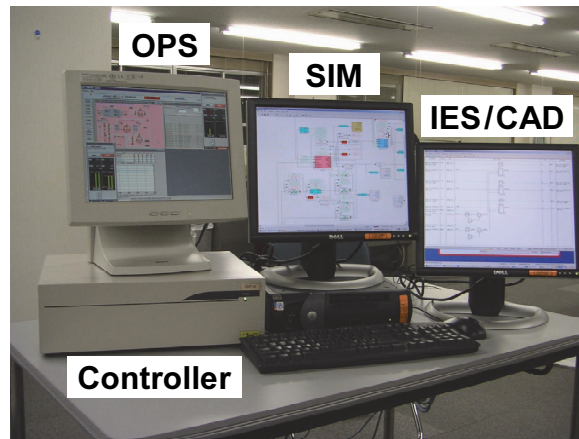


Fig. 3.11 Photograph of the controller test facility. The design engineers can carry out plant and control system design while evaluating the dynamic response of the plant in a seamless system.

SIM simulates all the physical dynamic responses such as pressures, temperatures, and flow rates on the basis of the control signals obtained through the network and sends them back to the controller in a similar manner. A real-time simulation program has been developed. This program synchronizes a PC's system clock with SIM. Further, an interface program between the SIM and the controller has been developed. This program communicates the control signals through a PC's shared memory. OPSs are human-machine communication terminals such as those for control instruction input and trend graph display. SIM, OPS, IES, and CAD tools work on a PC, and the controller comprises a CPU (central processing unit) board and an I/O (input/output) board. This system is compact in size and can therefore be set up on an office desk. The design engineers can carry out plant and control system design while evaluating the dynamic response of the plant in a seamless system.

3.6 Virtual design environment with the hardware-less simulation system

Enhancement of the operational efficiency of thermal power plants has become increasingly important since the 2011 Great East Japan Earthquake. Consequently, it has become necessary to reduce the production lead times and site installation periods at the time of controller renewal, such as plant supervisory monitoring and control system replacement.

Recently, we extended the controller test facility explained in Section 3.5.3, and developed an innovative production environment that encompasses each of the development processes, from design and manufacturing to the testing of distributed control systems (DCSs) for thermal power plants, including plant supervisory monitoring and control systems, by applying virtualization technologies. This production environment consists of both a virtual design environment that facilitates the streamlining of design processes and a simulation system that realizes hardware-less tests. We applied this environment to a plant supervisory monitoring and control system, and it was confirmed that it not only reduces the production lead times and site installation periods, but it also improves the plant operational efficiency.

The hardware-less simulation system employs virtualization technology that uses many operating systems, as a virtual machine on one server. This system installs real control software in the virtual OPS and the virtual emulated control device (emulator), which performs the exactly same operation as a real DCS. This system is also used to verify the validity of a control system and software without using the hardware of the DCS. Moreover, to verify the controllability of a plant, it cooperates with the plant simulator based on this dynamic simulation tool. Figure 3.12 shows the diagram of the

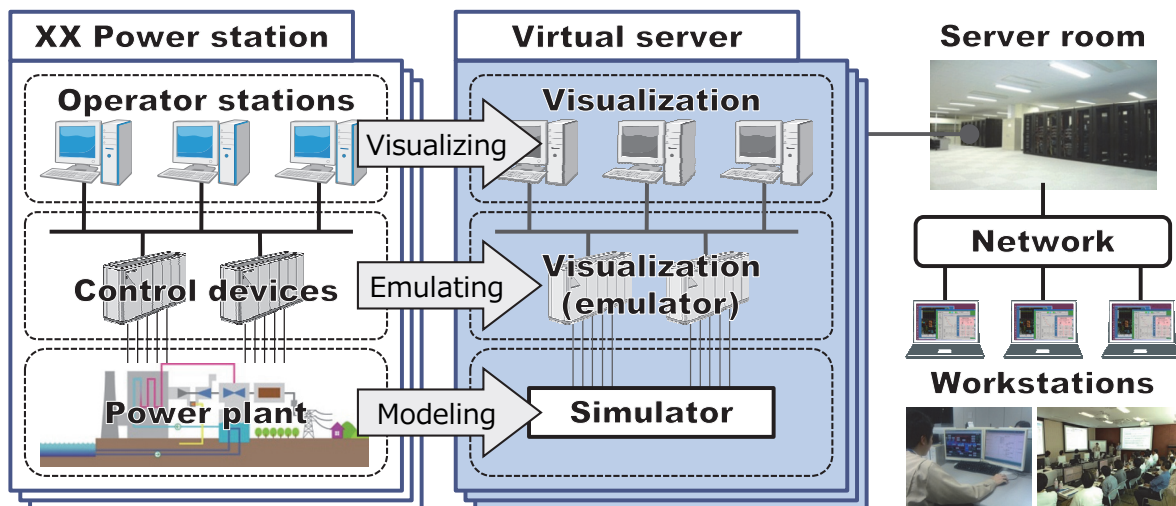


Fig. 3.12 Overview of the virtual design environment with the hardware-less simulation system for thermal power plants applying virtualization technologies. The desk verification test of plant operation and control is possible using a plant simulator.

hardware-less simulation system. For example, a large-scale DCS for thermal power plants consists of 12 OPS terminals, 15 control devices, and seven IES and maintenance tool terminals. By simulating the control action of a plant, we can test and verify the control software, including the dynamic characteristics (plant start-up and shutdown, emergency operation, boiler–turbine coordination control, etc.), and adjust a control parameter; this can improve the quality of the control software and the human–machine interface. Moreover, because the desk test of the control software can be performed without a real DCS, testing is simplified and testing automation is possible. Furthermore, the control software used in this system for verification and adjustment is installed in the real DCS. Thereby, software and hardware can be tested separately, and it becomes possible to ship real hardware at an early stage.

3.7 Conclusions

The author has developed a dynamic simulation tool that can be easily used by design engineers as a design tool for a thermal power plant. The results of this study are summarized as follows:

1. This tool enables a flexible plant configuration by modeling the respective component equipment on the basis of object orientation. The plant model can be easily constructed by connecting different equipment models by using a graphical user interface, and this is similar to creating a plant system diagram.
2. Each equipment model can be easily changed and added individually, managed in an integrated fashion, and recycled by the model library, thus providing excellent versatility, function extensibility, and maintainability.
3. This tool was verified by using actual plant operation data, and excellent agreements in the results were obtained. This indicates that this tool can be used for plant and control system design.
4. A virtual design environment was developed with the hardware-less simulation system to which virtualization technology is applied to improve the operational efficiency of thermal power plants.

At present, Toshiba Energy Systems & Solutions Corporation has used this tool as a standard analysis code in which the actual plant and control system design is evaluated, and the use of this tool has accelerated and enhanced its product design and testing processes.

Appendices

A1. Notation

A	heat transfer area [m^2]
C	specific heat [$\text{kJ}/(\text{kg} \cdot ^\circ\text{C})$]
G	mass flow rate [kg/s]
h	specific enthalpy [kJ/kg]
k	heat transfer coefficient [$\text{kW}/(\text{m}^2 \cdot ^\circ\text{C})$]
M	mass [kg]
n	total number of partitions
P	pressure [kPa]
Q	heat flow [kW]
S	section area [m^2]
t	time [s]
T	temperature [$^\circ\text{C}$]
v	specific volume [m^3/kg]
X	experimental experience value (e.g., $X = 0.69$)
Y	experimental experience value (e.g., $Y = 0.8$)
α	heat transfer parameter = $k \times A$ [$\text{kW}/^\circ\text{C}$]
ζ	flow coefficient [$-$]
$\Delta\theta$	logarithmic mean temperature difference [$^\circ\text{C}$]

Subscripts

g	gas (fluid outside the tube)
i	number of partitions (steam side)
in	inlet
j	number of partitions (gas side), $j = n - i + 1$
m	heat exchanger tube metal
out	outlet
stm	steam (fluid inside the tube)

A2. Glossary of object-oriented terms

Class	an abstraction of an object that specifies its static and behavioral characteristics, including its public and private
-------	------------------------------------------------------------------------------------------------------------------------

Constructor	characteristics. A class is an abstract data type with a constructor template from which object instances are created an operation performed by a class member function that initializes a newly created class instance
Encapsulation	a modeling and implementation technique (information hiding) that separates the external aspects of an object from the internal ones, namely, the implementation details of the object
Inheritance	the relationship between classes whereby one class inherits a part of or the complete public description of another base class and instances inherit all the properties and methods of the classes they contain
Instance	an individual example of a class invoked via a class constructor
Message	a request to an object to carry out one of its operations
Method	a class member function encapsulated as a class member
Object	a concept or item with definite boundaries and meanings for the problem at hand; a class instance
Object orientation	a software development strategy that organizes software as a collection of objects that contain both structural and behavioral data
Polymorphism	the ability of a function/operator with a single name to refer to arguments or return the types of different classes at run time
Subclass	a class that inherits from one or more classes, which are referred to as its immediate super-classes

References

- Ahluwalia, K. S. and Domenichini, R., Dynamic modeling of a combined-cycle plant, *Journal of Engineering for Gas Turbines and Power*, Transactions of the ASME, Vol. 112, No. 2, (1990), pp. 164–167.
- Akiyama, T., Matsushima, T., Nagafuchi, N., Nakajima, A., Yoshizaki, H. and Matsumura, J., Dynamic simulation of an advanced combined cycle plant with three pressure and reheat cycle, *Proceedings of the JSME-ASME International*

- Conference on Power Engineering-93 (ICOPE-93), The Japan Society of Mechanical Engineers, Vol. 1, (1993), pp. 221–226, Tokyo, Japan.
- Akiyama, T., Matsumoto, H. and Asakura, K., Dynamic simulation and its applications to optimum operation support for advanced combined cycle plants, *Energy Conversion and Management*, Vol. 38, Issues 15–17, (1997), pp. 1709–1723.
- Bartolini, A., Leva, A. and Maffezzoni, C., Power plant simulator embedded in a visual programming environment, *Proceedings of the IFAC Symposium on Control of Power Plants and Power Systems (SIPOWER'95)*, (1995), pp. 153–158, Cancún, México.
- Casella, F. and Leva, A., Modelling of thermo-hydraulic power generation processes using Modelica, *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 12, Issue 1, (2006), pp. 19–33.
- Elmegaard, B., Simulation of boiler dynamics—development, evaluation and application of a general energy system simulation tool, Ph.D. Thesis, ET-PhD 99-02, Department of Energy Engineering, Technical University of Denmark (DTU), (1999).
- Faille, D., Hiret, A. and Davelaar, F., Integrated interface for power plant modelling, *IFAC Proceedings Volumes, The 5th IFAC Symposium on Power Plants and Power Systems Control 2003*, Vol. 36, Issue 20, (2003), pp. 433–438, Seoul, the Republic of Korea.
- Hashemi, A., Lin, C. S., Munis, A. and Lee, S., Dynamic simulation and performance evaluation of fossil power plants, EPRI report CS-6047, (1988), The Electric Power Research Institute (EPRI), Palo Alto, CA, the United States.
- Leva, A., Bartolini, A. and Maffezzoni, C., A process simulation environment based on visual programming and dynamic decoupling, *Simulation*, Vol. 71, Issue 3, (1998), pp. 183–193.
- Leva, A. and Maffezzoni, C., Modelling of power plants, In: *Thermal Power Plant Simulation and Control* (Flynn, D., ed.), (2003), pp. 17–60, The Institution of Engineering and Technology (IET), London, the United Kingdom.
- Lu, S., Swidenbank, E. and Hogg, B. W., An object-oriented power plant adaptive control system design tool, *IEEE Transactions on Energy Conversion*, Vol. 10, No. 3, (1995), pp. 600–605.
- Lu, S., Swidenbank, E. and Hogg, B. W., Integrated environment for power plant performance analysis and control design, *Proceedings of the IFAC Symposium on*

- Control of Power Plants and Power Systems (SIPOWER'95), (1995), pp. 31–36, Cancún, México.
- Lu, S. and Hogg, B. W., “Power plant analyser”—a computer code for power plant operation studies, *IEEE Transactions on Energy Conversion*, Vol. 11, No. 4, (1996), pp. 742–747.
- Lu, S., Dynamic modelling and simulation of power plant systems, *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, Vol. 213, No. 1, (1999), pp. 7–19.
- Maffezzoni, C., Issues in modelling and simulation of power plants, *Proceedings of the IFAC Symposium on Control of Power Plants and Power Systems*, (1992), pp. 15–23, Munich, Germany.
- Mansour, F. M., Abdul Aziz, A. M., Abdel-Ghany, S. M. and El-shaer, H. M., Combined cycle dynamics, *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, Vol. 217, No. 3, (2003), pp. 247–258.
- Ordys, A., Pike, A. W., Johnson, M. A., Katebi, R. M. and Grimble, M. J., *Modelling and Simulation of Power Generation Plants*, (1994), Springer-Verlag London, Ltd., London, the United Kingdom.
- Shirakawa, M., Nakai, A., Hosaka, S. and Tani, A., Development of a new dynamic simulation tool for thermal power plants, *Gas Turbine Society of Japan*, Vol. 33, No. 4, (2005), pp. 333–342 (in Japanese).
- Shor, S. W. W., *Modular modeling system (MMS)*, Volumes 1–5, EPRI report CS/NP-3016, (1987), The Electric Power Research Institute (EPRI), Palo Alto, CA, the United States.
- Taylor, D. A., *Object-Oriented Technology: A Manager's Guide*, (1991), Addison-Wesley Publishing Co., Boston, MA, the United States.

Part II

Multi-objective optimization system for plant operation design

Chapter 4

Intelligent start-up schedule optimization system for a thermal power plant

Abstract

This study proposes an intelligent start-up schedule optimization system for a thermal power plant. This system consists of a dynamic simulation, an artificial neural network, and an interactive multi-objective programming technique. In this study, a novel intelligent optimization method using the artificial neural network and a genetic algorithm to realize satisficing decision-making, which is an interactive multi-objective programming technique, has been developed and introduced into the system. The features of this system are as follows. (1) The start-up schedule can be optimized based on multi-objective evaluation and (2) an optimal and flexible start-up schedule can be determined with a reasonable computing time and calculation accuracy through human–computer interactions. The system is applied to a dynamic simulation for a combined cycle power plant, and to optimize from among multiple objectives, based on varying daily requirements. The application results show that optimal and flexible start-up schedules can be obtained within a reasonable computing time and with acceptable calculation accuracy.

Keywords

Artificial neural network, Dynamic simulation, Genetic algorithm, Intelligent system, Interactive optimization, Multi-objective optimization, Optimal schedule, Power plant, Satisficing trade-off method, Steam turbine

4.1 Introduction

Typically, the start-up scheduling problem for a thermal power plant has several conflicting objective functions such as those for the start-up time, fuel consumption rate, lifetime consumption rate of the machine, and pollutant emission rate. These

parameters are affected by the varying market price of electricity as well as the fuel, maintenance, and environmental costs. Therefore, in the overall plant management strategy, it is important to achieve a flexible start-up schedule based on a multiple-criteria decision-making process.

The start-up characteristics are evaluated by using a dynamic simulation (Ordys et al., 1994); however, determining the optimal start-up schedule is complicated because it is necessary to iterate the dynamic simulation on the basis of a trial-and-error heuristic approach, using the engineer's intuition and experience. Several methods for optimizing the start-up schedule have been proposed. For instance, a fuzzy expert system that uses an expert's knowledge in fuzzy reasoning has been proposed (Matsumoto et al., 1996, 1997). A genetic algorithm with enforcement operation to search along the edges of the operational constraints has been proposed (Kamiya et al., 1999). Furthermore, a dynamic optimization technique using a simultaneous nonlinear programming formulation has been applied (Bausa and Tsatsaronis, 2001a, 2001b). However, all these studies have aimed to optimize a single-objective function (e.g., only the start-up time is minimized under the operational constraints). In cases with multi-objective functions, it is very difficult to adjust the weights of each objective function. Moreover, the computing time increases drastically, making applications to practical problems impossible.

This study proposes an intelligent start-up schedule optimization system for a thermal power plant. This system consists of a dynamic simulation, an artificial neural network, and an interactive multi-objective programming technique. This system can determine an optimal and flexible start-up schedule from among multiple objectives, with a reasonable computing time and calculation accuracy through human-computer interactions.

4.2 Combined cycle power plant

4.2.1 Plant configuration

A combined cycle power plant is a high-efficiency thermal power plant. This study considers a multi-shaft type combined cycle power plant, as shown in Fig. 4.1. It consists of three gas turbine units, three heat recovery steam generator (HRSG) units, and one steam turbine unit. The gas turbines and the steam turbine drive the generators. In addition, the HRSGs generate steam for the steam turbine using waste heat from the

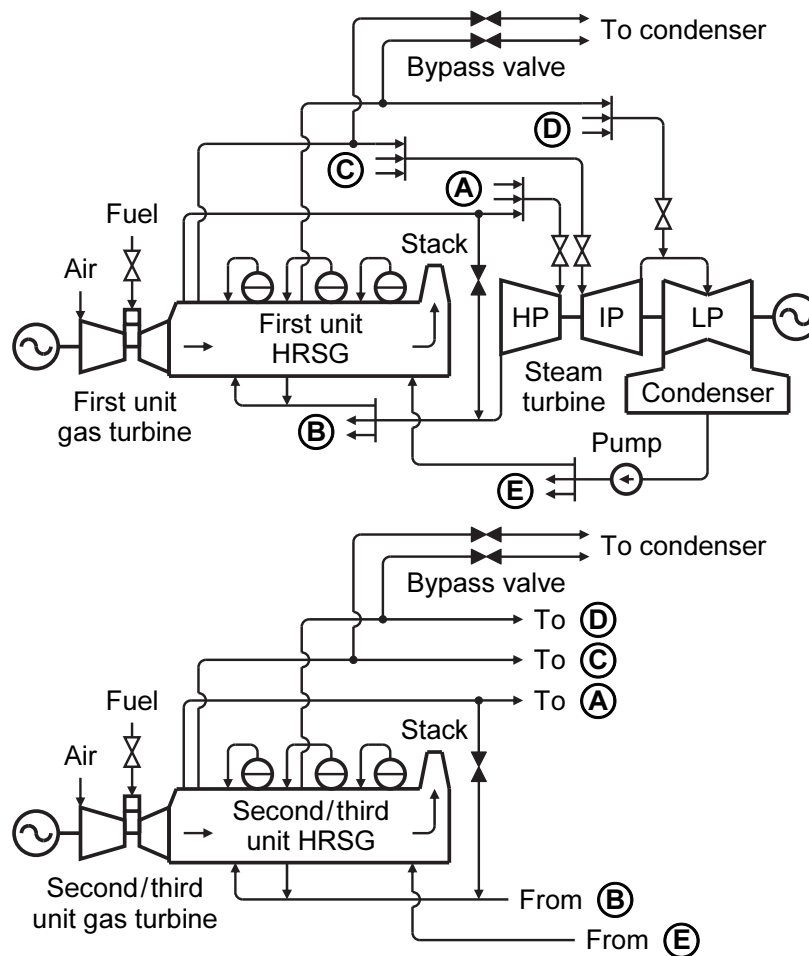


Fig. 4.1 Schematic diagram of combined cycle power plant (multi-shaft type). This plant consists of three gas turbine units, three HRSG units, and one steam turbine unit.

gas turbines. This plant generates a total output of 670 MW. The gas turbines are of the 1300°C class. The HRSGs are of the triple pressure and reheat type. The steam turbine has a high-pressure (HP) turbine, an intermediate-pressure (IP) turbine, and a low-pressure (LP) turbine.

4.2.2 Start-up scheduling method

The start-up process of this plant is shown in Fig. 4.2. Initially, the first gas turbine starts, purges, fires, and increases its speed toward the rated speed. After synchronizing, the first gas turbine goes up to a minimum load of 12.5%. The first HRSG warms at the minimum gas turbine load, and the bypass valves of the first unit control the steam

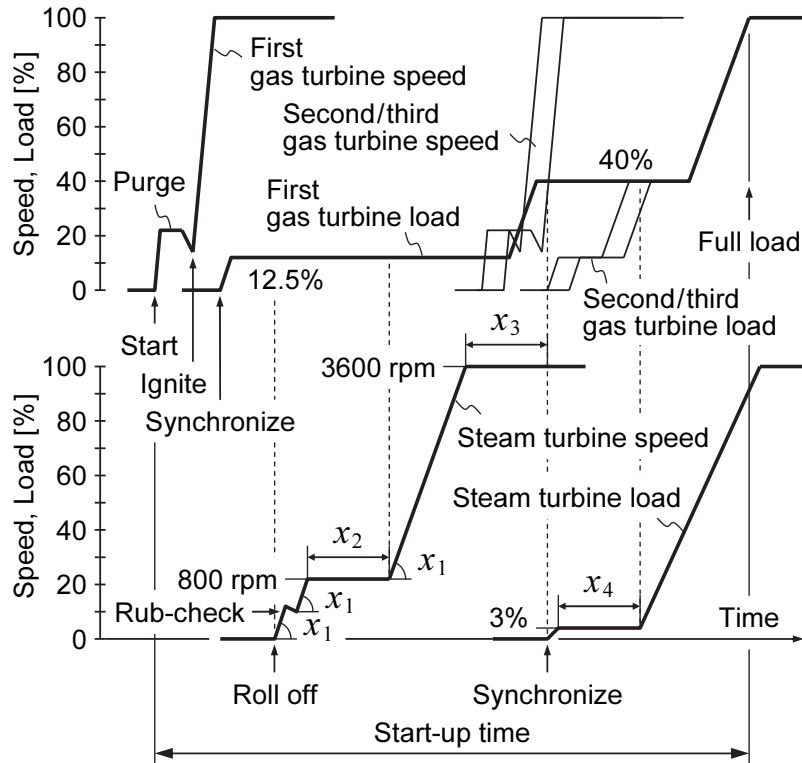


Fig. 4.2 Start-up curve for a multi-shaft type combined cycle power plant. Start-up curve sets a detailed schedule of the rate for the gas and steam turbines to reach nominal speed and full load.

pressures. When the steam properties have reached the required conditions, the steam turbine starts, and the steam turbine is put into the speed and load control mode. After conducting a rub-check, the steam turbine increases its speed toward a predefined speed of 800 rpm at an acceleration rate of x_1 , and maintains this speed during the low-speed heat soak time, x_2 . The steam turbine increases its speed again toward the rated speed (3600 rpm), and maintains this speed during the high-speed heat soak time, x_3 . After synchronizing, the steam turbine goes to an initial load of 3%, and maintains this load during the initial-load heat soak time, x_4 . Before completing the high-speed heat soak operation, the first gas turbine increases its load to a predefined load of 40% so that the steam flow rate is secured to drive the steam turbine. After completing the initial-load heat soak operation, the bypass valves of the first unit are closed, and the steam turbine is then put into the inlet pressure control mode. The second gas turbine and HRSG are started in the same manner as the first unit. The second gas turbine reaches a merged load of 40% and maintains this load. The bypass valves of the second unit are then

closed to merge the second HRSG steam into the steam turbine inlet steam. Afterwards, the third gas turbine and HRSG are operated in the same manner to merge the third HRSG steam into the steam turbine inlet steam. Lastly, all the gas turbines increase their load again toward the rated load. After the gas turbines have reached the rated load, the steam turbine reaches its rated load with a time delay due to the delay in the generation of steam in the HRSGs.

4.3 Start-up scheduling problem

In this study, the following schedule variables, objective functions, and operational constraints are considered to optimize the start-up schedule.

4.3.1 Schedule variables

The steam turbine start-up schedule has a significant effect on the start-up characteristics of the entire plant. This is due to the thermal stress that develops in the steam turbine rotors, which is a factor particularly decisive in reducing the start-up time and fuel consumption rate. Therefore, four schedule variables are selected, i.e., the steam turbine acceleration rate, x_1 , low-speed heat soak time, x_2 , high-speed heat soak time, x_3 , and initial-load heat soak time, x_4 ; these are shown in Fig. 4.2. These schedule variables restrain the developed thermal stress by gently warming the steam turbine rotors.

4.3.2 Objective functions

The objective functions are to minimize the start-up time, fuel consumption rate, and thermal stress of the steam turbine rotors. A smaller thermal stress has the effect of further extending the service lifetime of the steam turbine. Expectedly, the start-up time and fuel consumption rate reductions improve the load following capability and fuel economy. However, these objective functions have a trade-off depending on the above-mentioned schedule variables (x_1 , x_2 , x_3 , and x_4). As acceleration rate, x_1 , increases and heat soak times, x_2 , x_3 , and x_4 , decrease, both the start-up time and fuel consumption rate decrease; however, the thermal stress of the steam turbine rotors increases. This trend of the fuel consumption rate becomes more significant as the initial-load heat soak time, x_4 , is varied, because the gas turbine load is higher with x_4 than with the other schedule variables (x_1 , x_2 , and x_3).

4.3.3 Operational constraints

There exist many operational constraints in this plant. However, most constraints (e.g., the drum water level and the steam temperature) are safety-controlled to within given limits at any time. As a result, the operational constraints include only the thermal stress of the steam turbine rotors and the NO_x emission rate from the plant. The thermal stress of the steam turbine rotors in the above-mentioned objective functions has an upper limit to prevent metal creep and fatigue. The NO_x emission rate from the plant increases significantly with a rapid start-up; however, it has an upper limit in accordance with the environmental regulations.

4.4 Concept of the system

A high-accuracy plant simulator has been developed by the author, as described in Chapters 2–3. The dynamic simulation models are derived from first principles (thermo-hydraulic conservation equations), and are implemented in the software package MATLAB/Simulink—a widely used package software created by The MathWorks, Inc. When the values of the schedule variables are provided, the values of the evaluation functions (the evaluation functions represent both the objective functions and operational constraints) can be obtained by using the dynamic simulation. However, because detailed, large-scale, and nonlinear models are used, the high-accuracy plant simulator is extremely time-consuming. Moreover, there are many optimal solutions that are referred to as “Pareto-optimal solutions” in the multi-objective optimization problem; hence, a considerable amount of labor is required to find the final solution. Therefore, it is well known that multi-objective optimization methods using dynamic simulation require an unrealistic amount of time to obtain the optimal start-up schedule because they require a large number of dynamic simulation calls. On the other hand, the user (i.e., the plant operator, manager, or design engineer) ought to make a decision regarding the start-up schedule within a short time, according to the varying daily requirements, i.e., the market price of electricity as well as the fuel, maintenance, and environmental costs.

In order to obtain quick solutions, the dynamic simulation, an artificial neural network, and an interactive multi-objective programming technique are integrated within a cooperative human–machine system. Figure 4.3 shows the functional structure of the intelligent start-up schedule optimization system proposed in this study. The

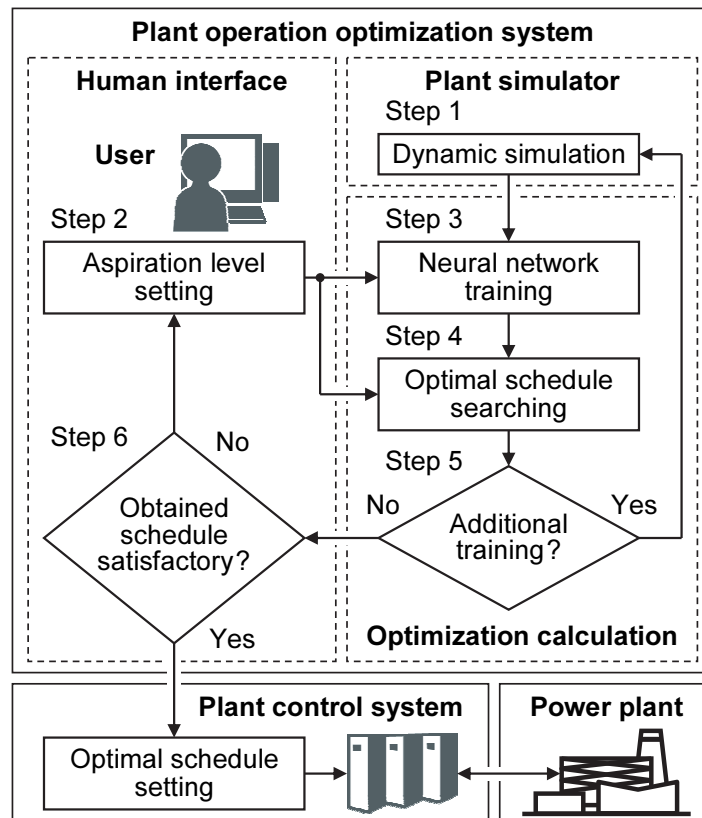


Fig. 4.3 Plant operation optimization system. A novel intelligent optimization method has been developed using an interactive multi-objective programming technique.

major part of this system consists of a plant simulator, a human interface, and an optimization calculation.

4.4.1 Plant simulator part

First of all, to maintain the calculation accuracy, the plant simulator is adjusted as the simulation results are in good correspondence with the actual plant data. After that, the dynamic simulation of a variety of start-up schedules is executed beforehand. Further, the simulation results are saved as training data for the neural network. These training data contain datasets of the schedule variables and evaluation functions.

4.4.2 Human interface part

The user must find a final solution from among the Pareto-optimal solutions, taking into account the trade-off relationships between the objective functions. The proposed

system can easily and interactively determine the final solution through human–computer interactions. The user performs only two functions in the human interface section. The first is to set the desired value of the objective functions, i.e., the “aspiration level.” This is sent to the optimization calculation part, which determines the start-up schedule (one of the Pareto-optimal solutions) that is closest to the aspiration level. The second is to judge whether or not the obtained start-up schedule is satisfactory. If the user is not satisfied, the user modifies the aspiration level, which is resent to the optimization calculation part. If the user is satisfied, the optimal start-up schedule is set in the plant control system. The plant control system subsequently operates the power plant according to the optimal start-up schedule.

4.4.3 Optimization calculation part

A novel intelligent optimization method has been developed using an artificial neural network and a genetic algorithm to realize satisficing decision-making, which is an interactive multi-objective programming technique. This method is explained in detail in Section 4.5. Since the dynamic simulation is computationally expensive, approximate functions using the neural network are introduced to create inexpensive surrogate models. Further, the start-up schedule closest to the aspiration level is searched with the approximate functions by using the genetic algorithm. The judgment of whether additional training for the neural network is required is used to improve calculation accuracy.

4.5 Intelligent optimization method

4.5.1 Satisficing trade-off method

Multi-objective programming problems are typically expressed as follows:

$$\begin{cases} \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_r(\mathbf{x})]^T, \\ \text{subject to } g_\ell(\mathbf{x}) \leq 0, \quad \ell = 1, 2, \dots, s, \end{cases} \quad (4.1)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ represents the vector of the schedule variables (design variables); f , the objective functions; and g , the operational constraints (constraining conditions) used in this study.

The satisficing trade-off method is an interactive multi-objective programming

technique based on aspiration levels (Nakayama, 1995). This method can solve Eq. (4.1) via human–computer interactions while eliciting information from the user. The vector of the aspiration level at the k -th iteration, $\overline{\mathbf{f}}_k$, is modified as follows:

$$\overline{\mathbf{f}}_{k+1} = T \circ P(\overline{\mathbf{f}}_k). \quad (4.2)$$

Here, operator P selects the Pareto-optimal solution nearest to $\overline{\mathbf{f}}_k$. Operator T is the trade-off operator that changes $\overline{\mathbf{f}}_k$ if the user does not agree with displayed solution, $P(\overline{\mathbf{f}}_k)$. Indeed, since $P(\overline{\mathbf{f}}_k)$ is one of the Pareto-optimal solutions, there exists no feasible solution whose criteria are all better than the criteria of $P(\overline{\mathbf{f}}_k)$; thus, the user will have to trade off some criteria to improve others. Given this trade-off, a new aspiration level, $T \circ P(\overline{\mathbf{f}}_k)$, is identified. The process continues in a similar fashion until the user obtains an agreeable solution.

4.5.1.1 On operation P

The operation that returns the Pareto-optimal solution $P(\overline{\mathbf{f}}_k)$ nearest to $\overline{\mathbf{f}}_k$ is performed with the following auxiliary scalar optimization of Eq. (4.4), using the Chebyshev scalarization function, F , of Eq. (4.3). Here, the operational constraints can be included in Eq. (4.4) as the penalty function.

$$F(\mathbf{x}) = \max_{1 \leq i \leq r} w_{i,k} [f_i(\mathbf{x}) - \overline{f}_{i,k}] + \alpha \sum_{i=1}^r w_{i,k} f_i(\mathbf{x}), \quad (4.3)$$

$$\left\{ \begin{array}{l} \min_{\mathbf{x}} F(\mathbf{x}) + \sum_{\ell=1}^s \beta_{\ell} K_{\ell} g_{\ell}(\mathbf{x}), \\ K_{\ell} = \begin{cases} 0, & \text{if } g_{\ell}(\mathbf{x}) \leq 0, \\ 1, & \text{otherwise,} \end{cases} \end{array} \right. \quad \ell = 1, 2, \dots, s, \quad (4.4)$$

where α is usually set to a sufficiently small positive number (such as 10^{-6}), and penalty parameter β_{ℓ} is a sufficiently large positive number for each constraining condition g_{ℓ} . Weight $w_{i,k}$ can be expressed as follows:

$$w_{i,k} = \frac{1}{f_{i,k} - f_i^*}, \quad (4.5)$$

where f_i^* is an ideal point, usually given such that $f_i^* \leq \min [f_i(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}]$, where \mathbf{X} is the set of all feasible solutions.

4.5.1.2 On operation T

If the user is dissatisfied with solution $P(\overline{\mathbf{f}}_k)$, the user is asked to provide a new aspiration level, $\overline{\mathbf{f}}_{k+1}$, and classify the objective functions into three groups:

1. the class of criteria that are to be improved further,
2. the class of criteria that can be relaxed, and
3. the class of criteria that are acceptable as they are.

Let the index sets for Classes 1, 2, and 3 be denoted as I_I^k , I_R^k , and I_A^k , respectively. Clearly, $\overline{f_{i,k+1}} \leq f_i(\mathbf{x}_k)$ for all $i \in I_I^k$. Usually, for $i \in I_A^k$, we set $\overline{f_{i,k+1}} = f_i(\mathbf{x}_k)$. For $i \in I_R^k$, the user must agree to increase the value of $\overline{f_{i,k+1}}$. Note that an appropriate sacrifice of f_j for $j \in I_R^k$ is required to improve f_i for $i \in I_I^k$.

For example, we consider a minimization problem with two objective functions, as shown in Fig. 4.4. The user sets the ideal point (the symbol * in Fig. 4.4) and the first aspiration level (the symbol \square in Fig. 4.4). In Fig. 4.4, line IA is the segment between the ideal point and the first aspiration level. The dashed curve is the Pareto frontier, which is usually an unknown function. The first optimal solution (the symbol \blacksquare in Fig. 4.4) is the intersection of line IA and the Pareto frontier. If the user modifies the

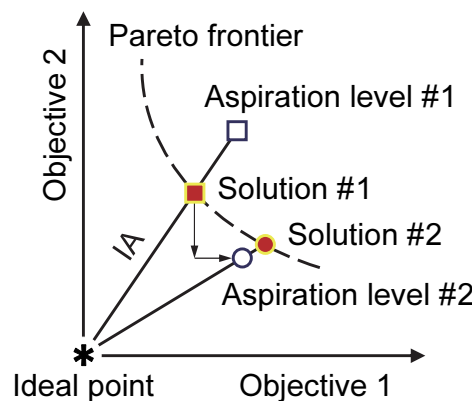


Fig. 4.4 Concept of the satisficing trade-off method. Here, the line IA is the segment between the ideal point and the aspiration level. The dashed curve is the Pareto frontier. The obtained Pareto-optimal solution corresponds to the intersection of the line IA and the Pareto frontier.

aspiration level to the second aspiration level (the symbol \bigcirc in Fig. 4.4), the solution moves to the second optimal solution (the symbol \bullet in Fig. 4.4). The user continues this process until a desirable solution is obtained.

4.5.2 Radial basis function network

A radial basis function network (RBFN) is a type of artificial neural network that consists of a number of radial basis functions (Nakayama et al., 2002). An RBFN consists of three layers: the input layer, hidden layer, and output layer. The input values are each assigned to a neuron in the input layer and passed directly to the hidden layer without weights. A Gaussian function is used as an activation function for the hidden neurons. Subsequently, linear output weights are added between the hidden and output layers. The overall input–output mapping equation of an RBFN is presented as follows:

$$O'(\mathbf{x}) = \sum_{j=1}^q w_j h_j(\mathbf{x}), \quad (4.6)$$

$$\text{where } h_j(\mathbf{x}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{c}_j\|^2}{R_j^2}\right) \quad (4.7)$$

and \mathbf{c}_j represents the center of the hidden neurons; h_j , the Gaussian function; O' , the output values; q , the number of hidden neurons; R_j , the radial parameter; w_j , the weights; $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, the vector of the input values; and $\|\cdot\|$, the Euclidean norm.

The training equation of an RBFN is defined as follows:

$$\min_{\mathbf{w}} E(\mathbf{w}) = \sum_{i=1}^p [y_i - O'(\mathbf{x}_i)]^2 + \sum_{j=1}^q \lambda_j w_j^2, \quad (4.8)$$

where p represents the number of training values; $\mathbf{w} = [w_1, w_2, \dots, w_q]^T$, the weight vector; and y_i , the training values. λ_j is introduced for the purpose of regularization. Equation (4.8) can be reduced to the normal equation as follows:

$$\mathbf{H}^T \mathbf{y} = \mathbf{H}^T \mathbf{O}' + \lambda \mathbf{I} \mathbf{w} = \mathbf{H}^T \mathbf{H} \mathbf{w} + \lambda \mathbf{I} \mathbf{w} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}) \mathbf{w}, \quad (4.9)$$

$$\text{where } \mathbf{H} = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \cdots & h_q(x_1) \\ h_1(x_2) & h_2(x_2) & \cdots & h_q(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ h_1(x_p) & h_2(x_p) & \cdots & h_q(x_p) \end{bmatrix}, \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}, \quad (4.10)$$

$$\mathbf{O}' = \begin{bmatrix} O'(x_1) \\ O'(x_2) \\ \vdots \\ O'(x_p) \end{bmatrix}, \lambda \mathbf{I} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_q \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_q \end{bmatrix},$$

$$\therefore \mathbf{w} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \mathbf{H}^T \mathbf{y}. \quad (4.11)$$

Hence, RBFN training can be solved directly with the linear simultaneous equation in Eq. (4.11). As a result, the training speed improves significantly and local minima are avoided. Moreover, even if the RBFN is not relearned from the beginning by updating the inverse matrix $\mathbf{A}^{-1} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1}$, additional RBFN learning and forgetting can be performed easily.

Our proposed method generates approximate functions by combining the polynomial functions and RBFN as follows:

$$O(\mathbf{x}) = \zeta(\mathbf{x}) + O'(\mathbf{x}), \quad (4.12)$$

where O represents the approximate functions; O' , the RBFN output values from Eq. (4.6); and ζ , the polynomial functions. The polynomial functions are used as base functions, and the RBFN estimates errors from them. This method can improve the accuracy near the side constraints of the schedule variables (design variables), where the training values become sparse.

4.5.3 Genetic range genetic algorithm

The minimization of Eq. (4.4) is performed by using a genetic algorithm. A genetic range genetic algorithm (GRGA) (Arakawa and Ishikawa, 2002) is applied in this study. Since it automatically adopts a searching range according to the situation of the generation, a GRGA converges rapidly. The theory of GRGA is explained as follows.

4.5.3.1 Definition of searching range

In a GRGA, we define the searching range by using a Gaussian distribution, as shown in Eq. (4.13).

$$N(x_i) = \exp\left[-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right]. \quad (4.13)$$

With Eq. (4.13), the basic searching range is determined by the mean value, μ_i , and standard deviation, σ_i , of the surviving individuals for each schedule variable, x_i ($i = 1, 2, \dots, n$). In the case of continuous values, with this basic searching range, decoding is performed as shown in Eq. (4.14); this process is illustrated in Fig. 4.5.

$$R_c(p_i) = \begin{cases} \mu_i^k - \sigma_i^{L,k} \sqrt{-2 \ln \left[LB_i + \frac{\Delta B_i C(p_i)}{2^{m_i-1} - 1} \right]}, & \text{if } C(p_i) \leq 2^{m_i-1}, \\ \mu_i^k + \sigma_i^{R,k} \sqrt{-2 \ln \left\{ LB_i - \frac{\Delta B_i [C(p_i) - 2^{m_i-1}]}{2^{m_i-1} - 1} \right\}}, & \text{otherwise,} \end{cases}$$

$$\text{where } \Delta B_i = UB_i - LB_i \quad (4.14)$$

and m_i represents the number of bits; p_i , the chromosome; $R_c(p_i)$, the decoded

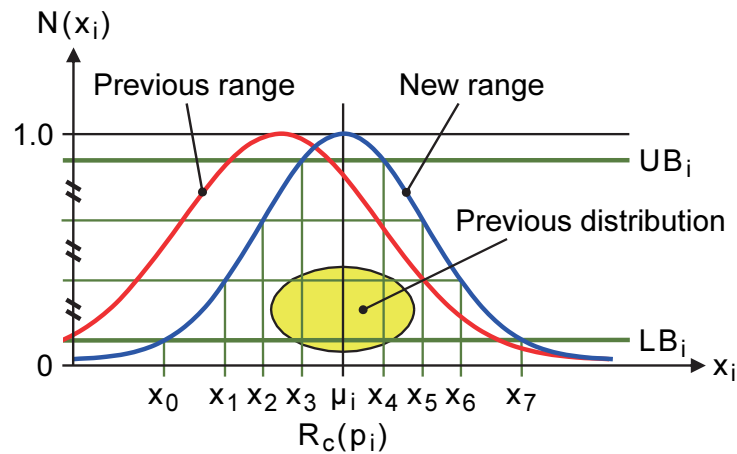


Fig. 4.5 Concept of the GRGA. In this method, a range of real numbers will move adaptively in each generation by using the mean value and the standard deviation of the previous generation.

continuous value; $C(p_i)$, the decoded integer value by using gray coding; σ_i^L and σ_i^R , the parameters to determine the range on left hand side and right hand side, respectively; UB_i and LB_i , the upper and lower bounds, respectively, for $N(x_i)$ in Fig. 4.5; and superscript k , the k -th searching range.

In case of discrete values, we can use the same expression for the mean value as the key for decoding.

$$b_i^k = \min \left\{ \max \left[0, \text{int} \left(\mu_i^k - 2^{m_i-1} + 0.5 \right) \right], DB_{U_i} - 2^{m_i} + 1 \right\}. \quad (4.15)$$

Based on Eq. (4.15), decoding can be performed based on the database number as follows:

$$R_d(p_i) = \text{database} \left[b_i^k + C(p_i) \right], \quad (4.16)$$

where $R_d(p_i)$ is the decoded discrete value, and DB_{U_i} is the database number, counting from 0. Thus, in a GRGA, we can treat both continuous and discrete values by using the same setting for the searching range.

4.5.3.2 Inheritance of searching range

In simple genetic algorithms, only the crossover of genotype is implemented. However, in a GRGA, the searching range must also be inherited. As shown in Fig. 4.6, in a GRGA, a parent string usually has four offspring, with two having the same searching range and the other two having the same gene. Moreover, an immigrant mutation happens only according to the mutation probability.

As for survival, we use the following roulette-wheel selection process. Before we start the selection, we normalize the fitness function value, F_j^* , as follows:

$$F_j^* = \max \left(5 \pm 5 \frac{F_j - F_{\text{mean, init}}}{F_{\text{min/max, init}} - F_{\text{mean, init}}}, F_{LB}^* \right), \quad (4.17)$$

where F_{LB}^* represents the lower bound for the normalized fitness function value; in the case of maximizing the fitness function, we use $\langle + \rangle$, and in the other case, we use $\langle - \rangle$. $F_{\text{mean, init}}$ represents the mean value of the fitness function for initial generation; $F_{\text{min/max, init}}$, the maximum value of the fitness function for initial generation in case of

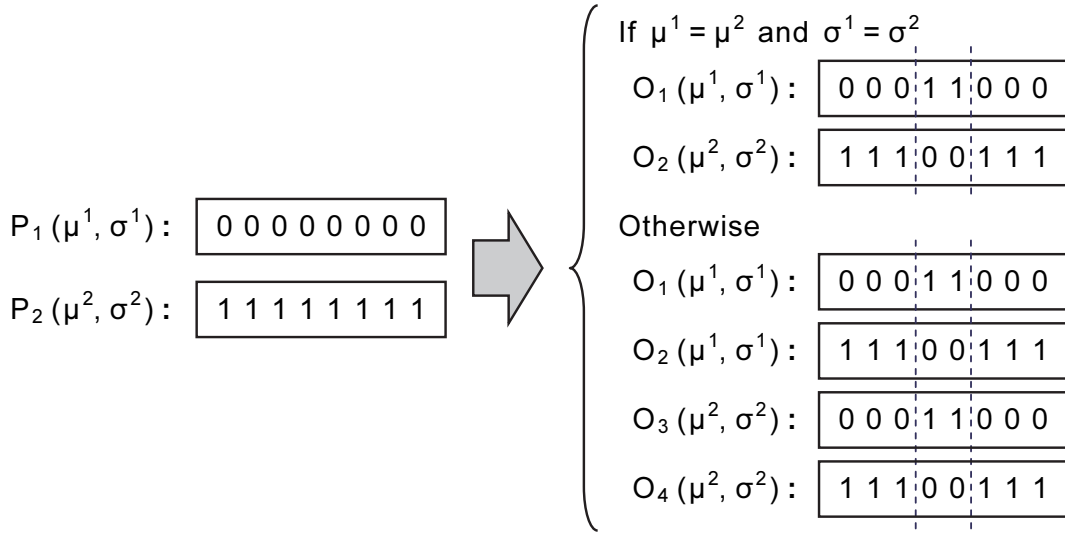


Fig. 4.6 Crossover in the GRGA. In this method, a parent string usually has four offspring, with two having the same searching range and the other two having the same gene.

maximization and minimum value in minimization; F_j , the fitness function value; and subscript j , the individual. The fitness function value, F_j , is a value in which the Chebyshev scalarization function, F , of Eq. (4.3) is regularized from 0 to 10.

4.5.3.3 Initial searching range

The key to obtaining the searching range is to provide the mean value as well as both the right and left hand side standard deviations.

$$\mu_i^k = x_{i, \min, \text{init}} + \left[\gamma + (1 - \gamma) \text{rand} \left(x_{i, \max, \text{init}} - x_{i, \min, \text{init}} \right) \right], \quad (4.18)$$

$$\sigma_i^{L, k} = \sqrt{-\frac{\left(x_{i, \min, \text{init}} - \mu_i^k \right)^2}{2 \ln(LB_i)}}, \quad (4.19)$$

$$\sigma_i^{R, k} = \sqrt{-\frac{\left(x_{i, \max, \text{init}} - \mu_i^k \right)^2}{2 \ln(LB_i)}}, \quad (4.20)$$

where $x_{i, \max, \text{init}}$ and $x_{i, \min, \text{init}}$ represent the maximum and minimum values of the

searching range for initial generation, respectively, and γ represents the parameter used to prevent the mean value from being extremely biased toward the edges.

4.5.3.4 New searching range

By using the genetic algorithms process, the number of individuals with the same searching range will increase or decrease. If the number of individuals becomes lower than a set minimum, the searching range is annihilated and all the individuals are erased. If there is any annihilation, a new searching range must be produced. There are three patterns for obtaining ranges:

$$\mu_i^k = \begin{cases} \mu_i, \\ x_{i, \text{best}}, \\ x_{i, \text{min}} + \text{rand}(x_{i, \text{max}} - x_{i, \text{min}}), \end{cases} \quad (4.21)$$

where μ_i represents the mean value of the surviving individuals; $x_{i, \text{best}}$, the best obtained solution; and $x_{i, \text{max}}$ and $x_{i, \text{min}}$, the maximum and minimum values of the searching range before its generation, respectively. The second pattern is used for minute local searching, and the third for global searching. We provide percentages for each setting and choose them randomly according to those settings. In order to ensure that the new range remains within the boundaries, we introduce the following functions:

$$\begin{cases} \mu_i^k = \min \left[\max \left(x_{i, \text{bound, min}} + s, \mu_i^k \right), x_{i, \text{bound, max}} - s \right], \\ s = \sqrt{-2\sigma_{i, \text{min}} \ln(LB_i)}, \end{cases} \quad (4.22)$$

$$\begin{cases} \sigma_i^{L, k} = \min \left\{ \max \left[\sigma_{i, \text{min}}, \min \left(\sigma_{i, \text{bound, min}}, \sigma_i \right) \right], \sigma_{i, \text{max}} \right\}, \\ \sigma_{i, \text{bound, min}} = \sqrt{-\frac{\left(x_{i, \text{bound, min}} - \mu_i^k \right)^2}{2 \ln(LB_i)}}, \end{cases} \quad (4.23)$$

$$\begin{cases} \sigma_i^{R, k} = \min \left\{ \max \left[\sigma_{i, \text{min}}, \min \left(\sigma_{i, \text{bound, max}}, \sigma_i \right) \right], \sigma_{i, \text{max}} \right\}, \\ \sigma_{i, \text{bound, max}} = \sqrt{-\frac{\left(x_{i, \text{bound, max}} - \mu_i^k \right)^2}{2 \ln(LB_i)}}, \end{cases} \quad (4.24)$$

where σ_i represents the standard deviation of the surviving individuals; $\sigma_{i, \max}$ and $\sigma_{i, \min}$, the maximum and minimum values of the standard deviation, respectively; and $x_{i, \text{bound}, \max}$ and $x_{i, \text{bound}, \min}$, the maximum and minimum values of the searching boundary, respectively. The maximum value of the standard deviation is set to control the spreading of the searching range.

4.5.4 Numerical algorithm

A calculation flowchart is shown in Fig. 4.3. The numerical algorithm has the following steps:

Step 1. Plant simulator part: The training data for the RBFN are prepared by dynamic simulation. These training data contain datasets of the schedule variables and evaluation functions.

Step 2. Human interface part: The user sets the aspiration level of the objective functions.

Step 3. Optimization calculation part: In order to create the approximate functions, the RBFN is trained using the values of the schedule variables as the input values, and the values of the evaluation functions, in terms of the schedule variables, as the teacher's values. The approximate function of the Chebyshev scalarization function, F , of Eq. (4.3) for the aspiration level is created by either of the following methods:

1. A method that directly acquires the form of F by similarly using the RBFN
2. A method that calculates the function using the created approximate evaluation functions

Step 4. Optimization calculation part: The start-up schedule closest to the aspiration level is searched to minimize the auxiliary scalar optimization of Eq. (4.4) on the approximate functions by using the GRGA; it is obtained as an approximate optimal solution.

Step 5. Optimization calculation part: In order to evaluate the approximation errors, the dynamic simulation is executed according to the obtained start-up schedule. If the approximation errors are large, some additional training points for the RBFN are provided in the neighborhood of the obtained start-up schedule. Further, Steps 1–5 are repeated until the approximation errors are reduced. If the approximation errors are small, the obtained start-up schedule is displayed to the user.

Step 6. Human interface part: The user judges whether or not the obtained start-up

schedule is satisfactory. If the user is not satisfied, the user modifies the aspiration level. Subsequently, Steps 2–6 are repeated until the user obtains an agreeable start-up schedule. If the user is satisfied, the optimal start-up schedule is determined.

4.6 Application results

4.6.1 Formulation

Simulation studies have been executed for a warm start-up condition, i.e., the initial temperature of the steam turbine rotors is 180°C to 290°C. The steam turbine acceleration rate, x_1 , low-speed heat soak time, x_2 , high-speed heat soak time, x_3 , and initial-load heat soak time, x_4 , are treated as the schedule variables (see Fig. 4.2). Here, x_1 is a discrete variable with three values, and x_2 , x_3 , and x_4 are continuous variables with upper and lower limits. The lower limit of x_2 is set to prevent metal brittle fracture of the steam turbine rotors. The remaining upper and lower limits of x_2 , x_3 , and x_4 are set to provide the operational range. The start-up time, f_1 , is from the start of the first gas turbine to the realization of the plant base-load operation. The fuel consumption rate, f_2 , is the gross weight utilized during start-up. The thermal stress of the steam turbine rotors, f_3 ($= g_{c1}$), is the maximum value during start-up. The NO_x emission rate from the plant, g_{c2} , is the maximum value of the moving average per hour during start-up. Further, g_{c1} and g_{c2} have safe upper limits g_{U1} and g_{U2} , respectively. The problem is to find the vector of the schedule variables (design variables), $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$, in which the objective functions f_1 , f_2 , and f_3 are minimized under the operational constraints (constraining conditions) $g_{c1} \leq g_{U1}$ and $g_{c2} \leq g_{U2}$. This problem can be summarized as follows:

$$\begin{cases} \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]^T, \\ \text{subject to } \begin{cases} g_1(\mathbf{x}) = g_{c1}(\mathbf{x}) - g_{U1} \leq 0, \\ g_2(\mathbf{x}) = g_{c2}(\mathbf{x}) - g_{U2} \leq 0, \end{cases} \end{cases} \quad (4.25)$$

$$\text{where } \begin{cases} x_1 \in (120 \text{ rpm/min}, 180 \text{ rpm/min}, 360 \text{ rpm/min}), \\ 5 \text{ min} \leq x_2 \leq 60 \text{ min}, \\ 0 \text{ min} \leq x_3, x_4 \leq 60 \text{ min}. \end{cases} \quad (4.26)$$

4.6.2 Training of RBFN

The dynamic simulation executes 250 start-up schedules randomly in order to prepare datasets of schedule variables and evaluation functions. The datasets are divided into two parts. The first (80% of the datasets) is used as training data for the RBFN, and the second (20% of the datasets) is used as validation data. The RBFN is trained for all the evaluation functions, f_1 , f_2 , f_3 , g_{c1} , and g_{c2} , on the training data. The results obtained for the validation data are plotted in Fig. 4.7. Figures 4.7a, 4.7b, and 4.7c show the values of f_1 , f_2 , and f_3 , respectively, made dimensionless by the upper limit. Figure 4.7d shows the values of the Chebyshev scalarization function, F ,

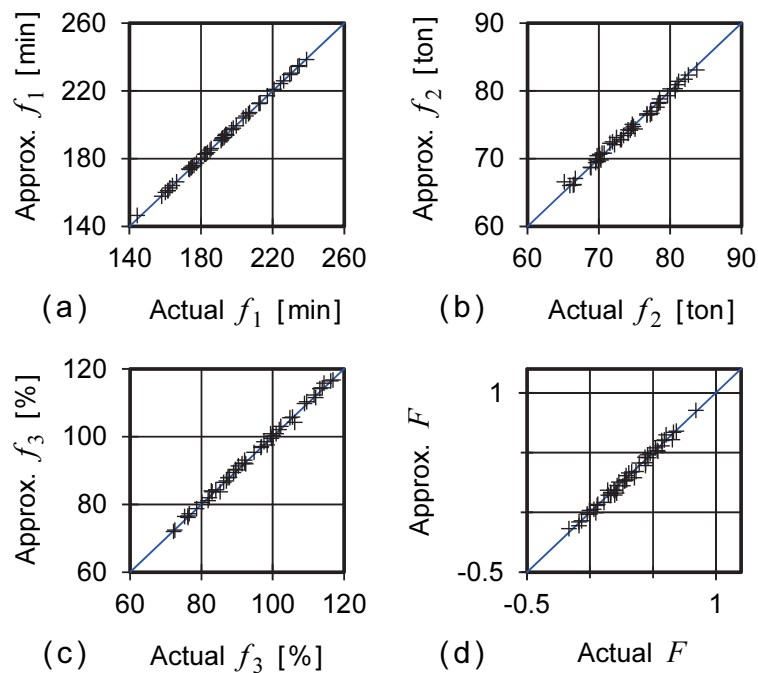


Fig. 4.7 Training results of the RBFN models. This figure shows the comparative results for the value of validation data (Actual) and the approximate function created from the RBFN (Approx.).

Table 4.1 Approximation errors of the RBFN models

Error	f_1	f_2	f_3, g_{c1}	g_{c2}
Average	0.13%	0.49%	0.58%	0.21%
Maximum	1.43%	2.14%	1.92%	0.58%

of Eq. (4.3) to the aspiration level $(\overline{f_1}, \overline{f_2}, \overline{f_3}) = (200 \text{ min}, 74.4 \text{ ton}, 94.9\%)$. The average and maximum values of the approximation error defined by the relative error are summarized in Table 4.1. The approximate functions created from the RBFN agree well with the actual dynamic simulation results.

4.6.3 Multi-objective optimization

As a typical example, the results of the following interactive multi-objective optimization are explained. In this case, the initial temperature of the steam turbine rotors is 180°C . Figure 4.8 shows the result of the following multi-objective optimization processes as a relation between three objective functions in two dimensions. The ideal point (the symbol * in Fig. 4.8) is set to the minimum value of each objective function in the above-mentioned training data. The initial start-up schedule $(x_1, x_2, x_3, x_4) = (180 \text{ rpm/min}, 40.0 \text{ min}, 40.0 \text{ min}, 30.0 \text{ min})$ shown in Fig. 4.9 is obtained as the result of the actual dynamic simulation corresponding to the first aspiration level. This start-up schedule is determined by an expert only through a trial-and-error heuristic approach; hence, it is inadequate. Let the first aspiration level be $(\overline{f_{1,1}}, \overline{f_{2,1}}, \overline{f_{3,1}}) = (200 \text{ min}, 74.4 \text{ ton}, 94.9\%)$ (the symbol \square in Fig. 4.8). Then,

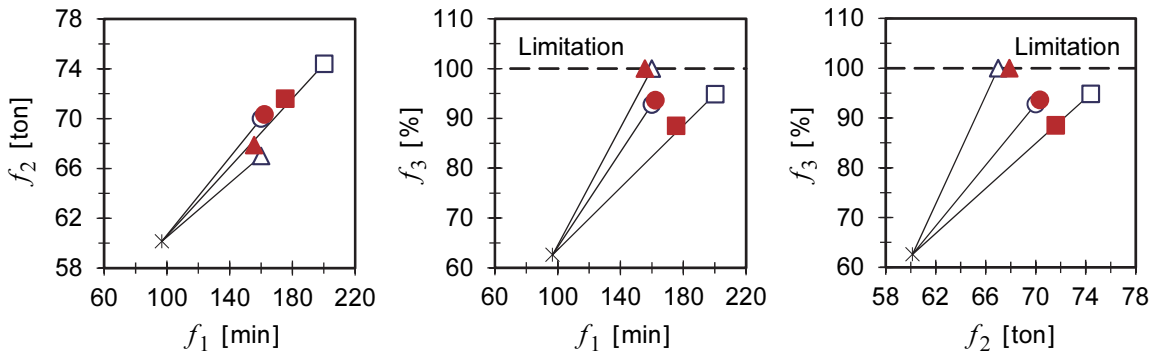


Fig. 4.8 Results of the multi-objective optimization processes as a relation between three objective functions in two dimensions. In this figure, the dashed line represents the limitation value of the normalized thermal stress of the steam turbine rotors. The symbols \square and \blacksquare show the first aspiration level and the first optimal solution, respectively. The symbols \circ and \bullet show the second aspiration level and the second optimal solution, respectively. The symbols \triangle and \blacktriangle show the third aspiration level and the third optimal solution, respectively. The symbol * is the ideal point.

the first optimal solution is $(f_{1,1}, f_{2,1}, f_{3,1}) = (175 \text{ min}, 71.6 \text{ ton}, 88.5\%)$ (the symbol ■ in Fig. 4.8); here, all the criteria are improved. Now, suppose that the user wants to further decrease the value of the start-up time (f_1) and that of the fuel consumption rate (f_2) significantly. Since the present solution is already one of the Pareto-optimal solutions, it is impossible to improve all the criteria. Therefore, suppose that the user agrees to relax the thermal stress of the steam turbine rotors (f_3) and let the second aspiration level be $(\overline{f_{1,2}}, \overline{f_{2,2}}, \overline{f_{3,2}}) = (160 \text{ min}, 70.0 \text{ ton}, 92.8\%)$ (the symbol ○ in Fig. 4.8). Then, the second optimal solution is $(f_{1,2}, f_{2,2}, f_{3,2}) = (162 \text{ min}, 70.3 \text{ ton}, 93.6\%)$ (the symbol ● in Fig. 4.8). Although the obtained solution does not completely attain the aspiration level of the start-up time and the fuel consumption rate, it should be noted that the solution is an improvement over the previous one. The improvement in f_1 and f_2 does not match the requirement of the user because the amount of relaxation in f_3 is not sufficiently large to compensate for the improvement in f_1 and f_2 . Moreover, suppose that the user wants to further decrease the value of f_1 slightly, and that of f_2 significantly, and therefore, the user decides to relax f_3 up to the upper limit and lets the third aspiration level be $(\overline{f_{1,3}}, \overline{f_{2,3}}, \overline{f_{3,3}}) = (160 \text{ min}, 67.0 \text{ ton}, 100\%)$ (the symbol ▲ in Fig. 4.8). Then, the third optimal solution is $(f_{1,3}, f_{2,3}, f_{3,3}) = (156 \text{ min}, 67.9 \text{ ton}, 100\%)$ (the symbol ▲ in Fig. 4.8). The improvement in f_2 does not match the requirement of the user because f_3 reaches the upper limit. This solution is an operating limitation of this plant, where both the start-up time and fuel consumption rate cannot be reduced any

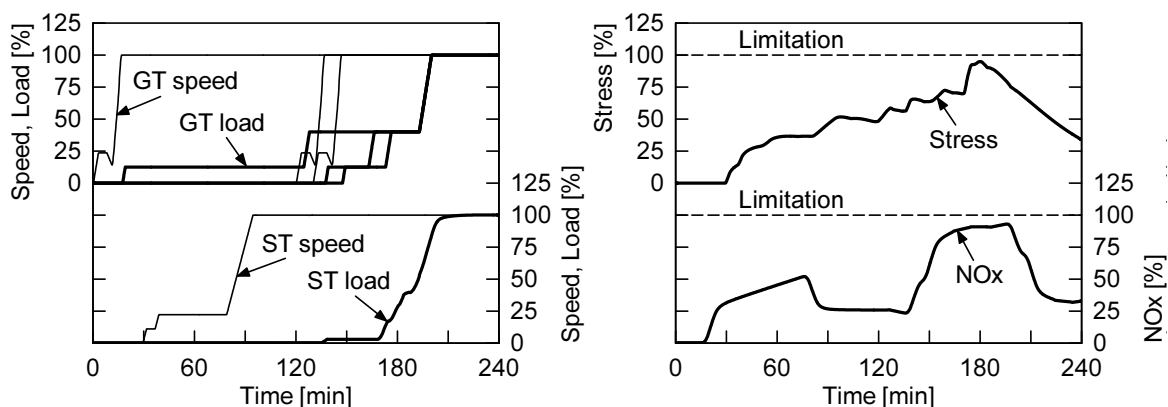


Fig. 4.9 Initial start-up schedule corresponding to the first aspiration level (the symbol ■ in Fig. 4.8).

further.

The optimal start-up schedule of the second solution $(x_{1,2}, x_{2,2}, x_{3,2}, x_{4,2}) = (360 \text{ rpm/min}, 5.0 \text{ min}, 22.4 \text{ min}, 53.6 \text{ min})$ demonstrated in Fig. 4.10 is the result of an actual dynamic simulation. In Figs. 4.9 and 4.10, GT denotes the gas turbines; ST, the steam turbine; Stress, the normalized thermal stress of the steam turbine rotors; and NO_x , the normalized NO_x emission rate from the plant. Here, the dashed lines represent the respective limitation values. The results are summarized in Table 4.2, where Solution (RBFN) represents the results of the RBFN models, and Solution (Actual) represents the results of the actual dynamic simulation. All these solutions are the

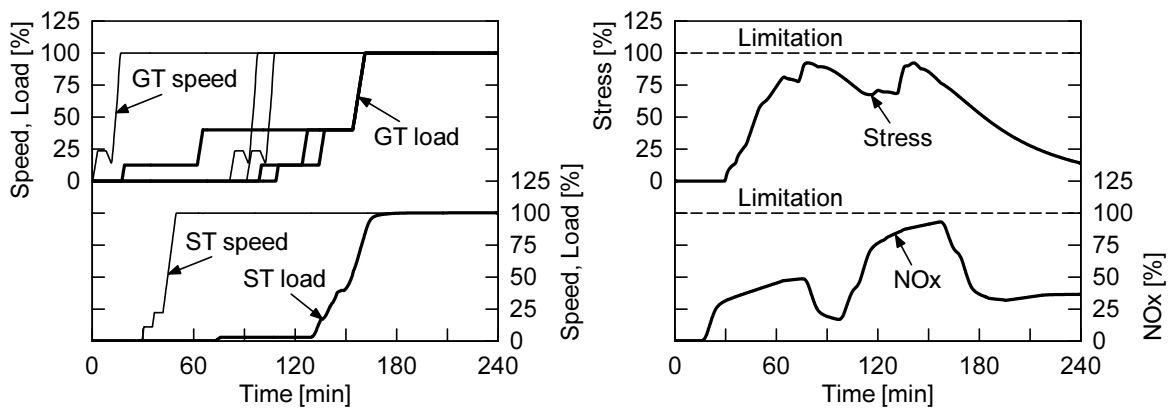


Fig. 4.10 Optimal start-up schedule obtained for the second aspiration level (the symbol ● in Fig. 4.8).

Table 4.2 Simulation results of the objective functions

	f_1	f_2	f_3
Aspiration level #1	200 [min]	74.4 [ton]	94.9%
Solution #1 (RBFN)	175 [min]	71.6 [ton]	88.5%
Solution #1 (Actual)	175 [min]	71.9 [ton]	87.7%
Aspiration level #2	160 [min]	70.0 [ton]	92.8%
Solution #2 (RBFN)	162 [min]	70.3 [ton]	93.6%
Solution #2 (Actual)	161 [min]	69.8 [ton]	92.3%
Aspiration level #3	160 [min]	67.0 [ton]	100%
Solution #3 (RBFN)	156 [min]	67.9 [ton]	100%
Solution #3 (Actual)	156 [min]	67.6 [ton]	100%

Pareto-optimal solutions that are start-up schedules closest to each aspiration level and satisfy the operational constraints.

4.6.4 Computing time

The computing time depends on the size of the training data. In this case, it takes approximately 90 s (details: training of the RBFN takes approximately 30 s, and searching by using the GRGA takes approximately 60 s) to obtain an optimal start-up schedule as per the aspiration level, using a personal computer (IntelTM PentiumTM M processor 2.13 GHz). This is sufficiently quick for practical use such as daily start-up schedule planning.

4.7 Discussions

The first advantage of this system is that it determines a solution that satisfies the user with the minimum possible number of interactions. In Fig. 4.8, the first aspiration level is determined by an expert only through a trial-and-error heuristic approach. This aspiration level is quite different from that indicated by the optimal solution because the user cannot obtain the accurate solution space. However, the second aspiration level can be roughly directed near the optimal solution because the user has to consider only the trade-off by considering the first optimal solution. In addition, the user determines the operating limitation of this plant from only three interactions. Thus, it is inferred that the user can appropriately understand the solution space by performing a small number of interactions by using this system.

The second advantage of this system is its ability to obtain quick solutions. Since the high-accuracy plant simulator consumes considerable amount of time. The system requires an unrealistic amount of time for obtaining the optimal solution. To overcome this problem, in the proposed method, the optimal solution can be obtain within a short computing time by constructing the approximate functions with the aid of the RBFN from the plant simulator and optimizing with these approximate functions.

Therefore, this system can support the decision of the user for a flexible start-up schedule because the optimal solution can be obtain with a small number of interactions and within a short computing time by employing a multiple-criteria decision-making process.

4.8 Economic evaluation

Against the background of recent developments in energy liberalization and the increasing renewable energy sources such as wind and solar radiation, there is a greater need to move away from the conventional operational methods for thermal power plant wherein everything is preplanned, and instead to become more flexible in the thermal power plant operations and to be able to respond to the changes in the market. In this section, we consider the economic benefit of the plants that are operated in a flexible manner by responding to the changes in fuel prices in the market by seeking for solutions using the aforementioned multi-objective optimization system for plant operations.

Fuel cost fluctuates with uncertainty based on such changes in factors such as material cost and the exchange rate of currency. By applying multi-objective optimization to the start-up schedule in response to such fluctuations in price, we adjusted the amount of fuel consumption rate at start-up. In other words, rapid start-up was achieved when the fuel cost was high in order to minimize the fuel consumption rate, and the aspiration level of the steam turbine thermal stress was set to be on the high side. In contrast, when the fuel cost was low, despite the greater fuel consumption rate during the start-up process, the steam turbine thermal stress would be curbed by starting up the plant slowly. In addition, the steam turbine thermal stress during each start-up process would be tallied with the equipment lifetime consumption rate so that the life cycle of the steam turbine could be managed within certain parameters.

These flexible operations were evaluated for their economic viability using the Monte Carlo simulation method (Kamiyama and Satoh, 1997). In this work, the price of the fuel (natural gas) was set as 30 000 yen/ton, assuming a standard deviation of 3000 yen/ton as a normal distribution of market fluctuations. The number of plant start-ups was assumed as 100 times to find the total sum of the fuel costs during start-ups. The results of the Monte Carlo simulations (30 000 trials) are shown in Fig. 4.11. Figure 4.11a shows the result of not using the flexible start-up method (i.e., the conventional method), where the start-up schedule was fixed as the equivalent of the first aspiration level (the symbol \square in Fig. 4.8). Figure 4.11b shows the result of flexible operations after using our system for multi-objective optimization (i.e., the flexible start-up method). Note that the equipment lifetime consumption rate attributed to steam turbine thermal stress was optimized so that both were identical. The economic benefit of using

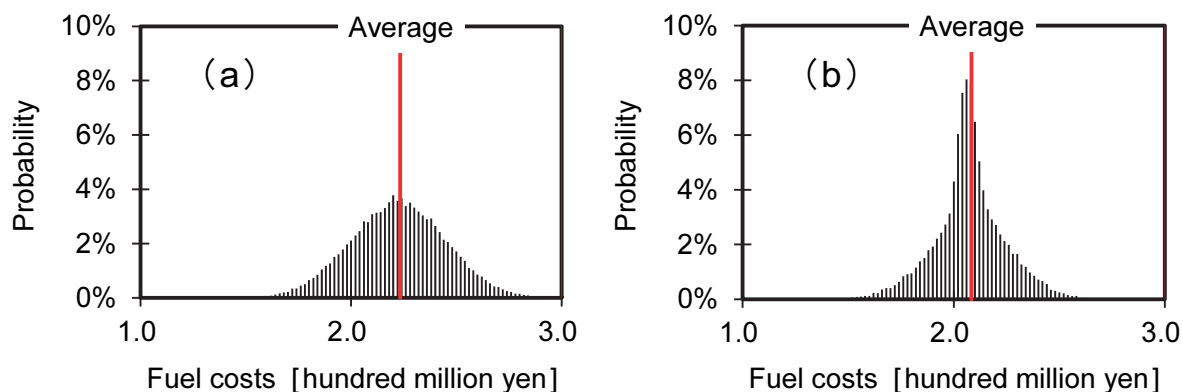


Fig. 4.11 Results of the Monte Carlo simulations. (a) This figure shows the result of not using the flexible start-up method, where the start-up schedule was fixed by an expert only through a trial-and-error heuristic approach. (b) This figure shows the result of flexible operations after using our system for multi-objective optimization.

this system is that while the start-up fuel cost (average value) using the conventional start-up method was 223 million yen, that when using the flexible start-up method was 208 million yen, achieving a cost reduction of 6.7%. Furthermore, the flexible start-up method reduced the fuel cost with a probability of 84% reliability, compared to the average value of the conventional start-up method. Additionally, with the flexible start-up method, the probability distribution was concentrated near the average value, demonstrating that the flexible method offers the advantage of its ability to improve robustness against price fluctuation.

4.9 Conclusions

A cooperative human–machine system is proposed to optimize from among multiple objectives for the start-up schedule in a thermal power plant. The proposed system consists of a dynamic simulation, an artificial neural network, and an interactive multi-objective programming technique. This plant operation system can be used to

1. optimize the start-up schedule based on multi-objective evaluation, and
2. achieve an optimal and flexible start-up schedule with a reasonable computing time and calculation accuracy through human–computer interactions.

The proposed system yielded good results in all our experiments. The main results

are as follows:

1. In this study, a novel intelligent optimization method using an artificial neural network and a genetic algorithm has been developed to realize a satisficing trade-off method in an interactive multi-objective programming technique.
2. The system was applied to a dynamic simulation for a combined cycle power plant, to optimize from among multiple objectives, based on varying daily requirements. The results reveal that optimal and flexible start-up schedules can be obtained within a reasonable computing time with acceptable calculation accuracy.
3. We proposed the use of this system for flexible plant operations, and, through Monte Carlo simulations, demonstrated the economic benefits of the system.

Finally, some issues that will be considered in the following chapters are as follows:

1. The proposed system deals with a dynamic optimization problem by transposing it to a static optimization problem. Thus, the entire start-up schedule has to be initially determined offline. Although the requirement of preplanning the plant operation by operators (or designers) does not necessarily impair the usefulness of the proposed system, the establishment of an online real-time control method is expected to improve plant operability and controllability.
2. During multiple-criteria decision-making of plant operations, apart from the multi-objective optimization methods, the evaluation methods of the obtained diverse optimal solutions are also important.

The above two issues are considered in Chapters 5 and 6 of Part II, respectively.

References

- Arakawa, M. and Ishikawa, H., Development of adaptive range genetic algorithms considering inheritance of searching range, Proceedings of the 2002 ASME Design Engineering Technical Conferences (DETC-DAC2002), CD-ROM, Paper number DETC2002/DAC-34081, (2002), Montreal, Quebec, Canada.
- Bausa, J. and Tsatsaronis, G., Dynamic optimization of startup and load-increasing processes in power plants — Part I: Method, Journal of Engineering for Gas Turbines and Power, Transactions of the ASME, Vol. 123, No. 1, (2001a), pp. 246–250.
- Bausa, J. and Tsatsaronis, G., Dynamic optimization of startup and load-increasing

- processes in power plants—Part II: Application, *Journal of Engineering for Gas Turbines and Power*, Transactions of the ASME, Vol. 123, No. 1, (2001b), pp. 251–254.
- Kamiya, A., Kawai, K., Ono, I. and Kobayashi, S., Adaptive-edge search for power plant start-up scheduling, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 29, No. 4, (1999), pp. 518–530.
- Kamiyama, S. and Satoh, A., *Monte Carlo Simulation*, (1997), Asakura Publishing Co., Ltd., Tokyo, Japan (in Japanese).
- Matsumoto, H., Ohsawa, Y., Takahashi, S., Akiyama, T. and Ishiguro, O., An expert system for start-up optimization of combined cycle power plants under NO_x emission regulation and machine life management, *IEEE Transactions on Energy Conversion*, Vol. 11, No. 2, (1996), pp. 414–422.
- Matsumoto, H., Ohsawa, Y., Takahashi, S., Akiyama, T., Hanaoka, H. and Ishiguro, O., Startup optimization of a combined cycle power plant based on cooperative fuzzy reasoning and a neural network, *IEEE Transactions on Energy Conversion*, Vol. 12, No. 1, (1997), pp. 51–59.
- Ordys, A., Pike, A. W., Johnson, M. A., Katebi, R. M. and Grimble, M. J., *Modelling and Simulation of Power Generation Plants*, (1994), Springer-Verlag London, Ltd., London, the United Kingdom.
- Nakayama, H., Aspiration level approach to interactive multi-objective programming and its applications, In: *Advances in Multicriteria Analysis* (Pardalos, P. M., Siskos, Y. and Zopounidis, C., eds.), (1995), pp. 147–174, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Nakayama, H., Arakawa, M. and Sasaki, R., Simulation-based optimization using computational intelligence, *Optimization and Engineering*, Vol. 3, (2002), pp. 201–214, Kluwer Academic Publishers, Dordrecht, the Netherlands.

Chapter 5

Intelligent multi-objective model predictive control applied to steam turbine start-up

Abstract

This study proposes an intelligent multi-objective model predictive control method in which an artificial neural network and a genetic algorithm are used to realize satisficing decision-making, which is an interactive multi-objective programming technique. We considered model predictive optimization under a dynamic environment with multiple objectives. To predict nonlinear function forms with dynamic plant characteristics, we applied a recurrent radial basis function network, which is a type of artificial neural network. For optimization with multiple objectives, we applied a satisficing trade-off method along with metaheuristic optimization in the form of genetic algorithms. The features of this control method are as follows. (1) Several conflicting control objectives can be optimized in online control based on multi-objective evaluation through human–computer interactions and (2) an optimal and flexible plant control can be performed within a restrained practical computing time for real-time applications, with acceptable control quality using online adaptive model prediction. This study demonstrates the success of model prediction using computational intelligence combined with an interactive optimization technique for multi-objective model predictive control problems by applying the proposed method to steam turbine start-up control with multiple objectives consisting of the start-up time and rotor thermal stress of the steam turbine. The dynamic simulation results showed an effective control performance within a reasonable computing time.

Keywords

Genetic algorithm, Model predictive control, Multi-objective optimization, Power plant, Radial basis function network, Real-time optimal control, Recurrent neural network, Satisficing trade-off method, Steam turbine, Thermal stress

5.1 Introduction

When function forms in mathematical models cannot be given explicitly in terms of design variables, the function values are usually determined by performing numerical or real experiments. Since these experiments are often expensive, it is important to develop techniques for obtaining solutions with as few experiments as possible. To this end, model predictive optimization is intended to find optimal solutions in parallel with predicting the function forms in mathematical models. Metamodeling (Wang and Shan, 2007) and surrogate-based optimization (also called sequential approximate optimization; Nakayama et al., 2009) methods apply the same concepts. Several methods have been developed for this purpose, of which the response surface (Myers et al., 2016), design of experiments (Montgomery, 2017), Kriging (Cressie, 1988), artificial neural network (Papadrakakis et al., 1998), and support vector machine (Yun and Nakayama, 2016) methods are well known but mainly handle static optimization. For dynamic optimization problems, the model predictive control method (Maciejowski, 2002) has been developed using ideas analogous to those mentioned above. However, the objectives of almost all conventional model predictive control studies have been to optimize a single-objective function (e.g., only time-optimal control problem). Research on multiple-objective control problems has been restricted in its applications (Bien et al., 1999; Bemporad and Muñoz de la Peña, 2009; Zavala and Flores-Tlacuahuac, 2012; Núñez et al., 2014). In cases with multi-objective functions, it is very difficult to adjust the weights of the objective functions using a trial-and-error heuristic approach. Moreover, the computing time increases drastically upon generation of a Pareto frontier in these cases, making practical control problems impossible. To avoid these assignments, it is necessary to determine a priority beforehand among multiple objectives (Rojas et al., 2013; He et al., 2015).

To balance computing time and accuracy, we proposed a novel intelligent optimization algorithm that combines surrogate-based optimization and the satisficing trade-off method (Nakayama, 1995), which is an interactive multi-objective programming technique (see Chapter 4 for details). Furthermore, in Chapter 4, we developed a multi-objective optimization system for thermal power plant operations wherein an operator can interact with a computer to optimize a start-up schedule based on multi-objective evaluation. However, since this system deals with a dynamic optimization problem by transposing it to a static optimization problem, it must initially

determine the entire start-up schedule offline. That is, although this system can optimize multi-objective functions, it cannot be applied to online optimal control problems. Such systems cannot cope with unexpected changes in thermodynamic state values resulting from disturbances during plant operations. Several online optimal control methods for steam turbine start-up using model predictive control have been proposed (Nakai et al., 1996; D'Amato, 2006; Matsumoto et al., 2010; Biyik et al., 2012; Tica et al., 2012; Miyazawa et al., 2015; Schuermann et al., 2016); however, the objectives of these studies have been to optimize only a single-objective time-optimal control function. Typically, steam turbine start-up control for a thermal power plant involves several conflicting objective functions, such as those for faster start-up, less thermal stress due to reduced lifetime consumption of the steam turbine rotors, reduced fuel consumption, and reduced auxiliary power consumption. Given recent developments in electricity deregulation and increases in renewable energy, there is a greater need for thermal power plant operations to become more flexible and able to respond to market changes.

Therefore, we extended the optimization algorithm discussed in Chapter 4 for online optimal control. In this study, for a dynamic environment with multiple objectives, we propose a multi-objective model predictive control method using computational intelligence, which is specifically a recurrent radial basis function network (RRBFN) and satisficing trade-off method. This method maintains the framework of human–computer interaction features studied in Chapter 4 and can be applied to online optimal control from among multiple objectives within a reasonable computing time. The proposed method is applied herein to dynamic simulations of a 600 MW steam power plant, and its effectiveness is demonstrated using the example of steam turbine start-up control with multiple objectives consisting of the start-up time and rotor thermal stress of the steam turbine.

5.2 Proposed multi-objective model predictive control

Consider a dynamic optimization problem. Following the optimal control theory, let u denote the control input and $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ denote the vector of state variables. Here, f represents the objective function (evaluation function) and $\mathbf{g} = [g_1, g_2, \dots, g_s]^T$ represents the vector of the constraining conditions. The theory assumes the following mathematical model:

$$\left\{ \begin{array}{l} \min_{\mathbf{x}, u} f = \varphi[\mathbf{x}(T)] + \sum_{k=0}^{T-t-1} V[\mathbf{x}(t+k), u(t+k), t+k], \\ \text{subject to} \left\{ \begin{array}{l} \mathbf{x}(t+k+1) = v[\mathbf{x}(t+k), u(t+k), t+k], \quad \mathbf{x}(t) = \mathbf{x}_o, \\ \mathbf{g}[\mathbf{x}(t+k), u(t+k), t+k] \leq 0, \quad k = 0, 1, \dots, T-t-1, \end{array} \right. \end{array} \right. \quad (5.1)$$

where $T \geq t$, t represents the present time; T , the terminal time; and \mathbf{x}_o , the vector value of actual observed state variables at time t .

If the function form in the above model is given explicitly, then some techniques from optimal control theory can be applied. However, we assume that some function forms cannot explicitly be given due to their nonlinear models for complex system dynamics. In this case, we predict some future states $\mathbf{x}(t+1)$, $\mathbf{x}(t+2)$, \dots , $\mathbf{x}(t+p_1)$ for the control input sequence $u(t+1)$, $u(t+2)$, \dots , $u(t+p_2)$, where $p_1 \geq p_2$, p_1 represents the prediction period, and p_2 represents the control period. Our aim is to determine the optimal control input sequence $u(t+k)$ over $[t, T]$.

Suppose that the problem has the following multiple objectives:

$$\mathbf{f} = [f_1, f_2, \dots, f_r]^T. \quad (5.2)$$

For example, in the case of a thermal power plant, those objectives are the start-up time, lifetime consumption rate, fuel consumption rate, and pollutant emission rate of the machine. To obtain a final decision for those multi-objective problems, we apply the satisficing trade-off method (Nakayama, 1995) which is an aspiration-level-based method (to be described in Section 5.3.1). For predicting the future states, we apply an artificial neural network technique, namely, an RRBFN (to be described in Section 5.3.3.1). The original radial basis function network (RBFN) provides less training computing time than other artificial neural networks. Further, the optimal control input sequence closest to the aspiration level is searched in conjunction with the RRBFN prediction model using a genetic algorithm (GA) (which we describe in Section 5.3.2).

Figure 5.1 illustrates the structure of the multi-objective model predictive control system developed in this study. This system applies an interactive multi-objective programming technique called the satisficing trade-off method as its framework via human-computer interactions and integrates the intelligence of the computer with the knowledge of the user. Figure 5.2 shows an example of the discrete model predictive

control scheme. Referring to Figs. 5.1 and 5.2, we explain the procedure of the proposed method with discrete-time periods.

Step 1. Controller: The RRBFN prediction model is trained to predict the dynamic system model based on observed past states \mathbf{x} and control input sequence u . The training datasets for the RRBFN prediction model are as follows:

$$\begin{cases} \mathbf{x}(k-p_3), \mathbf{x}(k-p_3+1), \dots, \mathbf{x}(k), \\ u(k-p_3), u(k-p_3+1), \dots, u(k), \quad k = t-p+1, t-p+2, \dots, t, \end{cases} \quad (5.3)$$

where p denotes the depth of the sampling training data, and p_3 represents the past sample period.

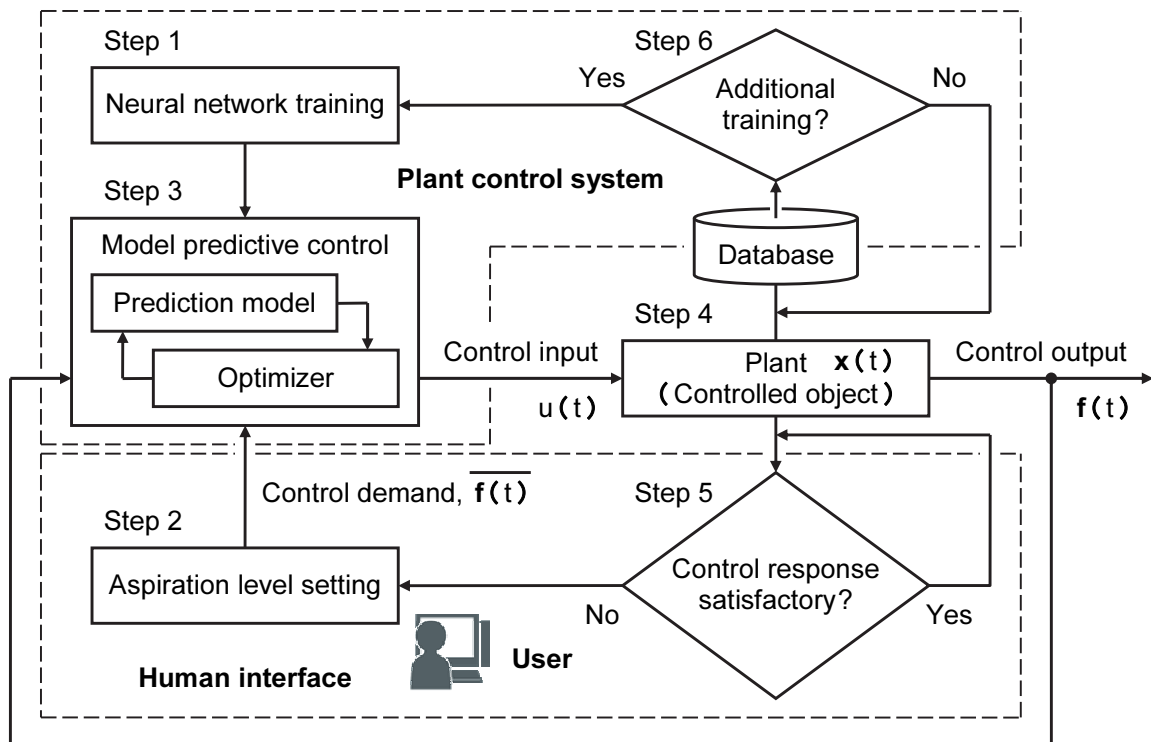


Fig. 5.1 Structure of the proposed multi-objective model predictive control system. A cooperative human–machine system is proposed to optimize control input $u(t)$ by online control to aspiration level $\overline{f}(t)$ from among multiple objectives $f(t)$ at each sampling time t . The training datasets based on observed states $\mathbf{x}(t)$ for the RRBFN prediction model are stored in the database.

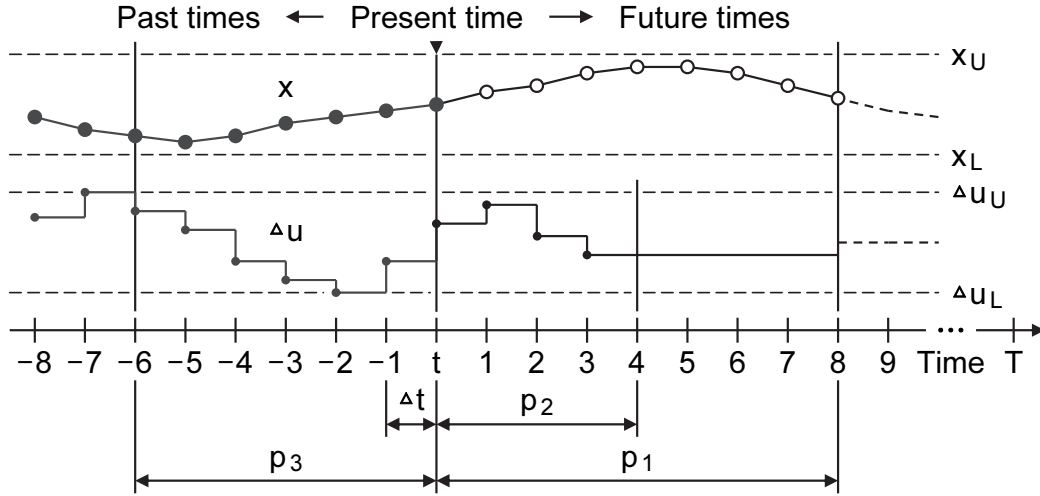


Fig. 5.2 Discrete model predictive control scheme. This figure is an example, with a prediction period p_1 of 8 samples, control period p_2 of 4 samples, and past sample period p_3 of 6 samples, where t represents the present time; T , the terminal time; Δt , one sampling period; Δu_L and Δu_U , the lower and upper bounds, respectively, for control input rate Δu ; and x_L and x_U , the lower and upper bounds, respectively, for state x . These parameters are adjusted based on control problems.

Step 2. Human interface: The user sets the aspiration level (desired values) \bar{f}_i corresponding to each objective function f_i ($i = 1, 2, \dots, r$) as follows:

$$\bar{\mathbf{f}} = [\bar{f}_1, \bar{f}_2, \dots, \bar{f}_r]^T. \quad (5.4)$$

Step 3. Controller: The optimal solution for control input rate sequence Δu^{best} (or control input sequence u^{best}) closest to $\bar{\mathbf{f}}$ is sought in conjunction with the RRBFN prediction model using a GA:

- i. N initial individuals of control input rate sequence Δu_j are generated randomly as follows:

$$\Delta u_{L1} \leq \Delta u_j(t+k) \leq \Delta u_{U1}, \quad j = 1, 2, \dots, N, \quad k = 0, 1, \dots, p_2 - 1, \quad (5.5)$$

$$\text{with } \Delta u_j(t+k) = [\Delta u_{L2} \leq \Delta u_j(t+p_2-1) \leq \Delta u_{U2}], \quad (5.6)$$

$$j = 1, 2, \dots, N, \quad k = p_2, p_2 + 1, \dots, p_1 - 1,$$

where Δu_{L1} and Δu_{U1} represent the lower and upper bounds, respectively, during control period p_2 ; and Δu_{L2} and Δu_{U2} , the lower and upper bounds, respectively, during prediction period p_1 beyond p_2 .

- ii. The future state values of dynamic system model \mathbf{x}_j for each control input sequence u_j are predicted using the RRBFN model during prediction period p_1 :

$$\begin{cases} \mathbf{x}_j(t+k+1) = \hat{v}[\mathbf{x}_j(t+k), u_j(t+k), t+k], \\ u_j(t+k+1) = u_j(t+k) + \Delta u_j(t+k), \\ j = 1, 2, \dots, N, \quad k = 0, 1, \dots, p_1 - 1, \end{cases} \quad (5.7)$$

where \hat{v} denotes the predicted function of dynamic system model v in Eq. (5.1), and $u_j(t)$ and $\mathbf{x}_j(t)$ are set to actual observed values $u_j(t) = u_o$ and $\mathbf{x}_j(t) = \mathbf{x}_o$, respectively.

- iii. For future state values of dynamic system model \mathbf{x}_j , for each control input sequence u_j , the value of the auxiliary scalar function of the satisficing trade-off method is estimated for each F_j :

$$\begin{aligned} F_j = \max_{1 \leq i \leq r} \left[\frac{f_i(\mathbf{x}_j, u_j) - \bar{f}_i}{\bar{f}_i - f_i^*} \right] + \alpha \sum_{i=1}^r \frac{f_i(\mathbf{x}_j, u_j)}{\bar{f}_i - f_i^*} \\ + \sum_{\ell=1}^s \beta_\ell \max_{\ell} \left[0, g_\ell(\mathbf{x}_j, u_j) \right]^2, \quad j = 1, 2, \dots, N, \end{aligned} \quad (5.8)$$

where f_i^* is an ideal point, and u_j and \mathbf{x}_j are $u_j = u_j(t+1), u_j(t+2), \dots, u_j(t+p_1)$ and $\mathbf{x}_j = \mathbf{x}_j(t+1), \mathbf{x}_j(t+2), \dots, \mathbf{x}_j(t+p_1)$, respectively. α is usually set to a sufficiently small positive number (such as 10^{-6}). The third term on the right side of Eq. (5.8) is the penalty function, where penalty parameter β_ℓ is a sufficiently large positive number for each constraining condition g_ℓ .

- iv. The individuals in control input rate sequence Δu_j are evaluated using the value of F_j , and N new individuals of Δu_j are generated through natural selection and genetic operators.
- v. Steps ii–v are repeated until a stop condition—for example, the number of iterations—occurs.
- vi. The best control input rate sequence Δu^{best} that minimizes the value of F_j is decided.

Step 4. Plant: Only the best control input rate $\Delta u^{\text{best}}(t)$ at time t is extracted, and the control input $u(t+1) = u(t) + \Delta u^{\text{best}}(t) = u_o + \Delta u^{\text{best}}(t)$ is input to the plant. Actual state values $\mathbf{x}(t+1)$ according to $u(t+1)$ are observed, and u_o and \mathbf{x}_o are set to actual observed values $u_o = u(t+1)$ and $\mathbf{x}_o = \mathbf{x}(t+1)$, respectively. Stop if $t+1 \geq T$; otherwise, perform the update $t \leftarrow t+1$, and return to Step 3. However, if the user wants to modify the aspiration level, go to Step 5, and/or if the RRBFN predictive model requires additional training, go to Step 6.

Step 5. Human interface: The user judges whether the observed control responses are satisfactory. If the user is not satisfied, the user modifies the aspiration level. Subsequently, Steps 2–5 are repeated until the user obtains agreeable control responses. If the user is satisfied, return to the end of Step 4.

Step 6. Controller: If approximation errors between actual observed values and predicted values are large, the RRBFN prediction model is further trained (offline or online). Subsequently, Steps 1–6 are repeated until the approximation errors are reduced. If the approximation errors are small, return to the end of Step 4.

5.3 Intelligent optimization method

In this section, we detail a new intelligent optimization method for multi-objective model predictive control. This method combines an RRBFN with a GA and employs the satisficing trade-off method to enable multi-objective optimization of plant controls. The reason that we focus on the satisficing trade-off method is that it does not require the generation of a Pareto frontier or priorities and weight adjustments for multiple objectives, as was necessary in conventional multi-objective control studies. Moreover, in model predictive control, technology for predicting dynamic plant characteristics is important. To achieve real-time control performance, we devised a recurrent neural network that extends existing RBFNs and integrates an efficient data degeneracy method using cluster analysis to shorten its learning time. In addition, the accuracy of this prediction model was verified using actual thermal power plant operation data.

5.3.1 Satisficing trade-off method

Multi-objective optimization implies the simultaneous optimization of multiple objective functions that have a trade-off relationship with one another under a given condition. This problem is typically expressed as follows:

$$\begin{cases} \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_r(\mathbf{x})]^T, \\ \text{subject to } g_\ell(\mathbf{x}) \leq 0, \quad \ell = 1, 2, \dots, s, \end{cases} \quad (5.9)$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is a vector of design variables, f represents the objective functions, and g specifies the constraining conditions.

Because it is difficult to find the solution if the problem is expressed in such a vector format, the problem is scalarized for ease of solving. In this case, when the formula is scalarized by using the satisficing trade-off method, which is an interactive multi-objective programming technique employing the aspiration-level-based method. See Chapter 4, Section 4.5.1 for more details. The following auxiliary scalar optimization of Eq. (5.11) is obtained, using the Chebyshev scalarization function F of Eq. (5.10). Here, the operational constraints can be included in Eq. (5.11) as the penalty function.

$$F(\mathbf{x}) = \max_{1 \leq i \leq r} w_i [f_i(\mathbf{x}) - \bar{f}_i] + \alpha \sum_{i=1}^r w_i f_i(\mathbf{x}), \quad (5.10)$$

$$\min_{\mathbf{x}} F(\mathbf{x}) + \sum_{\ell=1}^s \beta_\ell \max_{\ell} [0, g_\ell(\mathbf{x})]^2, \quad (5.11)$$

where α is usually set to a sufficiently small positive number (such as 10^{-6}), and the penalty parameter β_ℓ is a sufficiently large positive number for each constraining condition g_ℓ . Weight w_i can be expressed as follows:

$$w_i = \frac{1}{\bar{f}_i - f_i^*}, \quad (5.12)$$

where \bar{f}_i is the aspiration level (desired values) of the objective functions, and f_i^* is an ideal point, usually given such that $f_i^* \leq \min [f_i(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}]$, where \mathbf{X} is the set of all feasible solutions.

5.3.2 GA with blend alpha crossover

Since we focused on continuous-variable optimization, Eq. (5.11) was minimized using a real-coded GA. Crossover is the most important operator in real-coded GAs. We

discuss the application of a blend alpha crossover (BLX-a) (Eshelman and Schaffer, 1993) in this study. BLX-a combines two parents P^1 and P^2 to generate two offspring C^1 and C^2 with a crossover probability by sampling new values in the range $[\min_i - I \cdot a, \max_i + I \cdot a]$ at each gene i . Here, \min_i and \max_i are the smallest and largest values, respectively, of the two parents at each gene i , and $I = \max_i - \min_i$. Parameter a is usually set to 0.5. The new offspring can be adjusted as necessary to fit within the following bounds:

$$C_i^b = (\min_i - I \cdot a) + \gamma^b \cdot |(\max_i + I \cdot a) - (\min_i - I \cdot a)|, \quad b = 1, 2, \quad (5.13)$$

where γ^b is a random number between 0 and 1.

The numerical algorithm for GA with BLX-a that is discussed in this study has the following steps:

Step 1. Initialization: Randomly generate N initial parent generation individuals.

Step 2. Crossover: Generate offspring using BLX-a.

Step 3. Evaluation: The fitness function value is in the Chebyshev scalarization function F of Eq. (5.10). The fitness function values of each individual are calculated with the penalty function of the constraining conditions, as in Eq. (5.11).

Step 4. Selection: When selecting the fittest individuals to pass to the next generation, elite retention selection is used. That is to say, N elite individuals with small fitness function values survive in the next generation. Further, Steps 2–4 are repeated until the maximum number of iterations or maximum computing time is reached. Here, we define the maximum computing time to be within one sampling period (optimization cycle) Δt , as shown in Fig. 5.2.

5.3.3 Model prediction using an RBFN

An RBFN is a type of artificial neural network that consists of several radial basis functions (Nakayama et al., 2002) and contains three layers: input, hidden, and output layers. The input values are each assigned to a neuron in the input layer and passed directly to the hidden layer without weights. A Gaussian function is used as an activation function for the hidden neurons. Subsequently, linear output weights are added between the hidden and output layers. See Chapter 4, Section 4.5.2 for more details. The overall input–output mapping equation of an RBFN is presented as follows:

$$O'(\mathbf{x}) = \sum_{j=1}^q w_j h_j(\mathbf{x}), \quad (5.14)$$

$$\text{where } h_j(\mathbf{x}) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{c}_j\|^2}{R_j^2}\right) \quad (5.15)$$

and \mathbf{c}_j is the center of the hidden neurons; h_j , the Gaussian function; O' , the output values; q , the number of hidden neurons; R_j , the radial parameter; w_j , the weights; $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, the vector of input values; and $\|\cdot\|$, the Euclidean norm.

Our proposed method generates approximate functions by combining the polynomial functions and RBFN as follows:

$$O(\mathbf{x}) = \zeta(\mathbf{x}) + O'(\mathbf{x}), \quad (5.16)$$

where O represents the approximate functions; O' , the RBFN output values from Eq. (5.14); and ζ , the polynomial functions. The polynomial functions are used as base functions, and the RBFN estimates errors from them. This method can improve the accuracy near the side constraints of the design variables, where the training values become sparse.

5.3.3.1 Proposed RRBFN prediction model

We propose an RRBFN, shown in Fig. 5.3, which was applied to predict the dynamic plant characteristics in this study. The RRBFN training method is similar to that of RBFNs, as described above, although the training values are given by the set of time-series data. The output values are predictive values at time $t + 1$:

$$O_1(k), O_2(k), \dots, O_{n_3}(k), \quad k = t + 1. \quad (5.17)$$

The input values are divided into the following three types:

1. time-dependent dynamic state values xd from time $t - p_3 + 1$ to $t + 1$,
2. less/non-time-dependent static state values xs at time $t + 1$, and
3. time-dependent feedback predictive values (output values) O from time $t - p_3 + 1$ to t .

$$\begin{cases} xd_1(k), xd_2(k), \dots, xd_{n_1}(k), & k = t - p_3 + 1, \dots, t, t + 1, \\ xs_1(k), xs_2(k), \dots, xs_{n_2}(k), & k = t + 1, \\ O_1(k), O_2(k), \dots, O_{n_3}(k), & k = t - p_3 + 1, \dots, t - 1, t, \end{cases} \quad (5.18)$$

where p_3 is the past sampling period, and n_1 represents the number of dynamic state values; n_2 , the number of static state values; n_3 , the number of output values; O , the

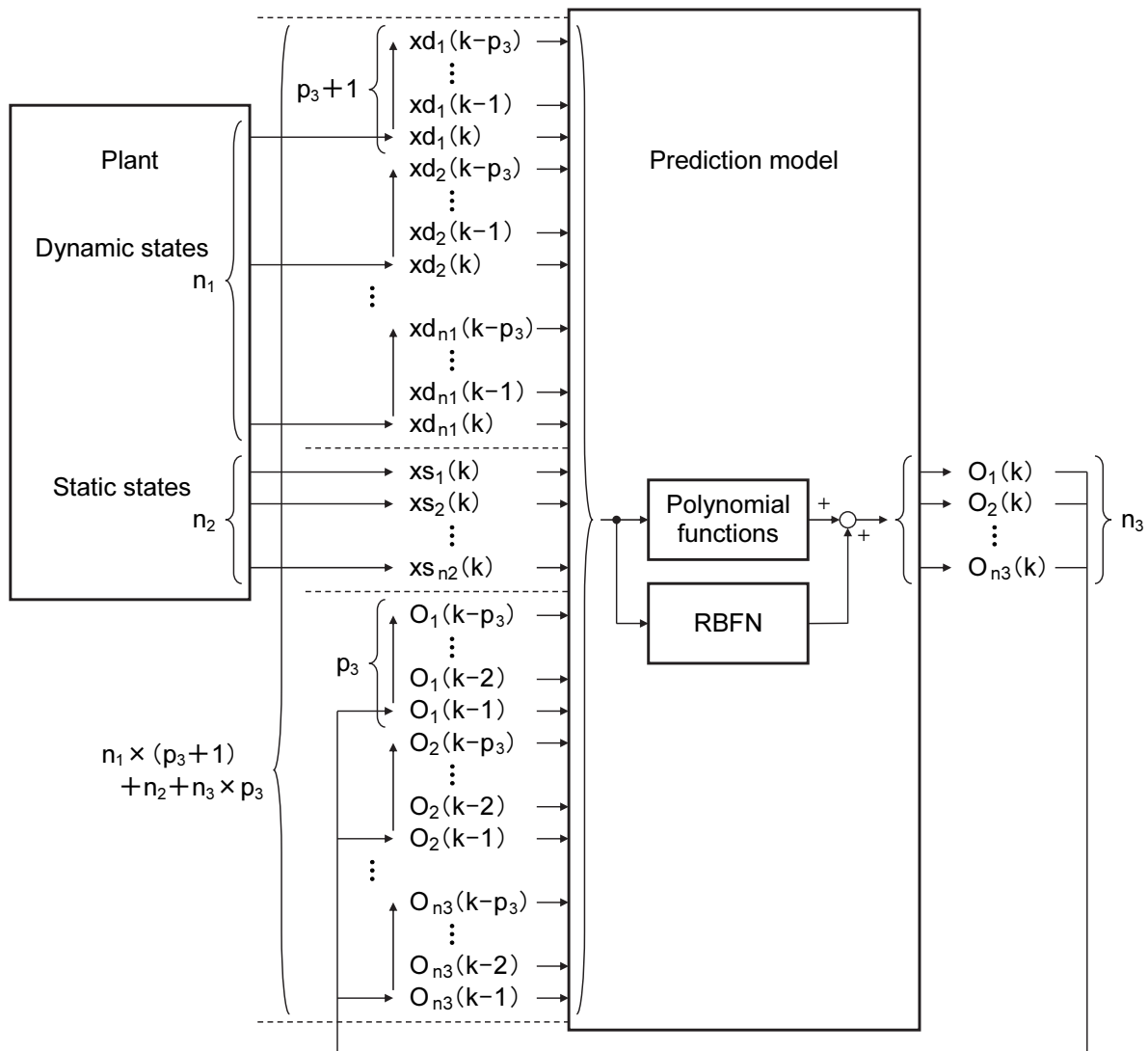


Fig. 5.3 Proposed RRBFN prediction model. The input values of dynamic states xd and static states xs , respectively, are input from the plant, and the predictive output values O are fed back to the input. The predictive output values are calculated using Eq. (5.16).

output values; t , the present time; xd , the dynamic state values; and xs , the static state values. Hence, the total number of input values is $n = n_1 \times (p_3 + 1) + n_2 + n_3 \times p_3$.

5.3.3.2 Cluster analysis for RRBFN training data

Many RRBFN training values can be obtained using a plant supervisory control and data acquisition (SCADA) system. However, because detailed sampling of long large-scale time-series data is used, RRBFN training is extremely time-consuming. The computation time depends on the size of the training data. To stay below a fixed computing time, we thin out the stored plant operation data on the SCADA system using cluster analysis before the RRBFN is trained. The cluster analysis procedure is designed to group observations based on similarities between them. The observations are datasets of RRBFN input and output values, each containing the values of quantitative variables $m = (n_1 + n_3) \times (p_3 + 1) + n_2$. To cluster observations, it is important to establish a similarity measure that joins like items. Similarity is typically determined based on the distance between observations in the m -dimensional variable space. We use Euclidian distance to measure the distance between two datasets (items) represented by x and y :

$$d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}. \quad (5.19)$$

In this study, we applied agglomerative hierarchical cluster analysis with a centroid method (Everitt et al., 2011). Agglomerative hierarchical cluster analysis begins by placing each dataset (item) in a separate cluster and then combining clusters based on their distance from each other. The process continues until the desired number of clusters is formed. At each stage, the clusters joined are the closest pair. The centroid method defines the distance between two clusters as the distance between the centroids of the clusters, where each centroid is located at the average values of each variable over all members of the cluster. Finally, only the dataset closest to the centroid of each cluster is preserved as training data for the RRBFN. Thus, the number of training data becomes the same as the number of clusters. For example, we consider a simple problem with a two-dimensional variable space. As shown in Fig. 5.4, three clusters— G_1 , G_2 , and G_3 —are enclosed with a dashed line. The © symbols indicate the centroids of the data (black points) in the clusters. The distances between the

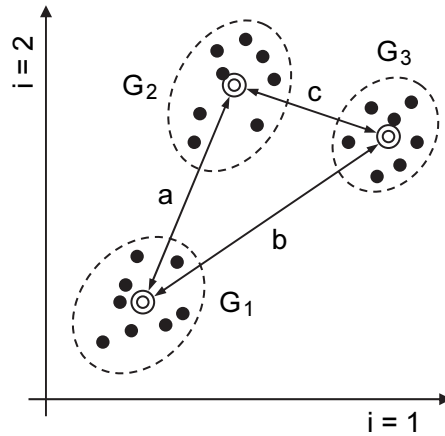


Fig. 5.4 Cluster analysis for a two-dimensional variable space.

clusters are defined as a , b , and c . Distance c between clusters G_2 and G_3 is the shortest. Therefore, clusters G_2 and G_3 are combined.

5.3.3.3 Verification of the proposed RRBFN prediction model

Storing sufficient start-up data of a steam turbine before the commercial operation of a power plant begins is difficult; hence, the proposed RRBFN prediction model is premised on performing online/offline learning after commissioning operation. Therefore, the nitrogen oxide (NO_x) decomposition process, as shown in Fig. 5.5, is considered herein as substitution operation data, to verify the accuracy of the RRBFN. These operation data are suitable for the verification of the RRBFN because the NO_x flow rate and the thermal stress of the steam turbine rotor have common characteristics that change in correlation with the power generation output. The proposed RRBFN prediction model was generated using the actual operation data from a thermal power plant. NO_x in a combustor exhaust gas flow is decomposed by ammonia (NH_3) injection as a reducer into a catalyst chamber, and NO_x is broken down into water vapor (H_2O) and nitrogen (N_2). Finally, nonreacted NO_x and NH_3 , in addition to the H_2O and N_2 , are discharged from a chimney into the atmosphere. It is generally considered difficult to predict this chemical decomposition process with high speed and accuracy in a physical model. This reaction is nonlinear and has a long lag time. Here, the dynamic state variables are the power generation output, combustor outlet NO_x flow rate, and NH_3 injection flow rate, so the number of dynamic state variables is $n_1 = 3$. The NO_x decomposition efficiency depends on the catalyst temperature, but the time

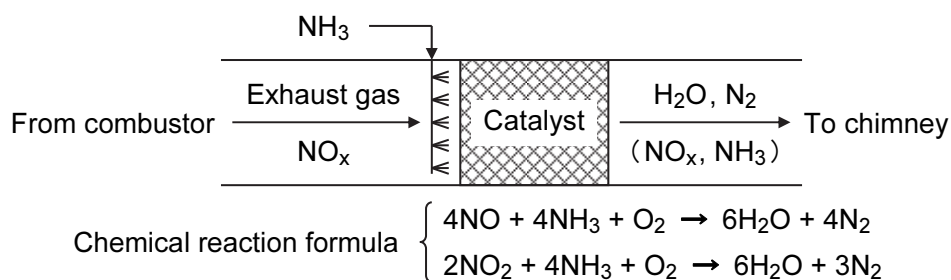


Fig. 5.5 Configuration of NO_x decomposition in a thermal power plant.

constant of the catalyst temperature is much longer than the other state variables. Hence, the static state variable is the catalyst temperature, so the number of static state variables is $n_2 = 1$. Finally, the output variable is the chimney inlet NO_x flow rate, so the number of output variables is $n_3 = 1$.

We prepared approximately 90 hours of actual plant operation data (32 464 datasets) of various operational statuses, such as start-ups, shutdowns, load changes, and load rejections. One sampling period Δt is 10 s, and the number of past sample periods p_3 is 10 (100 s). Thus, the datasets have 45 dimensions: $m = (3 + 1) \times (10 + 1) + 1 = 45$. Then, the number of datasets was reduced to 1/2 (16 232 datasets), 1/4 (8116 datasets), 1/8 (4058 datasets), and 1/16 (2029 datasets) using cluster analysis. We considered the number of prediction periods p_2 to be 10 (100 s) because the delay times (e.g., transport delay, lag time, and dead time) of this process are approximately 90 s. The RRBFN is trained using datasets of each clustering case. The RRBFN then performs a prediction calculation using test datasets of similar operation data. Figure 5.6 compares the predicted output value of 10 samples (100 s) ahead and the actual plant operation data corresponding to a load change between 100 and 50 per cent. Figures 5.6a, 5.6b, 5.6c, and 5.6d illustrate the 1/2, 1/4, 1/8, and 1/16 clustering cases, respectively (made dimensionless by normalization to the value at the base-load operation).

The computing times for RRBFN training in each clustering case using a personal computer (IntelTM CoreTM 2 Duo processor T7500 @ 2.20 GHz) are summarized in Table 5.1. The analysis results show that the prediction accuracy decreases with the number of clusters, but the computing time is considerably reduced. Of course, depending on the quality and quantity of the plant operation data, the predicted output values agree well with actual plant operation data when the number of clusters is more than 1/8 (4058 datasets) in this study.

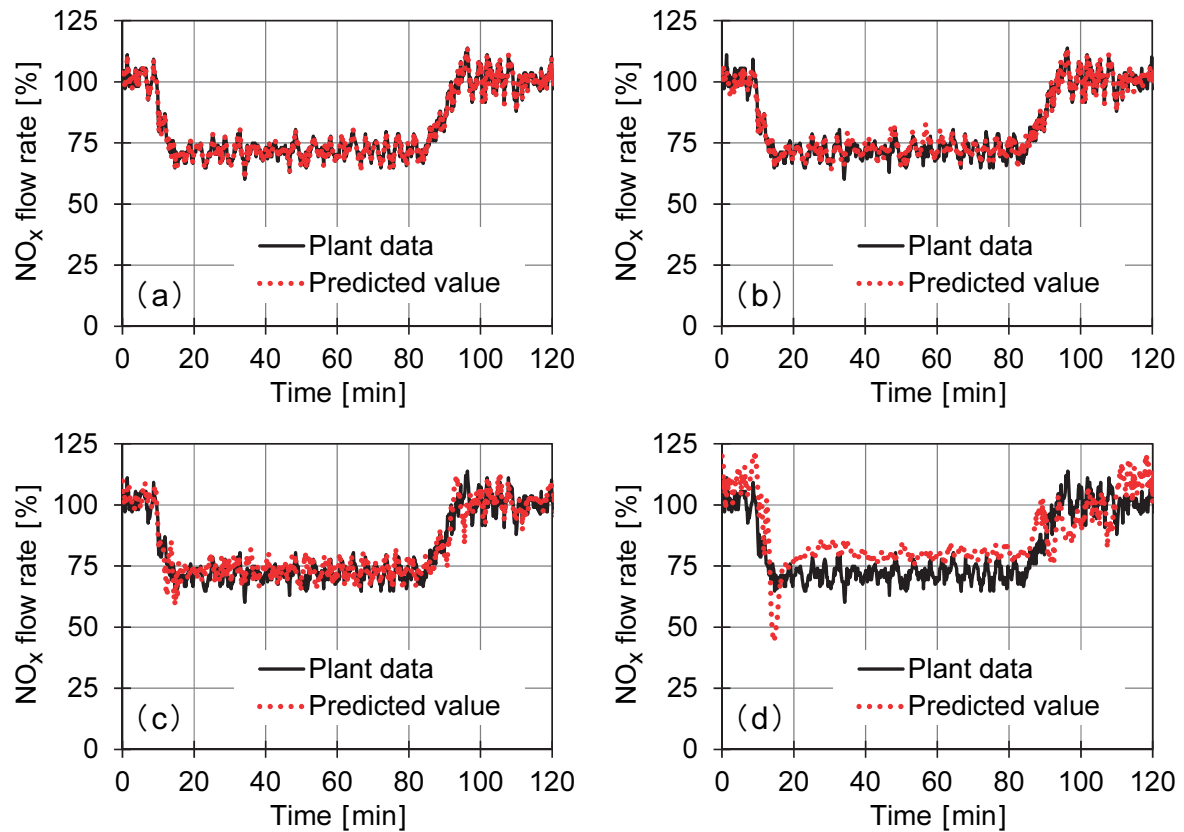


Fig. 5.6 Comparison of predicted and actual plant operation values of the chimney inlet NO_x flow rate. (a), (b), (c), and (d) Clustering cases of 1/2, 1/4, 1/8, and 1/16, respectively, made dimensionless by normalization to the value at the base-load operation.

Table 5.1 Computing times for training an RRBFN in each clustering case

Clustering case	Number of datasets	Computing time
(a) 1/2	16 232	120.0 [min]
(b) 1/4	8 116	3.0 [min]
(c) 1/8	4 058	1.0 [min]
(d) 1/16	2 029	0.5 [min]

5.4 Application results

To demonstrate the effectiveness of the proposed method, we applied it to an example of start-up control of a steam power plant. Generally, optimizing the start-up

of a steam turbine unit involves several conflicting objective functions, such as start-up time and the thermal stress of the steam turbine rotor. A lower rotor thermal stress further extends the service lifetime of the steam turbine. As expected, reducing the start-up time improves the load-following capability and fossil fuel economy. There exist many operational constraints for steam turbines. However, most constraints (e.g., rotor critical speed and vibration, and steam pressure and temperature) are safety-controlled to within given limits at any time. Thus, the operational constraints include only the thermal stress of the steam turbine rotor and the thermal expansion difference between a turbine rotor and labyrinth in the axial direction of the steam turbine rotor. Thermal stress is one of the objective functions, and it has an upper limit to prevent metal creep and fatigue. Thermal expansion increases significantly as the start-up time decreases. However, it has an upper limit determined by the speed-up and loading rates. Therefore, we considered the thermal expansion constraint indirectly by restricting the upper limit of the speed-up and loading rates in this study. In this section, we describe the configuration of the plant system, define the proposed multi-objective model predictive control problem, and present the results of a dynamic simulation study.

5.4.1 Plant system configuration

A typical steam power plant, such as that shown in Fig. 5.7, was considered in this study. The boiler generates steam for the steam turbine, and the steam turbine drives the generator. This steam power plant generates an output of 600 MW. A steam governing valve (GV) manipulates the steam flow rate swallowed by the steam turbine. The three sensors in Fig. 5.7 measure the boiler outlet steam pressure P_s , boiler outlet steam temperature T_s , and first-stage shell metal temperature of the steam turbine T_m , respectively. Generally, T_m is regarded as equivalent to the rotor surface metal temperature of the steam turbine. The steam turbine exhaust is conducted to the boiler reheater and recovered to the condenser via a low-pressure steam turbine.

During start-up, there is a large temperature difference between the hot steam and cold steam turbine rotor, and the rotor surface is heated by the higher-temperature steam. The temperature in the outer part rises higher than that of the inner part because the rotor has a large capacitance. Thermal stress on the steam turbine rotor occurs because of this temperature difference. It is difficult to measure the actual rotor thermal stress directly because the turbine rotor rotates at high speed through high-temperature steam.

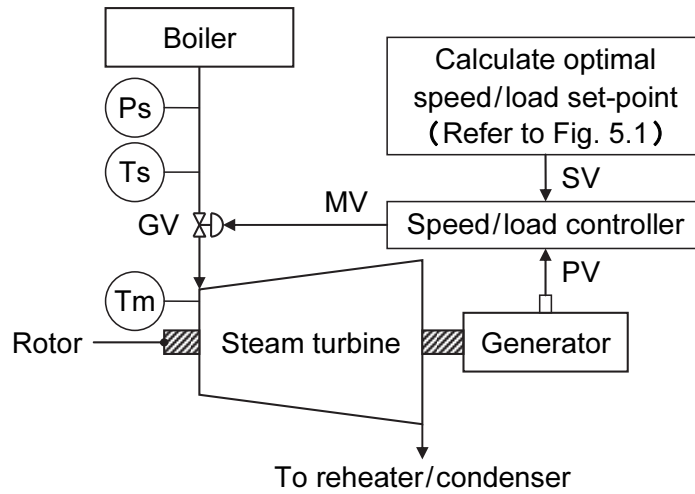


Fig. 5.7 Configuration of a typical steam power plant system. The speed and load set-points are derived from the proposed plant control system in Fig. 5.1.

Therefore, the thermal stress of the steam turbine rotor is obtained by the temperature distribution, which is calculated by a dynamic model of thermal conduction that divides the rotor into several vertical cylinders. Rotor thermal stress is described in Eqs. (5.20) and (5.21); it can be calculated by estimating the temperature distribution inside the rotor from Eq. (5.20), using the time-series behavior of the measured T_m as the boundary condition. Then, the values of the inner bore and outer surface of the rotor thermal stress (where thermal stress is concentrated) are calculated using Eq. (5.21).

$$\frac{\partial \theta}{\partial \tau} = \frac{\lambda_m}{C_m \rho_m} \left(\frac{\partial^2 \theta}{\partial r^2} + \frac{1}{r} \frac{\partial \theta}{\partial r} \right), \quad (5.20)$$

$$\sigma_{zz} = \frac{E_m \alpha_{\ell m}}{1 - \mu_m} \left(\frac{2}{R_o^2 - R_i^2} \int_{R_i}^{R_o} \theta r \, dr - \theta \right), \quad (5.21)$$

where C_m represents the specific heat; E_m , Young's modulus; r , the radial axis; R_i and R_o , the inner and outer radii, respectively; $\alpha_{\ell m}$, the coefficient of linear expansion; θ , the rotor temperature; λ_m , the thermal conductivity; μ_m , Poisson's ratio; ρ_m , the density; σ_{zz} , the thermal stress; and τ , the time.

The steam turbine speed/load controller modulates the manipulative variable (MV)

for the GV to adjust the process variable (PV) of the actual generator speed/load to its set variable (SV) from the multi-objective model predictive control system shown in Fig. 5.1.

5.4.2 Problem definitions for the proposed multi-objective model predictive control

As the parameter times of model predictive control in Fig. 5.2, we set the sampling period (optimization cycle) Δt to 1 min, prediction period p_1 to 20 min (20 samples), control period p_2 to 4 min (4 samples), and past sample period p_3 to 5 min (5 samples). These parameters were determined by adjusting each of them to a value that would enable completion of the optimization calculations within one sampling period Δt while confirming the validity of the control response. In this section, we discuss the formulation of this optimizer and describe the input and output variables of the RRBFN prediction model.

5.4.2.1 Optimizer

5.4.2.1.1 Design variables

The speed-up and loading rates of steam turbine Δu during control period p_2 are treated as design variables. Since the steam turbine generator automatically synchronizes with the electrical grid to raise the load after reaching the rated speed, the number of design variables is the same as the number of samples in control period p_2 . The speed and load of steam turbine u can be obtained by integrating as follows:

$$\Delta \mathbf{u} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+p_2-1) \end{bmatrix}, \quad (5.22)$$

$$\mathbf{u} = \begin{bmatrix} u(t+1) = u(t) + \Delta u(t) \\ u(t+2) = u(t+1) + \Delta u(t+1) \\ \vdots \\ u(t+p_2) = u(t+p_2-1) + \Delta u(t+p_2-1) \end{bmatrix},$$

where t represents the present time. The speed/load of the steam turbine at the present time $u(t)$ is given by the actual observed value $u(t) = u_o$.

5.4.2.1.2 Objective functions

The start-up time and rotor thermal stress of the steam turbine are treated as objective functions. The start-up time f_1 begins at the rolling-off of the steam turbine, ends at the realization of plant base-load operation, and corresponds to terminal time T in the mathematical model of Eq. (5.1). If the steam turbine load does not reach the base-load operation during prediction period p_1 , the steam turbine load is extrapolated to estimate f_1 . Otherwise, f_1 is the time taken to reach the base-load operation, which is the terminal time T . Therefore, f_1 during p_1 is minimized as follows:

$$\min f_1 = \begin{cases} \langle \text{before loading} \rangle \\ t + \left[p_1 + \frac{u_{S, \max} - u(t + p_1)}{\Delta u_S(t + p_1)} + \frac{u_{L, \max}}{\Delta u_{L, \text{tbl}}} \right] \Delta t, & \text{if } u(t + p_1) < u_{S, \max}, \\ \langle \text{during loading} \rangle \\ t + \left[p_1 + \frac{u_{L, \max} - u(t + p_1)}{\Delta u_L(t + p_1)} \right] \Delta t, & \text{if } u_{S, \max} = u(t + p_1) < u_{L, \max}, \\ T, & \text{otherwise,} \end{cases} \quad (5.23)$$

where $u_{S, \max}$ represents the steam turbine speed at the rated speed; $\Delta u_S(t + p_1)$, the predicted value of the steam turbine speed-up rate at the last time in prediction period $t + p_1$; $u_{L, \max}$, the steam turbine load in base-load (target load) operation; and $\Delta u_L(t + p_1)$, the predicted value of the steam turbine loading rate at the last time in prediction period $t + p_1$. For convenience, the loading rate $\Delta u_{L, \text{tbl}}$ is assumed by linearly interpolating from the speed-up rate $\Delta u_S(t + p_1)$ using Table 5.A in the Appendices.

The thermal stress of steam turbine rotor f_2 is the maximum value during start-up. It is used to calculate the values of the rotor bore and rotor surface at which the thermal stress is concentrated and to set their maxima. Therefore, f_2 during p_1 can be minimized as follow:

$$\min f_2 = \max \left[\sigma_{\max}, \hat{\sigma}_b(k), \hat{\sigma}_s(k) \right], \quad k = t + 1, t + 2, \dots, t + p_1, \quad (5.24)$$

where $\hat{\sigma}_b$ and $\hat{\sigma}_s$ represent the predicted values of the rotor bore and rotor surface thermal stresses, respectively, of the steam turbine during p_1 . The maximum thermal

stress of the steam turbine rotor during the past time σ_{\max} is given by actual observed values as follows:

$$\sigma_{\max} = \max[\sigma_b(k), \sigma_s(k)], \quad k = 0, 1, \dots, t, \quad (5.25)$$

where σ_b and σ_s represent observed values of the rotor bore and rotor surface thermal stresses, respectively, and are calculated using Eqs. (5.20) and (5.21) based on actual observed first-stage shell metal temperature T_m .

5.4.2.1.3 Constraining conditions

Due to the thermal expansion discussed at the beginning of this section, limiting values are added to the speed-up and loading rates of the steam turbine, which are the design variables in Eq. (5.22). Since the upper and lower bounds of the design variables can be limited by the ranges of the genetic operations described in Section 5.3.2, they need not be treated as penalty functions.

Before synchronizing the steam turbine generator to the electrical grid, the speed-up rate Δu is determined using Eq. (5.26). Here, between turbine speeds of 30% and 90%, the lower limit is set to avoid critical speeds.

$$\text{Speed-up rate} \begin{cases} 0 \text{ \%/min} \leq \Delta u \leq 10 \text{ \%/min}, & \text{if } 0\% \leq u < 30\%, \\ 3.333 \text{ \%/min} \leq \Delta u \leq 10 \text{ \%/min}, & \text{if } 30\% \leq u \leq 90\%, \\ 0 \text{ \%/min} \leq \Delta u \leq 10 \text{ \%/min}, & \text{if } 90\% < u \leq 100\%. \end{cases} \quad (5.26)$$

The speed-up rate of the steam turbine Δu from time $t + p_2$ to $t + p_1 - 1$ is fixed at the last value in the control period $\Delta u(t + p_2 - 1)$. The steam turbine speed u is obtained by integrating as follows:

$$\begin{cases} \Delta u(k) = \Delta u(t + p_2 - 1), \\ u(k + 1) = u(k) + \Delta u(k), \quad k = t + p_2, t + p_2 + 1, \dots, t + p_1 - 1. \end{cases} \quad (5.27)$$

The predicted steam turbine speed-up rate at the last time in the prediction period $\Delta u_S(t + p_1)$ in Eq. (5.23) is given by

$$\Delta u_S(t + p_1) = \max[\Delta u_{S, \min}, \Delta u(t + p_1 - 1)]. \quad (5.28)$$

For convenience, we set lower limit $\Delta u_{S, \min}$ to 3.333 %/min.

When the steam turbine generator is synchronized to the electrical grid, the loading rate Δu is given by Eq. (5.29). Here, between turbine loads of 0% and 5%, the lower limit is set to ensure power stabilization at the initial load.

$$\text{Loading rate} \begin{cases} 0.5 \text{ \%/min} \leq \Delta u \leq 10 \text{ \%/min} , & \text{if } 0\% \leq u \leq 5\% , \\ 0 \text{ \%/min} \leq \Delta u \leq 10 \text{ \%/min} , & \text{if } 5\% < u \leq 100\% . \end{cases} \quad (5.29)$$

The loading rate of the steam turbine Δu from time $t + p_2$ to $t + p_1 - 1$ is fixed at the last value in the control period $\Delta u(t + p_2 - 1)$. The steam turbine load u is obtained by integrating as follows:

$$\begin{cases} \Delta u(k) = \Delta u(t + p_2 - 1) , \\ u(k + 1) = u(k) + \Delta u(k) , \quad k = t + p_2, t + p_2 + 1, \dots, t + p_1 - 1 . \end{cases} \quad (5.30)$$

The predicted steam turbine loading rate at the last time in the prediction period $\Delta u_L(t + p_1)$ in Eq. (5.23) is given by

$$\Delta u_L(t + p_1) = \max[\Delta u_{L, \min}, \Delta u(t + p_1 - 1)] . \quad (5.31)$$

For convenience, we set lower limit $\Delta u_{L, \min}$ to 0.5 %/min. However, when u is higher than 50%, the boiler outlet steam temperature T_s is statically determinate (refer to Table 5.B in the Appendices) and the thermal stress of the steam turbine rotor is reduced, so $\Delta u_{L, \min}$ is switched to 10 %/min.

The thermal stress of steam turbine rotor f_2 has the following upper limit:

$$g = f_2 - \sigma_U \leq 0 , \quad (5.32)$$

where σ_U is the upper limit of the thermal stress of the steam turbine rotor. The constraining condition of Eq. (5.32) is considered in the penalty function of Eq. (5.11).

5.4.2.2 Prediction model

Since the mechanical structure of the steam turbine is complex, it is difficult to predict the thermal stress of the steam turbine rotor accurately with a short computing time using only a physical model. Therefore, we will predict the behavior of the first-stage shell metal temperature of the steam turbine T_m in the future using the

RRBFN proposed in Section 5.3.3.1. The thermal stress of the steam turbine rotor, which is one of the objective functions, is predicted by solving the physical model of Eqs. (5.20) and (5.21) with T_m as a boundary condition. That is, by using this gray-box model that combines the physical (white-box) and RRBFN (black-box) models, the computing time and accuracy can be balanced. The analytical method for implementing Eqs. (5.20) and (5.21) on the controller is well known from conventional research (Matsumoto et al., 1982; Nakai et al., 1996), so its description is omitted here. T_m is strongly correlated with boiler outlet steam pressure P_s , boiler outlet steam temperature T_s , and steam turbine speed and load u . The RRBFN input and output variables during prediction period p_1 are summarized below:

$$\begin{aligned} & \langle \text{Input variables} \rangle \\ & \left\{ \begin{array}{l} P_s(k-p_3), \dots, P_s(k-1), P_s(k), \\ T_s(k-p_3), \dots, T_s(k-1), T_s(k), \\ u(k-p_3), \dots, u(k-1), u(k), \\ T_m(k-p_3), \dots, T_m(k-2), T_m(k-1), \quad k = t+1, t+2, \dots, t+p_1, \end{array} \right. \end{aligned} \quad (5.33)$$

$$\begin{aligned} & \langle \text{Output variable} \rangle \\ & T_m(k), \quad k = t+1, t+2, \dots, t+p_1. \end{aligned} \quad (5.34)$$

Here, T_m , which is one of the input variables, is the feedback value of the predicted output variable. The steam turbine speed and load u in future times $[t+1, t+p_1]$ are given by the optimizer described above. For convenience, the boiler outlet steam pressure P_s and temperature T_s in $[t+1, t+p_1]$ are assumed by linearly interpolating from the steam turbine speed and load u using Table 5.B in the Appendices. Since P_s and T_s are controlled sufficiently faster than T_m responds; even if they are given by a static function, few problems occur. However, the optimal controls and prediction methods of P_s and T_s are left for future research. Input variables (P_s , T_s , u , and T_m) from past times $[t-p_3+1, t]$ are determined based on actual observed values.

5.4.3 Results of the dynamic simulation study

Prior to application to a real power plant, we verified the proposed multi-objective model predictive control method by performing dynamic simulation studies of a 600 MW steam power plant. The dynamic simulation models were verified using actual plant data, as described in Chapters 2–3. The simulation was executed under warm

start-up conditions in which the initial metal temperature of the steam turbine rotors was 170°C.

The dynamic simulation was executed for each aspiration level with the following procedure:

1. Initial training of the RRBFN prediction model was performed using the simulation results from two cases—the fastest and slowest start-up operations of the plant (corresponding to points A and B, respectively, in Fig. 5.8a). In these cases, the upper limit of the rotor thermal stress can be neglected.
2. The plant start-up simulation was executed three times during online learning of the RRBFN prediction model.

Figures 5.9, 5.10, and 5.11 show the plant start-up simulation results. The steam turbine speed (broken line) and load (solid line) in Figs. 5.9a, 5.10a, and 5.11a are depicted as dimensionless using the rated speed and base-load operation values, respectively. The optimization calculations were executed after rub-check holding at a speed of 13.333%. The rotor bore (broken line) and rotor surface (solid line) thermal stresses of the steam turbine in Figs. 5.9b, 5.10b, and 5.11b are shown as dimensionless by normalization to the upper limits. Figure 5.8 shows the result of the multi-objective

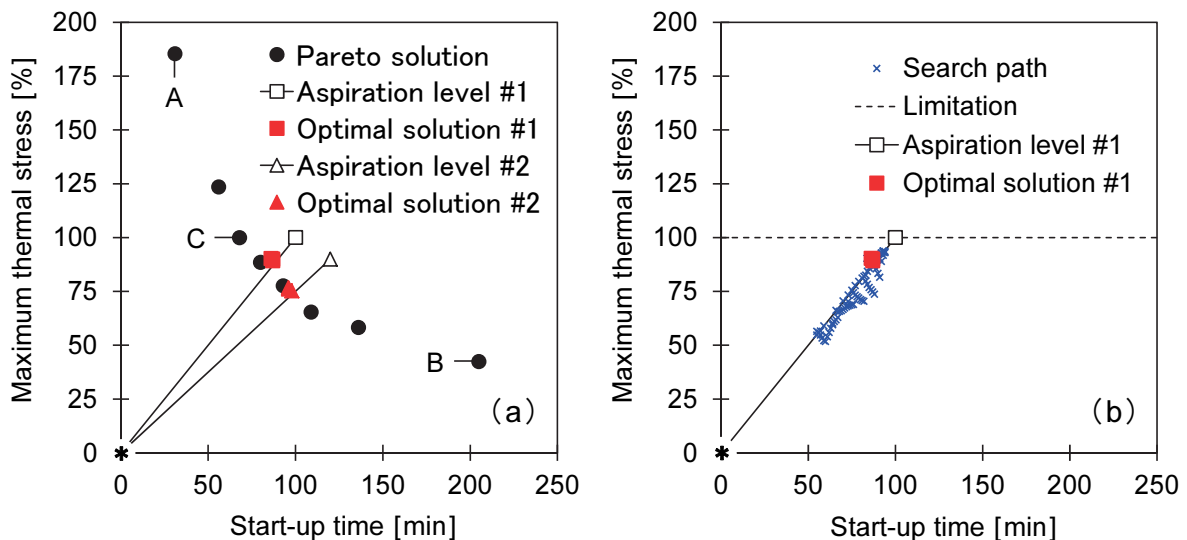


Fig. 5.8 Results of multi-objective optimization process. (a) Aspiration levels and comparison of the obtained optimal solutions to the Pareto-optimal solutions. (b) Search path history for the first optimal solution.

optimization process as a relation between start-up time (f_1) and the maximum value of the normalized rotor thermal stresses during start-up (f_2). The ideal point (the symbol * in Fig. 5.8) is set to $(f_1^*, f_2^*) = (0 \text{ min}, 0\%)$. The black circle points in Fig. 5.8a are the Pareto-optimal solutions obtained by changing the upper limit of the rotor thermal stress. That is, these solutions were obtained by partially modifying the proposed method to time-optimal model predictive control based on a single objective.

First, the results of time-optimal model predictive control with rotor thermal stress as a constraining condition are shown in Fig. 5.9. The qualitative trend of these results is very similar to that obtained by applying the conventional steam turbine start-up of model predictive controls (e.g., Nakai et al., 1996). Specifically, the start-up time required to reach the base-load operation is shortened by increasing the steam turbine speed and load while keeping the rotor thermal stresses at 100% of the upper limit. This result corresponds to point C in Fig. 5.8a. That is, $(f_1, f_2) = (68 \text{ min}, 100\%)$.

Next, we discuss the results of the proposed multi-objective model predictive control. Suppose that the user wants to decrease f_2 (thermal stress). Since above-mentioned solution (point C in Fig. 5.8a) is already a Pareto-optimal solution, it is impossible to improve both criteria. Therefore, suppose that the user agrees to relax f_1 (start-up time) and let the first aspiration level be $(\overline{f_{1,1}}, \overline{f_{2,1}}) = (100 \text{ min}, 100\%)$ (the symbol \square in Fig. 5.8a). Then, the first optimal solution is $(f_{1,1}, f_{2,1}) = (86 \text{ min}, 90\%)$ (the symbol \blacksquare in Fig. 5.8a); here, both criteria are improved from the first aspiration level. Moreover, suppose that the user agrees to relax f_1 further and let

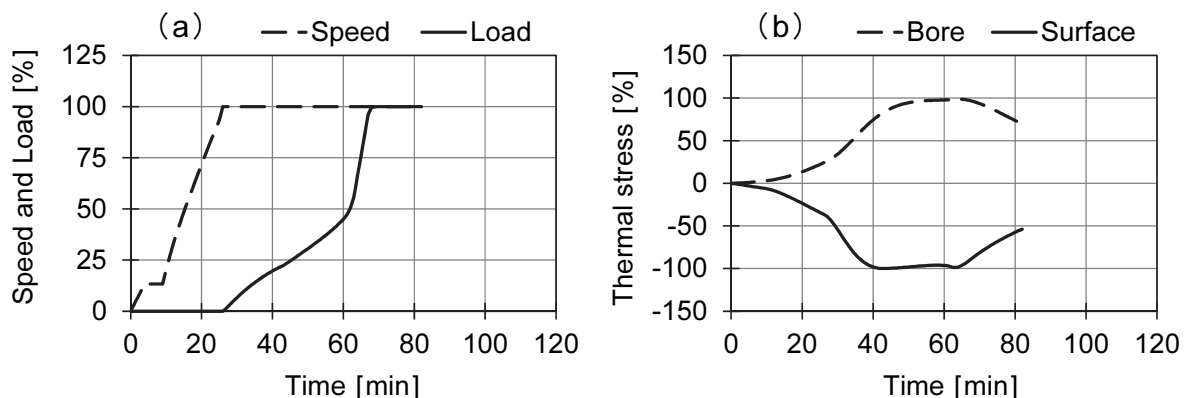


Fig. 5.9 Dynamic simulation results obtained for time-optimal model predictive control corresponding to point C in Fig. 5.8a. (a) Steam turbine speed and load. (b) Rotor bore and rotor surface thermal stresses.

the second aspiration level be $(\overline{f_{1,2}}, \overline{f_{2,2}}) = (120 \text{ min}, 90\%)$ (the symbol \blacktriangle in Fig. 5.8a). Then, the second optimal solution is $(f_{1,2}, f_{2,2}) = (96 \text{ min}, 77\%)$ (the symbol \blacktriangle in Fig. 5.8a); here, both criteria are improved from the second aspiration level. However, the second optimal solution shows that f_2 improves, but f_1 deteriorates compared to the first optimal solution, because there is a trade-off between f_1 and f_2 . The first optimal solutions in Fig. 5.8a all plot the results of three plant start-up simulations, but they are almost the same. The deviations in these solutions are only within 0.667 min for f_1 and 0.619% for f_2 . Likewise, the second optimal solutions in Fig. 5.8a all plot the results of three plant start-up simulations, but they are almost the same. The deviations in these solutions are only within 1.333 min for f_1 and 0.774% for f_2 . Since the optimal solutions are reproducible, the learning of the RRBFN prediction model is satisfactory. Although some gap results from the influence of the prediction error of the start-up time in Eq. (5.23), all these solutions obtaining multi-objective control substantially agree with the Pareto-optimal solutions obtaining single-objective control closest to each aspiration level. This agreement demonstrates the validity of the optimal solutions obtained using the proposed method. The control responses of the first optimal solution in Fig. 5.10 are the results of the third start-up simulation. This multi-objective control satisfies the operating constraints in all periods; the rotor thermal stresses in Fig. 5.10b are particularly reduced from the time-optimal control results in Fig. 5.9b. Note that a peak rotor thermal stress occurs around 80 min. Figure 5.8b plots the search path history for this optimal control input at

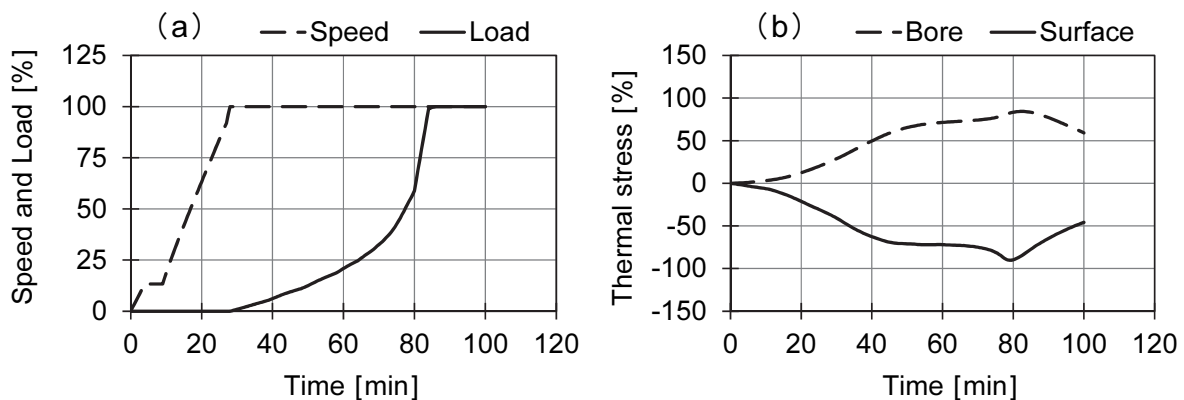


Fig. 5.10 Dynamic simulation results obtained for the first aspiration level. (a) Steam turbine speed and load with the proposed multi-objective control. (b) Rotor bore and rotor surface thermal stresses.

each sampling time. In the time-optimal control, the rotor thermal stress is limited to a constant upper limit along the broken line in Fig. 5.8b. In contrast, in the proposed multi-objective control, the objective functions are optimized along the solid line in Fig. 5.8b (the line connecting the aspiration level and the ideal point), so that the rotor thermal stress fluctuates near the Pareto-optimal solution. However, this peak can be made smaller by adjusting the prediction period p_1 and control period p_2 . The control responses of the second optimal solution in Fig. 5.11 are the results of the third start-up simulation. The control responses of the second optimal solution show that the start-up time is longer than it is for the first optimal solution and that the rotor thermal stresses are reduced. In this way, the proposed method has features that enable flexible online control based on multi-objective evaluation according to aspiration levels.

5.4.4 Computing time

We intend to implement the proposed multi-objective model predictive control method in an actual plant controller in future research. Here, we discuss the computing time of the proposed method, which is important for practical use. The GA parameters greatly affect the computing time. In Section 5.3.2, we used 20 individuals and set the maximum number of iterations to 50 generations. We confirmed that the solutions converged before 50 generations occurred. We executed start-up simulations using a personal computer (IntelTM CoreTM i7-3770S CPU @ 3.10 GHz). In this case, it took approximately 33 s (searching using the GA for approximately 9 s, learning and

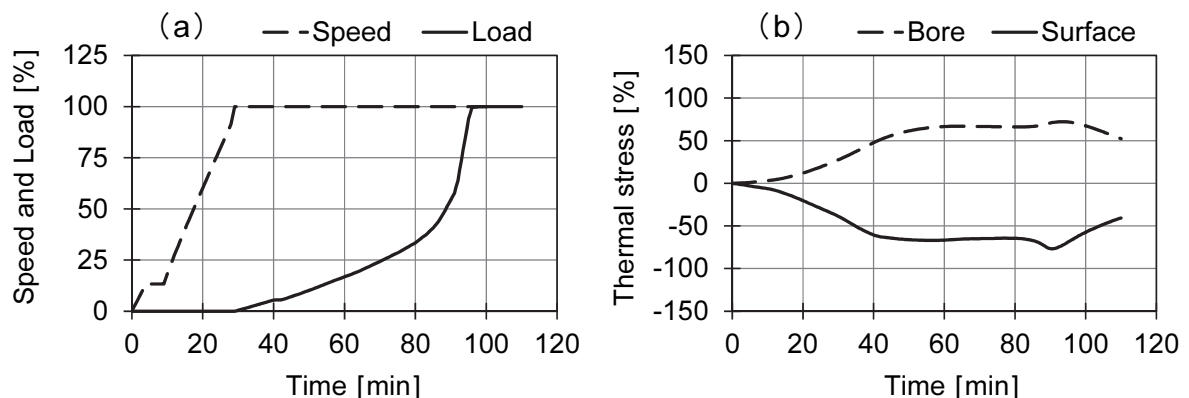


Fig. 5.11 Dynamic simulation results obtained for the second aspiration level. (a) Steam turbine speed and load with the proposed multi-objective control. (b) Rotor bore and rotor surface thermal stresses.

predicting the metal temperature for the RRBFN for approximately 6 s, and calculating the rotor thermal stress based on the physical model for approximately 18 s) to obtain the optimal control input per sampling period Δt . Any case can be processed within the 60 s sampling period. This time is sufficiently short for practical control problems such as steam turbine start-up.

5.5 Conclusions

We proposed herein a novel intelligent multi-objective model predictive control method, in which an RRBFN was applied for model prediction, and the satisficing trade-off method was used for multi-objective optimization. This control method can be used to

1. optimize several conflicting control objectives in online control based on multi-objective evaluation through human–computer interactions, and
2. achieve an optimal and flexible plant control within a restrained practical computing time for real-time applications and with acceptable control quality using online adaptive model prediction.

In addition, the proposed method maintained the framework of the previously studied human–computer interactions (see Chapter 4). Therefore, after optimizing the control schedule offline using an optimization system discussed in Chapter 4, the proposed method can be used for online optimal control.

The proposed method worked well throughout our experiments. The main results are as follows:

1. The accuracy of the proposed RRBFN prediction model was verified using the actual operation data of a NO_x decomposition process in a thermal power plant.
2. The RRBFN training data size can be reduced using a cluster analysis, which was important for maintaining predictive accuracy and shortening the learning time of the RRBFN prediction model.
3. The proposed multi-objective model predictive control method was applied to steam turbine start-up control. Although some gap resulted from the influence of the prediction error of the start-up time, the solutions obtained through dynamic simulations substantially agreed with the Pareto-optimal solutions, which were given by single objectives with constraints. However, the multi-objective control responses partially differed from the single-objective results with constraints in its

search path history.

4. Our application results indicated that the computing time of the proposed method was practical for implementation in a controller in the future research.

This study demonstrated the verification results of the proposed control method through dynamic simulations. The future work will field-test the proposed control method on a real power plant and add other operational constraints (e.g., NO_x emission rate from the plant described in Section 5.3.3.3, etc.)

Appendices

Table 5.A Functional table of the speed-up rate $\Delta u_S(t + p_1)$ and loading rate $\Delta u_{L, tbl}$ in Eq. (5.23)

Speed-up rate	[%/min]	0	3.333	5	10
Loading rate	[%/min]	0	0.5	3	6

Table 5.B Functional table of the steam turbine speed and load u , and the boiler outlet steam pressure P_s and temperature T_s

		Speed		Load				
u	[%]	0	100	0	25	50	99.75	100
P_s	[MPa]	8.46	8.46	8.46	8.46	9.34	16.66	16.66
T_s	[°C]	350	410	410	474	538	538	538

References

- Bemporad, A. and Muñoz de la Peña, D., Multiobjective model predictive control, *Automatica, A Journal of IFAC*, Vol. 45, Issue 12, (2009), pp. 2823–2830.
- Bien, Z., Kang, D.-O. and Yang, S.-H., Programming approach for fuzzy model-based multiobjective control systems, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 7, No. 4, (1999), pp. 293–300.
- Biyik, E., Long, C. E., D’Amato, F. J., Di Palma, S. and Baker, D. A., Enhanced operational flexibility of combined cycle through advanced controls, *Proceedings*

- of the 55th ISA Power Industry Division Symposium 2012 (POWID2012), Vol. 492, (2012), pp. 204–213, Austin, TX, the United States.
- Cressie, N., Spatial prediction and ordinary Kriging, *Mathematical Geology*, Vol. 20, Issue 4, (1988), pp. 405–421.
- D'Amato, F. J., Industrial application of a model predictive control solution for power plant startups, *Proceedings of the IEEE International Conference on Control Applications 2006 (CACSD-CCA-ISIC2006)*, (2006), pp. 243–248, Munich, Germany.
- Eshelman, L. J. and Schaffer, J. D., Real-coded genetic algorithms and interval-schemata, *Foundations of Genetic Algorithms*, Vol. 2, (1993), pp. 187–202, Morgan Kaufmann Publishers, Inc., San Mateo, CA, the United States.
- Everitt, B. S., Landau, S., Leese, M. and Stahl, D., *Cluster Analysis*, 5th Edition, (2011), John Wiley & Sons, Ltd., Chichester, the United Kingdom.
- He, D., Wang, L. and Sun, J., On stability of multiobjective NMPC with objective prioritization, *Automatica, A Journal of IFAC*, Vol. 57, (2015), pp. 189–198.
- Maciejowski, J. M., *Predictive Control with Constraints*, (2002), Prentice-Hall, Pearson Education, Ltd., Harlow, the United Kingdom.
- Matsumoto, H., Sato, Y., Kato, F., Eki, Y., Hisano, K. and Fukushima, K., Turbine control system based on prediction of rotor thermal stress, *IEEE Transactions on Power Apparatus and Systems*, Vol. 101, No. 8, (1982), pp. 2504–2512.
- Matsumoto, S., Yakushi, K. and Kitaguchi, N., Optimal turbine startup methodology based on thermal stress prediction, *Thermal and Nuclear Power Engineering Society, The Thermal and Nuclear Power*, Vol. 61, No. 9, (2010), pp. 798–803 (in Japanese).
- Miyazawa, K., Fukamachi, Y., Yakushi, K. and Asanaka, T., Development and experience of the commissioning of optimal startup methodology for steam turbine, *Thermal and Nuclear Power Engineering Society, The Thermal and Nuclear Power Generation Convention Collected Works*, Vol. 11, (2015), pp. 22–27 (in Japanese).
- Montgomery, D. C., *Design and Analysis of Experiments*, 9th Edition, (2017), John Wiley & Sons, Inc., New York, NY, the United States.
- Myers, R. H., Montgomery, D. C. and Anderson-Cook, C. M., *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, 4th Edition, (2016), John Wiley & Sons, Inc., New York, NY, the United States.

- Nakai, A., Nakamoto, M., Kakehi, A. and Hayashi, S., Turbine start-up algorithm based on prediction of rotor thermal stress, *IFAC Proceedings Volumes, The 13th World Congress of IFAC*, Vol. 29, Issue 1, (1996), pp. 6867–6872, San Francisco, CA, the United States.
- Nakayama, H., Aspiration level approach to interactive multi-objective programming and its applications, In: *Advances in Multicriteria Analysis* (Pardalos, P. M., Siskos, Y. and Zopounidis, C., eds.), (1995), pp. 147–174, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Nakayama, H., Arakawa, M. and Sasaki, R., Simulation-based optimization using computational intelligence, *Optimization and Engineering*, Vol. 3, Issue 2, (2002), pp. 201–214, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Nakayama, H., Yun, Y. and Yoon, M., Sequential approximate optimization, In: *Sequential Approximate Multiobjective Optimization Using Computational Intelligence* (Yoon, M., Yun, Y. and Nakayama, H., eds.), (2009), pp. 113–149, Springer-Verlag, Berlin, Heidelberg, Germany.
- Núñez, A., Cortés, C. E., Sáez, D., Schutter, B. D. and Gendreau, M., Multiobjective model predictive control for dynamic pickup and delivery problems, *Control Engineering Practice, A Journal of IFAC*, Vol. 32, (2014), pp. 73–86.
- Papadrakakis, M., Lagaros, N. D. and Tsompanakis, Y., Structural optimization using evolution strategies and neural networks, *Computer Methods in Applied Mechanics and Engineering*, Vol. 156, Issues 1–4, (1998), pp. 309–333.
- Rojas, C. A., Rodriguez, J., Villarroel, F., Espinoza, J. R., Silva, C. A. and Trincado, M., Predictive torque and flux control without weighting factors, *IEEE Transactions on Industrial Electronics*, Vol. 60, No. 2, (2013), pp. 281–690.
- Schuermann, N., Idelchi, A. and Yang, S., Model predictive control for increased steam turbine operational flexibility, *Proceedings of the 59th ISA Power Industry Division Symposium 2016 (POWID2016)*, (2016), pp. 16–28, Charlotte, NC, the United States.
- Tica, A., Gueguen, H., Dumur, D., Faille, D. and Davelaar, F., Hierarchical nonlinear model predictive control for combined cycle start-up optimization, *Proceedings of the 51st IEEE Conference on Decision and Control 2012 (CDC2012)*, (2012), pp. 2593–2598, Maui, HI, the United States.
- Wang, G. G. and Shan, S., Review of metamodeling techniques in support of engineering design optimization, *Journal of Mechanical Design, Transactions of*

the ASME, Vol. 129, No. 4, (2007), pp. 370–380.

Yun, Y. and Nakayama, H., Ensembled support vector machines for meta-modeling, In: Knowledge and Systems Sciences (Chen, J., Nakamori, Y., Yue, W. and Tang, X., eds.), Communications in Computer and Information Science, Vol. 660, (2016), pp. 203–212, Springer Nature Singapore Private, Ltd., Singapore.

Zavala, V. M. and Flores-Tlacuahuac, A., Stability of multiobjective predictive control: A utopia-tracking approach, Automatica, A Journal of IFAC, Vol. 48, Issue 10, (2012), pp. 2627–2632.

Chapter 6

Multi-objective evaluation method for combined cycle power plant start-up

Abstract

With the increase in the available types of renewable energy sources, such as wind and solar radiation, it is essential that thermal power plants operate more flexibly in order to stabilize their power system. For practical flexible operation, operators (or designers) need to consider multiple evaluation indicators. This study proposes a multi-objective evaluation method for plant operation using self-organizing maps and data envelopment analysis. Pareto-optimal solutions extracted from the results of start-up dynamic simulation of a combined cycle power plant verify the efficacy of the proposed method.

Keywords

Data envelopment analysis, Design of experiments, Dynamic models, Evaluation method, Multi-objective optimization, Power generation, Self-organizing maps, Simulation, Steam turbine, Tradeoff analytics

6.1 Introduction

The recent liberalization of the electricity markets, along with the rapid proliferation of renewable energy sources such as wind and solar radiation, which tends to influence power generation depending on weather conditions and seasons, is necessitating improvement of the flexibility of thermal power generation systems to balance power grids. State-of-the-art gas-turbine combined-cycle power plants are undergoing various technological developments designed to enable flexible operation that entails frequent start-ups and shutdowns, rapid start-ups and load changes, and provides frequency duty control (Greis et al., 2012; Kaplan et al., 2015; Smith et al., 2015). In this scenario, more flexible operation is significantly affected by thermal and

mechanical fatigue stresses on machines and emission of air pollutants. Thus, operators and manufacturers need to more intensively consider reducing these harmful effects than in a conventional preplanned inflexible operation.

In order to realize flexible operation, various multi-objective optimization methods that consider multiple conflicting evaluation functions, such as start-up time, fuel consumption rate, lifetime consumption rate, and pollutant emission rate of the machine, have been proposed (Gallestey et al., 2002; Bertini et al., 2012; Yoshida et al., 2017). In addition, we have developed an offline optimization system for plant operation using an interactive multi-objective optimization method, as described in Chapter 4. We have also proposed a multi-objective model predictive control method that enables flexible online operation, as described in Chapter 5. However, in a multi-objective optimization problem, there is no single solution that simultaneously optimizes all the evaluation functions. Instead, there are multiple Pareto-optimal solutions that each constitutes a trade-off among the evaluation functions. Therefore, in order to adopt the multi-objective optimization method for plant operation for practical use, an analysis method to support multiple-criteria decision-making for extracting the Pareto-optimal solutions desired by operators (or designers) is important.

Figure 6.1 shows the concept underlying the multiple-criteria decision-making system based on tradeoff analytics for plant operation considered in this study. Although several multi-objective optimization methods for plant operation have been

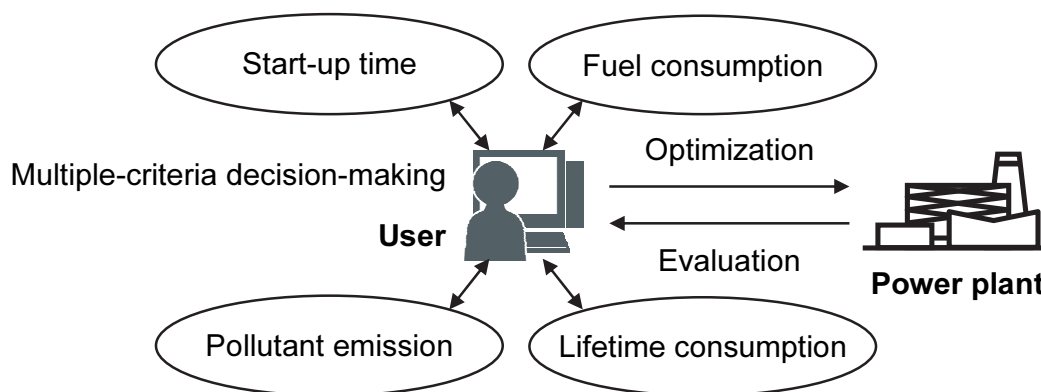


Fig. 6.1 Concept underlying the multiple-criteria decision-making system based on tradeoff analytics for plant operation. For practical flexible operation, the users (operators or designers) need to consider multiple evaluation indicators.

proposed (as described above), only a few studies have been conducted on multi-objective evaluation methods for plant operation. In this study, we propose a multi-objective evaluation method for plant operation using self-organizing maps (SOM) (Kohonen, 1982, 2001) and data envelopment analysis (DEA) (Charnes et al., 1978). In addition, the Pareto-optimal solutions obtained from our previous study (see Chapter 4, Section 4.6.3) are verified using the results of start-up dynamic simulation of a combined cycle power plant.

6.2 Combined cycle power plant start-up

This study considers a multi-shaft type combined cycle power plant consisting of three gas turbine units, three heat recovery steam generator (HRSG) units, and one steam turbine unit. The schematic diagram of this plant is illustrated in Fig. 4.1 (see Chapter 4, Section 4.2.1). The gas turbines and the steam turbine drive the generators. Further, the HRSGs generate steam for the steam turbine using waste heat from the gas turbines. This plant generates a total output of 670 MW.

The start-up curve for this plant is illustrated in Fig. 4.2 (see Chapter 4, Section 4.2.2). The steam turbine start-up schedule has a significant effect on the start-up characteristics of the entire plant. This is due to the thermal stress that develops in the steam turbine rotors, which is a factor that is particularly decisive for reducing the start-up time, fuel consumption rate, and NO_x emission rate from the plant. Therefore, the following four schedule variables are selected: ① steam turbine acceleration rate, x_1 , ② low-speed heat soak time, x_2 , ③ high-speed heat soak time, x_3 , and ④ initial-load heat soak time, x_4 . These schedule variables restrain the developed thermal stress by gently warming the steam turbine rotors. In addition, the following four evaluation functions are considered: ① start-up time, f_1 , which is from the start of the first gas turbine to the plant base-load operation; ② fuel consumption rate, f_2 , which is the gross weight during start-up; ③ thermal stress of the steam turbine rotors, f_3 , which is the maximum value during start-up; and ④ NO_x emission rate from the plant, f_4 , which is the maximum value of moving average per hour during start-up. The multi-objective optimization problem is to find the vector of the schedule variables, $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$, in which the multiple evaluation functions (f_1 , f_2 , f_3 , and f_4) are minimized under the operational constraints, with f_3 and f_4 having upper limits g_{U3} and g_{U4} , respectively. This problem can be summarized as follows:

$$\begin{cases} \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), f_4(\mathbf{x})]^T, \\ \text{subject to } \begin{cases} f_3(\mathbf{x}) - g_{U3} \leq 0, \\ f_4(\mathbf{x}) - g_{U4} \leq 0, \end{cases} \end{cases} \quad (6.1)$$

$$\text{where } \begin{cases} x_1 \in (120 \text{ rpm/min}, 180 \text{ rpm/min}, 360 \text{ rpm/min}), \\ 5 \text{ min} \leq x_2 \leq 66 \text{ min}, \\ 0 \text{ min} \leq x_3, x_4 \leq 66 \text{ min}. \end{cases} \quad (6.2)$$

6.3 Multi-objective evaluation methods

In this section, we explain the two multi-objective evaluation methods employed—namely, SOM and DEA. SOM is used to visualize and evaluate the correlation between the schedule variables and multiple evaluation functions described in Section 6.2. DEA is used to conduct comprehensive evaluation with the multiple evaluation functions, extract diverse Pareto-optimal solutions, and provide a quantitative grasp of the improvement values.

6.3.1 SOM

SOM, proposed by Kohonen (1982), models the nervous system of the cerebral cortex. It is a type of hierarchical neural network comprising two layers, an input layer and a competitive layer (an output layer), as shown in Fig. 6.2. The first layer is an n -dimensional input vector $\mathbf{x}(t)$. The second layer is called a competitive layer and is generally a two-dimensional array for visualizing output values.

SOM learning is unsupervised competitive learning. The learning steps are as follows:

Step 1: Determine the number of units (map size), randomly determine the initial elements of the reference (weight) vector $\mathbf{m}_i(0)$ in all units, and set the time $t = 0$.

Step 2: Give the input vector $\mathbf{x}(t)$. Then, search for a neuron i that minimizes the Euclidean distance between $\mathbf{x}(t)$ and $\mathbf{m}_i(t)$, and set the neuron, c , as defined in Eq. (6.3):

$$\|\mathbf{x}(t) - \mathbf{m}_c(t)\| = \min_i \|\mathbf{x}(t) - \mathbf{m}_i(t)\|. \quad (6.3)$$

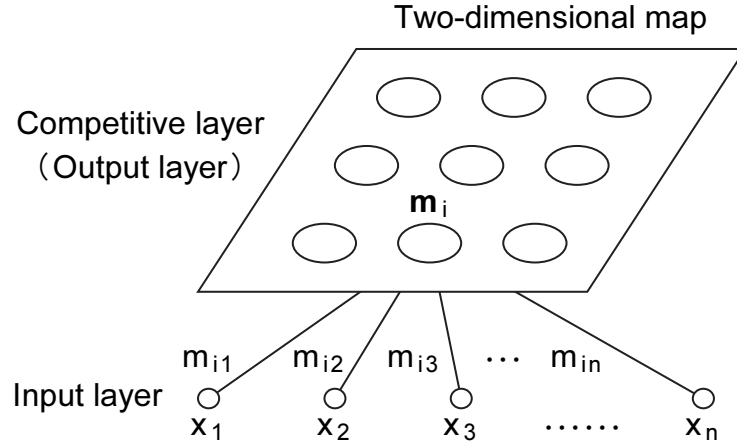


Fig. 6.2 Basic structure of a two-dimensional SOM known as a Kohonen network. This SOM has a feed-forward structure with a single computational layer arranged in rows and columns. Each neuron is fully connected to all the source nodes in the input layer.

Step 3: Let the neuron with the reference vector $\mathbf{m}_c(t)$ be the winner unit. When the winner unit is decided, the unit in the neighborhood $N_c(t)$ is updated, as defined in Eq. (6.4):

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t) [\mathbf{x}(t) - \mathbf{m}_i(t)], \quad (6.4)$$

where $h_{ci}(t)$ is a neighborhood function, defined as in Eq. (6.5) using a learning rate coefficient $\alpha(t)$:

$$h_{ci}(t) = \begin{cases} \alpha(t), & \text{if } i \in N_c(t), \\ 0, & \text{otherwise.} \end{cases} \quad (6.5)$$

Step 4: Repeat Steps 2–3 a maximum of $t \geq T$ times ($T = 200$ in this study) to learn the input vector, and perform the update $t \leftarrow t + 1$. Concurrently, $\alpha(t)$ and $N_c(t)$ are reduced according to the progress of learning, as defined in Eq. (6.6):

$$\begin{cases} \alpha(t) = \alpha(0) \left(1 - \frac{t}{T}\right), \\ N_c(t) = N_c(0) \left(1 - \frac{t}{T}\right), \end{cases} \quad (6.6)$$

where $\alpha(0)$ and $N_c(0)$ are the initial values of $\alpha(t)$ and $N_c(t)$, respectively.

Step 5: Repeat Steps 2–4 for all input vectors, which results in similar units to each input vector being gathered and placed on the map of the competitive layer. Thus, correlation of multidimensional data can be visualized in two dimensions by using SOM.

6.3.2 DEA

DEA is a business analysis method proposed by Charnes et al. (1978). It can be employed to evaluate the efficiency of multiple subjects of analysis in relation to one another. According to DEA, the greater the efficiency value θ —which is the ratio of the total input (total investment) I_{total} to the total output (total production) O_{total} —the more efficient is the production.

$$\theta = \frac{O_{\text{total}}}{I_{\text{total}}}. \quad (6.7)$$

To use this formula, the total input is set as “1 (one),” and the output is replaced with each of the evaluation functions. If the user would like to see the relative efficiency value of given data, the data are set to c and then, because there are multiple evaluation functions, the weight u_i^c ($i = 1, 2, \dots, r$) is put on number i 's evaluation functions f_i^c and multiplied and summed to represent the total output. At this point, the user would want to maximize the efficiency value. However, to determine the efficiency in comparison to other data, the CCR (Charnes, Cooper, and Rhodes) model (Charnes et al., 1978) was formulated as a linear programming problem, where the efficiency value θ^c is maximized under conditions in which the efficiency values of all data are less than one. In DEA, the CCR model, BCC (Banker, Charnes, and Cooper) model (Banker et al., 1984), and FDH (free disposal hull) model (Tulkens, 1993) are known as representative models incorporating preference structure of decision-makers. These DEA models differ in production possibility set when considering dual problems. Among them, the CCR model is the simplest DEA model. To obtain the relative advantage over other data, the most convenient weight can be sought. In the original CCR model, because the efficiency value for data c , which is to be known, is included in the restrictions, the maximum value of the efficiency value is limited to one. Therefore, if data c are removed from the restrictions, and the expression is reformulated, Eq. (6.8) will be obtained.

$$\left\{ \begin{array}{l} \max_{u^c} \theta^c = \sum_{i=1}^r u_i^c f_i^c, \\ \text{subject to} \left\{ \begin{array}{l} \sum_{i=1}^r u_i^c f_i^j \leq 1, \quad j = 1, 2, \dots, s, \quad j \neq c, \\ u_i^c \geq 0, \quad i = 1, 2, \dots, r. \end{array} \right. \end{array} \right. \quad (6.8)$$

Because in this formulation, data c are removed from the constraining conditions in the original CCR model, it is called the super CCR model. If the results are better than other data, the efficiency value can be expressed by values larger than one. That is, as the efficiency value of the Pareto-optimal solution is greater than one, it becomes easier to find the distinctive results among the data being compared.

Further, in this case, the weight u_i^c is the same as the indirectly obtained weight w_i when the multi-objective optimization problem is scalarized using the linear weighted sum, as shown in Eq. (6.9):

$$F(\mathbf{x}) = \sum_{i=1}^r w_i f_i(\mathbf{x}). \quad (6.9)$$

In the case of solutions that are not found on the Pareto frontier, the values will be identical in both the original and super CCR models. On the other hand, if the Pareto-optimal solutions are found on the Pareto frontier, for the original CCR model, all efficiency values will become one without any distinction. Thus, the user will not be able to conclude readily about the evaluation functions and the weight they should be assigned to realize the most advantageous position. However, with the super CCR model, the efficiency value will be more than one, and thus, the user will be able to readily conclude which functions should be assigned weight in order to obtain the maximum efficiency in comparison to other solutions, despite the distortions in the Pareto frontier. As a result, the user can find the most appropriate weight in every solution. Multi-objective optimization seeks the optimal solution in relation to the aspiration level of the user, which is expressed as a numerical target. In contrast, with DEA, the user finds the weight of the numerical target based on the obtained optimal solution.

For example, consider the simplified dual-objective problem depicted in Fig. 6.3.

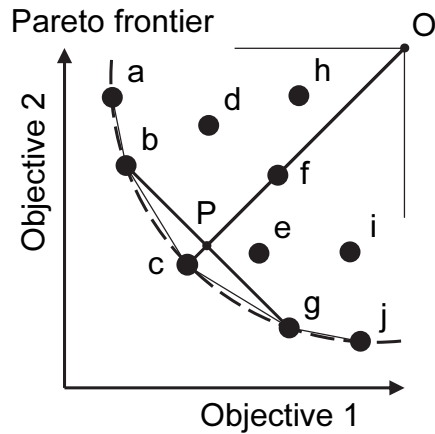


Fig. 6.3 DEA as a dual-objective problem. Here, the dashed curve represents the Pareto frontier. An intersection P , where the first line segment Oc crosses the second line segment bg , is determined.

Datasets a , b , c , g , and j are all on the envelope and, in all cases, they are Pareto-optimal solutions with efficiency values greater than one. For example, the efficiency value of dataset c is $(\overline{Oc} / \overline{OP}) \geq 1$. Moreover, the efficiency value of datasets d , e , f , h , and i are less than one, and each value represents a feasible solution. In addition, the relative position compared to the Pareto-optimal solution can be determined from the efficiency value.

6.4 Simulation evaluation

In this section, we present an evaluation of the solutions obtained from the results of the start-up dynamic simulation. The dynamic simulation models were verified using actual plant data, as described in Chapters 2–3. Simulations were conducted for a warm start-up condition, i.e., the initial temperature of the steam turbine rotors was 180°C.

6.4.1 Analysis of start-up solutions by design of experiments

To prepare datasets of the schedule variables and evaluation functions, dynamic simulation was executed with 400 start-up schedules using Latin hypercube sampling (LHS) in addition to the three Pareto-optimal solutions obtained from the previous study (see Chapter 4, Section 4.6.3). LHS is a method of sampling random numbers that attempts to distribute samples evenly over the sample space for the schedule variables.

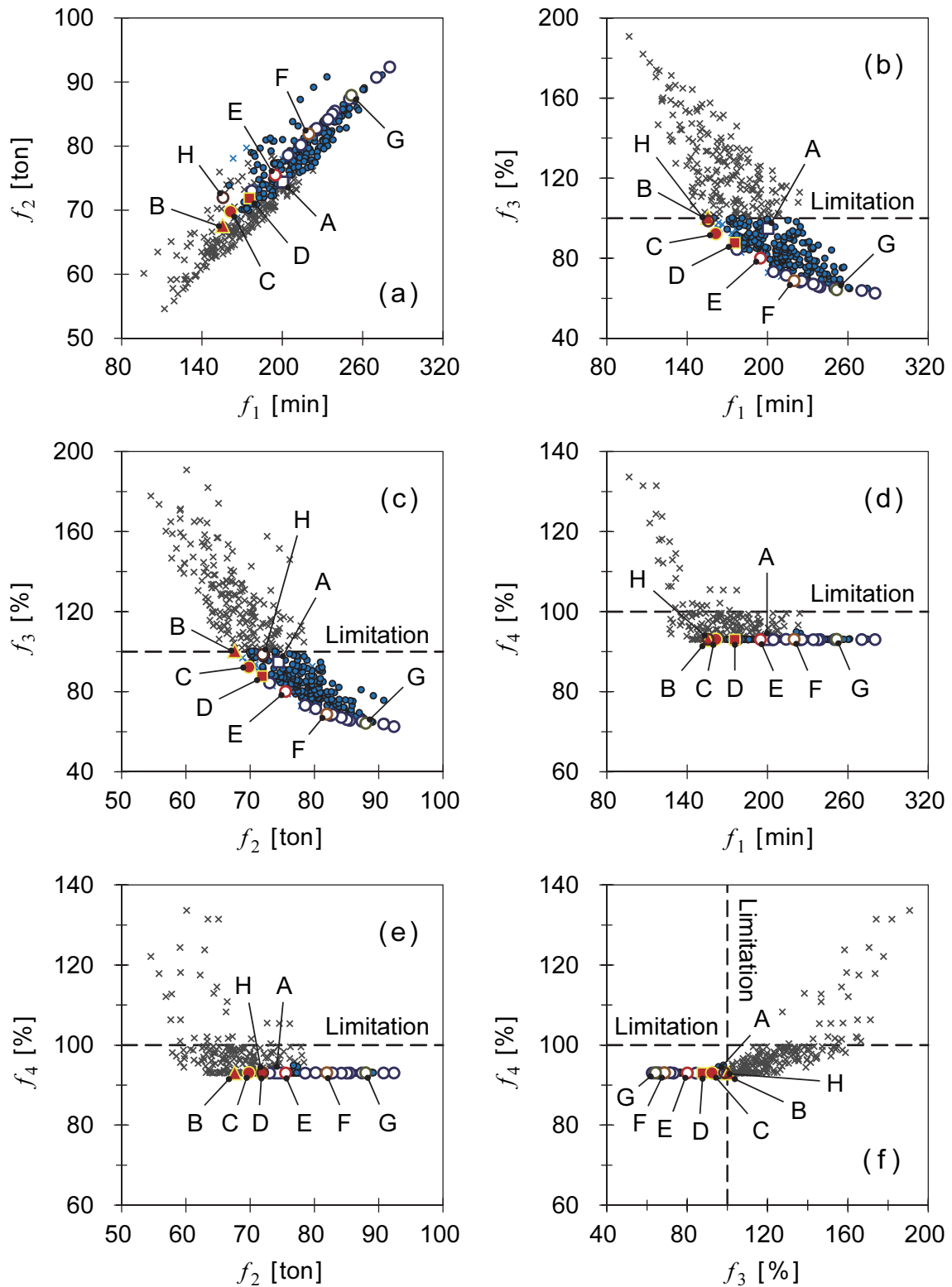


Fig. 6.4 Correlation diagram between each pair of evaluation functions in two dimensions. Here, solutions A–H show the listed representative start-up solutions in Table 6.1.

The correlation diagram between each pair of evaluation functions is shown in Fig. 6.4. In the figure, the values of f_3 and f_4 are made dimensionless by using their respective upper limit values. In Fig. 6.4, the points indicated by crosses (the symbol \times) are infeasible start-up solutions because operational constraints were not satisfied; the other points indicate feasible start-up solutions. Among the results, of all $403 = 400 + 3$ start-up dynamic simulations, there were 195 feasible start-up solutions and 208 infeasible start-up solutions. The linear relation is visually evident between f_1 and f_2 in Fig. 6.4a. On the other hand, the trade-off relations are visually evident between f_1 and f_3 in Fig. 6.4b, and between f_2 and f_3 in Fig. 6.4c. According to Figs. 6.4d and 6.4e, f_1 and f_2 decrease when the start-up is fast, and there is a negative correlation (trade-off relation) when f_4 sharply increases. However, referring to Fig. 6.4f, when f_4 exceeds its own upper limit value, f_3 also exceeds its own upper limit value. However, the values of f_4 are kept substantially constant for the feasible start-up solutions because the NO_x emission rate can be suppressed by the selective catalytic reduction control, depending on the catalyst temperature.

Let us now focus on the three Pareto-optimal solutions obtained from the previous study (see Chapter 4, Section 4.6.3). In Fig. 6.4, solution D (the symbol \blacksquare) is the first optimal solution, solution C (the symbol \bullet) is the second, and solution B (the symbol \blacktriangle) is the third. Referring to Figs. 6.4b and 6.4c, as these solutions are at the edge of the data group of feasible start-up solutions, it is clear that the optimal solutions on the Pareto frontier are accurately determined.

6.4.2 Evaluation of start-up solutions by SOM

The SOM input vector consists of all 403 start-up solutions for eight elements, comprising four schedule variables (x_1 , x_2 , x_3 , and x_4) and four evaluation functions (f_1 , f_2 , f_3 , and f_4). The map size is $15 \times 15 = 225$. In this case, the computation time for SOM learning, using a personal computer (IntelTM CoreTM i5-450M CPU @ 2.40 GHz), was approximately 6 s. In Fig. 6.5, the contour lines of each element of the input vector are visualized as the learning results of SOM. The darker colors represent relatively smaller values. As the contour diagrams of Fig. 6.5e (f_1) and Fig. 6.5f (f_2) are very similar, they have a linear relation. On the other hand, Fig. 6.5e (f_1) and Fig. 6.5g (f_3) are in a trade-off relation because the contour diagrams are virtually reversed. Fig. 6.5f (f_2) and Fig. 6.5g (f_3) similarly have a trade-off relation. The light color (corresponding to a large value) of Fig. 6.5h (f_4) overlaps with a part of the light color

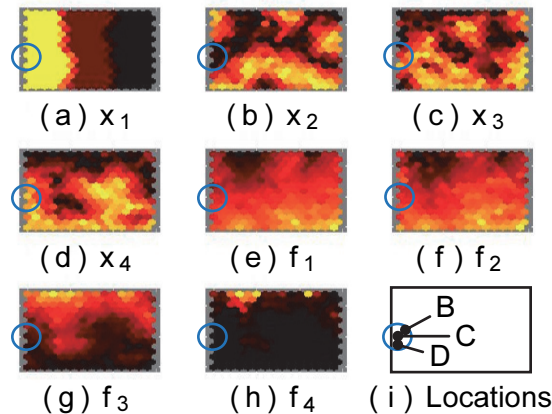


Fig. 6.5 Contour line visualizations by the SOM. If the contour line patterns of two inputs were very similar (the same or opposite color scheme), you can assume that the inputs are highly correlated.

of Fig. 6.5g (f_3). These results are in good agreement with the analysis by design of experiments described in the previous section.

Next, the relationship between the schedule variables and the evaluation functions is considered. The contour diagrams of Fig. 6.5a (x_1) and Fig. 6.5g (f_3) are unrelated as they are significantly different. However, Fig. 6.5d (x_4) and Fig. 6.5g (f_3) are negatively correlated because the contour diagrams are virtually reversed. The three Pareto-optimal solutions B, C, and D, obtained from the previous study (see Chapter 4, Section 4.6.3), are located in the left-central part (the blue circle in Fig. 6.5i). That is, referring to the blue circular parts in Figs. 6.5a–6.5h, x_1 sets the maximum value in 360 rpm/min, x_2 sets the minimum value in 5 min, x_3 is significantly adjusted according to the values of the evaluation functions, and x_4 tends to be a large value.

6.4.3 Evaluation of start-up solutions by DEA

DEA calculation was performed using the values of three evaluation functions (f_1 , f_2 , and f_3) for the 195 feasible start-up solutions described in Section 6.4.1. In this case, the computation time for the super CCR model, using a personal computer (Intel™ Core™ i7-3770S CPU @ 3.10 GHz), was approximately 2 s. The best value was temporarily converted to 10, whereas the worst values were converted to 1 for all the existing data, all of which were oriented for maximization. From the calculation results of the super CCR model, 15 start-up solutions with efficiency value greater than

0.995 are indicated in Fig. 6.4 by white circles (the symbol \bigcirc), in addition to the three Pareto-optimal solutions obtained from the previous study (see Chapter 4, Section 4.6.3). These start-up solutions can be regarded as Pareto-optimal solutions as their efficiency values are approximately greater than one ($\theta \geq 0.995 \cong 1$).

The representative calculation results are listed in Table 6.1, where the efficiency value in relation to the start-up solutions and weights of each of the evaluation functions are presented. Solution A (the symbol \square in Fig. 6.4) is a start-up solution corresponding to the first aspiration level in the previous study (see Chapter 4, Section 4.6.3). However, because its efficiency value is 0.922, which is sufficiently smaller than one, it is not a Pareto-optimal solution. The improved start-up solution from solution A is solution D. Solution D is the first optimal solution obtained from the first aspiration level because the ratio of the weight to each evaluation function is improved similarly to solution A. In the case of solution D, the efficiency value is slightly less than one. This is likely because, whereas the super CCR model assumes convexity of the Pareto frontier, the solution is found in the non-convex section. Because solutions B and C have a slightly greater efficiency value than one, these solutions can be seen as valid Pareto-optimal solutions.

We also performed evaluations based on weight. The characteristics of solution D indicate that it has less f_3 and no other characteristics, but in reality, f_2 should also be considered. This is because, within the data being compared, there is a better solution to f_3 . With solution C, it is clear that f_1 and f_3 should be balanced with little regard for f_2 . Whereas solution B is at the operational limit of this plant, it is clear that all that needs to be done is to minimize f_2 . This is likely because minimization of f_2 is a sufficient condition and not a necessary condition to minimize f_1 . Solution E has a weight on decreasing f_3 over solution D, which is a compromised start-up solution to make f_2 larger. Solution F is a start-up solution with weights suppressing f_3 and compromising f_1 . Solution G is a start-up solution that increases this tendency. Solution H is a weakly Pareto-optimal solution in which only f_1 surpasses other solutions slightly. Thus, by evaluating the optimal solutions obtained using DEA, the objects (evaluation indicators) that should be assigned priority can be indirectly evaluated to achieve the aspiration level of the user.

Solutions B and D, demonstrated in Figs. 6.6 and 6.7, are the sample results of a start-up dynamic simulation, respectively. In Figs. 6.6 and 6.7, GT denotes the gas turbines; ST, the steam turbine; Stress, the normalized thermal stress of the steam

Table 6.1 Representative calculation results of DEA

Solution	A	B	C	D	E	F	G	H
Symbol	□	▲	●	■	○	○	○	○
Efficiency value	0.922	1.082	1.017	0.997	0.996	1.003	1.004	1.005
	[pu]							
f_1	0	0	0.429	0	0	0.134	0.026	0.657
f_2	0.305	1.082	0.050	0.340	0.251	0	0	0
f_3	0.617	0	0.538	0.657	0.745	0.870	0.978	0.348
x_1	180	360	360	360	360	180	120	360
x_2	40.0	5.0	5.0	5.0	6.7	18.2	39.4	9.7
x_3	40.0	26.7	22.4	41.9	59.3	46.0	46.0	6.1
x_4	30.0	43.5	53.6	48.3	48.4	65.4	66.0	52.7
	[r/min ²]							
	[min]							
	[min]							
	[min]							
f_1	200	156	161	175	195	220	252	156
f_2	74.4	67.6	69.8	71.9	75.5	81.9	87.9	72.0
f_3	94.9	100	92.3	87.7	80.1	68.7	64.2	98.6
	[ton]							
	[%]							

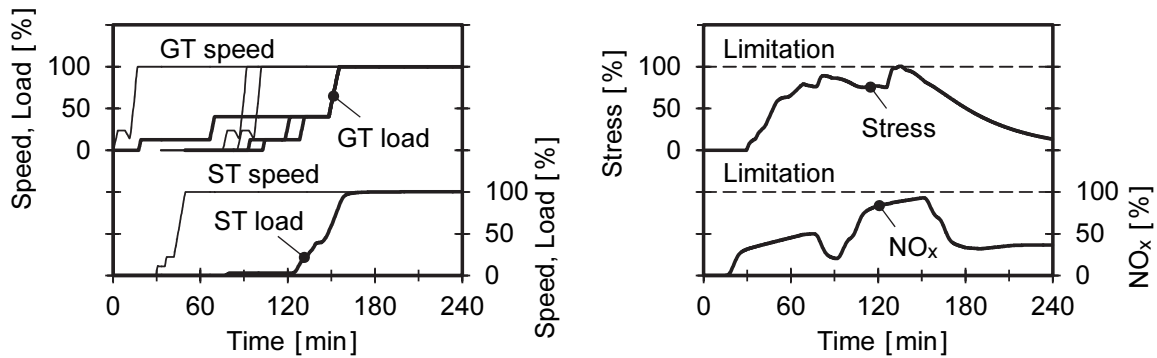


Fig. 6.6 Results of the start-up dynamic simulation for solution B (the symbol ▲ in Fig. 6.4). This solution is the third optimal solution obtained in Chapter 4, Section 4.6.3.

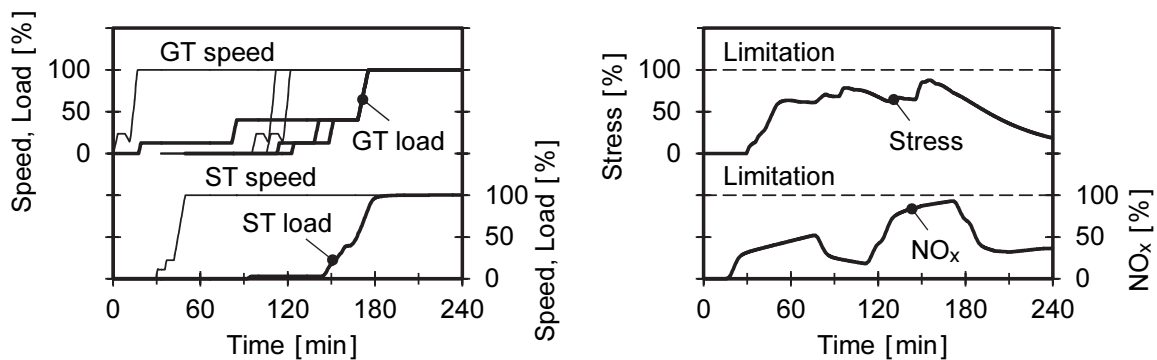


Fig. 6.7 Results of the start-up dynamic simulation for solution D (the symbol ■ in Fig. 6.4). This solution is the first optimal solution obtained in Chapter 4, Section 4.6.3.

turbine rotors; and NO_x , the normalized NO_x emission rate from the plant. In the figures, the dashed lines represent the respective limitation values.

6.5 Conclusions

We proposed an analysis method that uses multi-objective evaluation methods based on SOM and DEA to evaluate flexible operation for a thermal power plant. The results of this study are summarized as follows:

1. We introduced an analysis method in which the correlation between the schedule variables and multiple evaluation functions is projected onto a two-dimensional space using SOM, and the information on the solution space is provided based on its similarities.
2. Based on DEA, which analyzes the weight of multiple candidates obtained as potential optimal solutions, the relations among the various evaluation functions of the optimal solutions obtained through the multi-objective optimization method were clarified, allowing the user to decide the solutions that should be given priority in order to achieve their aspiration level.

References

- Banker, R. D., Charnes, A. and Cooper, W. W., Some models for estimating technical and scale inefficiencies in data envelopment analysis, *Management Science*, Vol. 30, Issue 9, (1984), pp. 1078–1092.
- Bertini, I., De Felice, M., Pannicelli, A. and Pizzuti, S., Optimizing the start-up operations of combined cycle power plants using soft computing methods, *Logic Journal of the IGPL*, Vol. 20, Issue 4, (2012), pp. 648–656.
- Charnes, A., Cooper, W. W. and Rhodes, E., Measuring the efficiency of decision making units, *European Journal of Operation Research*, Vol. 2, Issue 6, (1978), pp. 429–444.
- Gallestey, E., Stothert, A., Antoine, M. and Morton, S., Model predictive control and the optimization of power plant load while considering lifetime consumption, *IEEE Transactions on Power Systems*, Vol. 17, No. 1, (2002), pp. 186–191.
- Greis, J., Gobrecht, E. and Wendt, S., Flexible and economical operation of power plants: 25 years of expertise, *Proceedings of the ASME Turbo Expo 2012: Turbine Technical Conference and Exposition*, Vol. 6, (2012), pp. 497–504, Copenhagen, Denmark.
- Kaplan, B., Myrissidis, T., Bratowski, M., Ojira, Y., Miyasaka, T. and Abe, K., Operational flexibility upgrades on combined cycle power plants, *Proceedings of the International Gas Turbine Congress 2015 (IGTC2015)*, DVD-ROM, (2015), pp. 1416–1421, Tokyo, Japan.
- Kohonen, T., Self-organized formation of topologically correct feature maps, *Biological Cybernetics*, Vol. 43, Issue 1, (1982), pp. 59–69.

Kohonen, T., *Self-Organizing Maps*, (2001), Springer-Verlag, Berlin, Heidelberg, Germany.

Smith, G. R., Smith, R. and Karaca, E., GE rapid response combined cycle power plant, *Proceedings of the International Gas Turbine Congress 2015 (IGTC2015)*, DVD-ROM, (2015), pp. 1422–1428, Tokyo, Japan.

Tulkens, H., On FDH efficiency analysis: Some methodological issues and applications to retail banking, courts, and urban transit, *Journal of Productivity Analysis*, Vol. 4, No. 1/2, (1993), pp. 183–210.

Yoshida, Y., Yamanaka, K., Yamashita, A., Iyanaga, N. and Yoshida, T., Optimal start-up control of combined cycle power plants using the multi-objective evolutionary algorithm, *Transactions of the JSME (in Japanese)*, Vol. 83, No. 847, (2017), DOI: 10.1299/transjsme.16-00433.

Part III

Multi-objective optimization system for plant layout design

Chapter 7

Genetic algorithm approach for designing layout of plot plan for power plants

Abstract

While engineering a power plant layout, objective functions and constraints of the power plant frequently conflict each other in terms of factors such as safety, economical costs, maintainability, and construction term of works. This study proposes a multi-objective layout optimization system for a power plant through effective interaction between the designer and the computer; this system is developed by applying an interactive satisficing trade-off method for multiple-criteria decision-making. In order to obtain the layout candidates for decision-making, we present a suitable genetic algorithm (GA) approach to arrange the plot plan for the layout of equipment and building structures in a site. In this GA approach, the coding method and evolution operation method used for obtaining a diverse optimal layout solution are employed with a low computational load and simple design data. Finally, the validity and usefulness of this proposed system is demonstrated by solving a layout design problem for an actual gas-turbine combined-cycle power plant.

Keywords

Design engineering, Engineering optimization, Genetic algorithm, Interactive optimization, Layout design, Multi-objective optimization, Plant layout, Power plant, Satisficing trade-off method, Systems engineering

7.1 Introduction

In various large-scale plants such as power plants and petroleum chemical plants, it is necessary to arrange functionally the equipment, building structures, pipe conduits, electric circuits, and roads that constitute the plant system in accordance with various site-specific parameters (e.g., land use, water use, and environmental impact). This

kind of layout design requires the making of a layout plan that satisfies multiple evaluation indicators such as safety, economical costs, maintainability, and construction term of works. Further, evaluation indicators, which are difficult to formulate, such as aesthetics and harmony with the surrounding environment, are subjective and have always been thought as high-level design problems.

This study focuses on plot plans, that is, the layout of equipment constituting a plant and positions of buildings inside a site. It is obvious that a plot plan is the starting point for a subsequent detailed design based on three-dimensional CAD (computer-aided design), but for thermal power plants and the like, its role is relatively more important in the estimation of an engineering design. In other words, based on the design plans of a thermodynamic cycle, once the capacity and quantity of the equipment are decided on the basis of the plant functionality, the plot plan is examined immediately. Next, construction costs and duration are estimated based upon the decided plot plan. Therefore, although considering practicability and profitability is important when deciding on an optimal plot plan, the amount of time given to the plant manufacturer is significantly limited. For example, in the case of for thermal power plants built for exporting, the time period from customer inquiry to bidding is typically less than six months. Moreover, given the time required for tasks such as the verification of specifications requested by the customer and customer inquiry of construction and civil engineering companies, machines, and materials, sufficient time for designing cannot be secured.

Thus far, various optimization methods for plant layout design have previously been proposed. However, most of these methods target a small portion of systems in a process plant (Suzuki et al., 1991a, 1991b; Castell et al., 1998; Guirardello and Swaney, 2005) or are limited in their use in defining beforehand the degree of influence among components of a chemical plant (Penteado and Ciric, 1996; Fuchino et al., 1997; Patsiatzis et al., 2004), such as analyzing the safety risks. Further, in the case of knowledge-based design support system (Fujita et al., 1993, 1995; Mierswa, 2005), when the targeted plant changes or the scale of the plant system becomes large, the conversion of design knowledge such as layout rules into a computer database becomes a complex operation. Moreover, the fundamental problem of plant layout designs, that is, examining cases of a designer who considers multiple-criteria decision-making and many evaluation indicators are actually quite low; hence, it is doubtful whether a designer is truly satisfied with the obtained layout solutions. Based on our knowledge,

the design of a plot plan for a power plant still depends on the expertise and intuition of skilled designers.

In this study, we developed a multi-objective optimization system for plant layout design that supports a designer's multiple-criteria decision-making by an interactive operation between the designer and the computer. This system has the advantage of integrating the designer's knowledge without any prior input of design knowledge to the computer. Further, since the plot plan for the power plant is designed quickly in an upstream design phase, the design data tends to be limited. In light of these two features, in this study, we propose a layout optimization method depending on a genetic algorithm (GA) as a quasi-optimal layout solution while considering comparatively short computation times and simple design data. Further, in order to verify the usefulness of the proposed method, we discussed the application of this method to the general arrangement problem of actual gas-turbine combined-cycle power plants.

7.2 Plant layout design and its evaluation indicators

The main equipment in a thermal power plant includes boilers, turbines, and generators; the auxiliary equipment includes heat exchangers, pumps, and fans; the incidental facilities are utilized for fuel, water treatment, and power transmission and transformation; and the onsite buildings include control rooms, electrical rooms, and service buildings. In addition, many pipes, ducts, and cables, in accordance with the system configuration, connect these equipment and building structures. For example, a 1000-MW gas-turbine combined-cycle power plant has more than 50 equipment and building structures, several hundred pipes and ducts, and several thousand cables, indicating the complexity of the plant layout design.

Traditionally, when designing a plot plan, designers would make several layout plans and a final layout design would be decided on the basis of the consideration of several evaluation indicators. The following are the five major evaluation indicators:

① **Legality and safety:** The layout plan of a thermal power plant must conform to the requirements of various national and local laws and regulations on construction sites, such as plant location law, noise regulation law, and fire defense law. For example, according to the noise regulation law, the noise at the borderline of the site should be under a certain reference value. As part of the fire defense law, there are regulations about the required safety distance and open space for equipment used for handling

dangerous materials. In addition, safety and issues regarding customers' requests must be satisfied.

② **Cost consideration based on the length of pipes and ducts:** The length of pipes and ducts required for the transport of working fluids such as fuel, air, gas, water, and steam, particularly high-temperature and high-pressure pipes made from special metals that are very expensive, significantly affects construction costs. Therefore, the plant facilities are arranged efficiently according to the flow path of the working fluids so that the length of the pipes and ducts can be shortened as much as possible. Shortening the length of the pipes and ducts not only reduces the cost but also has the advantage of increasing efficiency by reducing pressure loss and thermal loss.

③ **Cost consideration based on the length of cables:** Similar to the length of pipes and ducts, the length of cables required for power supplies, controls, instrumentation, and electricity transmission from/to the electric machinery also significantly affects construction costs. By appropriately arranging the building structures including the control rooms, electrical rooms, service buildings, and power transformers, the length of the cables can be shortened as much as possible.

④ **Maintainability and expandability of the plant facilities:** Since the life cycle of a thermal power plant is approximately 30 years, it is necessary to ensure that there are sufficient pathways and workspace for operating, maintaining, inspecting, and repairing the plant. In addition, a wide space should be kept among corresponding equipment and building structures. Moreover, an expanded area should be provided considering plans to the future for the changing or addition of plant facilities.

⑤ **Ease of the construction works:** The construction costs and duration are closely related to the above-mentioned evaluation indicators ②–④. That is, if the required lengths of pipes, ducts, and cables are reduced, then the process of laying them becomes easy. If the space among equipment and building structures is wide, then the transportation of large equipment, performing of tasks involving heavy machinery and construction machines, and the installation of equipment become easy. An additional advantage is that civil constructions do not hinder tasks such as the installation of equipment, thereby reducing the construction costs and duration.

7.3 Multi-objective optimization system for plant layout design

As described in the previous section, the layout design of a thermal power plant

requires excellent economic efficiency while satisfying the regulations and safety standards of layout restrictions. However, an excess of economic designs placed equipment and building structures close together, impairing the maintainability and expandability of plant facilities and hindering construction. Therefore, the layout optimization of a thermal power plant should be a multi-objective optimization problem that considers various evaluation indicators. However, the decision-making in an actual design situation operates not only on the basis of objective (non-subjective) evaluation indicators such as the five items discussed in the previous section but also on the basis of the aesthetics of power plants, uncertainty of design data, and tacit knowledge such as the intuition and know-how of skilled designers. This makes it difficult to obtain layout solutions using a completely automatic computerized layout solution method that can satisfy designers. In our opinion, an interactive layout solution method that allows the designer to participate in the decision-making of an optimal layout solution based on the coordination between the designer and the computer is an effective approach.

7.3.1 Interactive layout solution method

Figure 7.1 illustrates the structure of the multi-objective optimization system for plant layout design developed in this study. This system applies one of the interactive multi-objective programming techniques called the satisficing trade-off method (Nakayama, 1995) as its framework, and involves the fusion of computer intelligence and the designer's knowledge. The computer intelligence is mainly leveraged to generate various candidates of the optimal layout solution based on the layout optimization method by using a GA that will be described in Section 7.4. On the other hand, the designer's knowledge is mainly utilized while finalizing the layout adjustments for decision-making regarding the layout plan and selecting among the various candidates of the optimal layout solution suggested by the system. Furthermore, this system is capable of incorporating three-dimensional CAD tools and layout data, and can visually display in three-dimensional by synthesizing plot plan and the shape model of equipment and building structures in the basic planning and design phase. This function, which can also consider the aesthetics and spatial arrangement of the entire plant, is utilized when the designer conducts a more detailed evaluation and review, and makes it easy to carry over the data of the three-dimensional CAD tool in the detailed design phase.

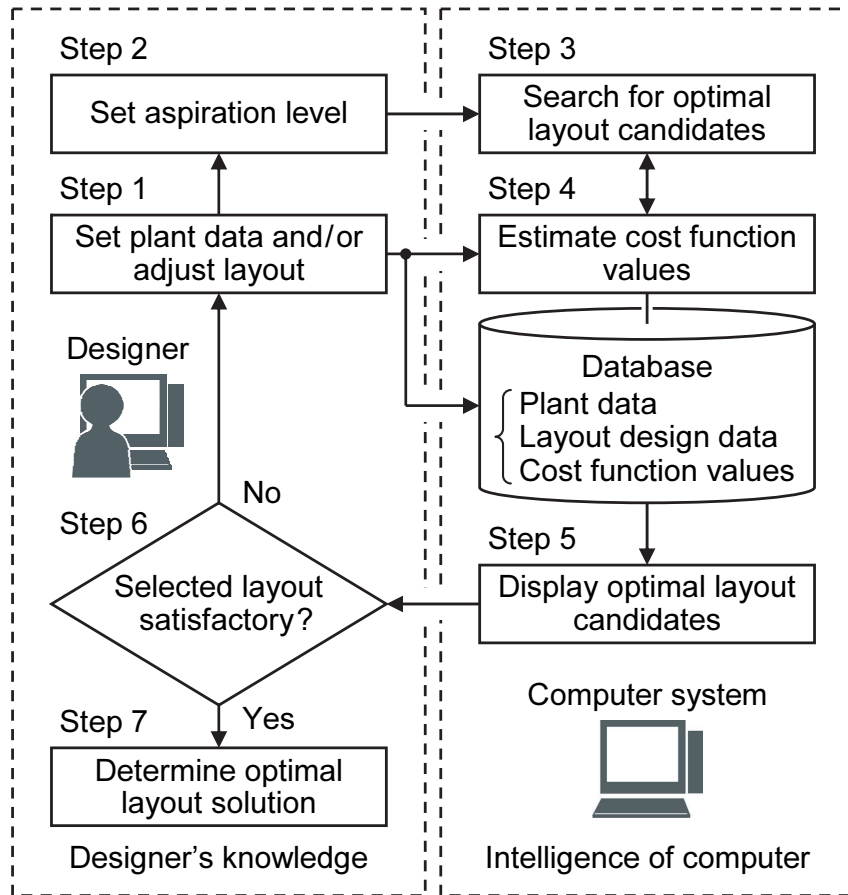


Fig. 7.1 Flowchart showing the functional structure of the proposed equipment layout design system. This system involves the fusion of computer intelligence and the designer's knowledge.

The satisficing trade-off method seeks the optimal layout solution, which is the Pareto-optimal solution as intended by the designer, by converting the multi-objective optimization problem into an auxiliary optimization problem by minimizing the scalarization function of the following equation:

$$\begin{cases} \min_{\mathbf{x}, z} z + \alpha \sum_{i=1}^r w_i [f_i(\mathbf{x}) - \bar{f}_i], \\ \text{subject to } \begin{cases} w_i [f_i(\mathbf{x}) - \bar{f}_i] \leq \beta_i z, & i = 1, 2, \dots, r, \\ \mathbf{x} \in \mathbf{X}, \end{cases} \end{cases} \quad (7.1)$$

where f_i is an evaluation function which formulized with the generalization of the five

items mentioned in Section 7.2 (to be described in Section 7.4.4), \overline{f}_i is the aspiration level (designer's target value corresponding to each f_i), and $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the design variable vector. \mathbf{X} represents the feasible region of \mathbf{x} , α denotes a small positive value (that is, 10^{-6}), and w_i is the weight factor and is automatically determined by using the following formula:

$$w_i = \frac{1}{f_i - f_i^*}, \quad (7.2)$$

where f_i^* is an ideal point and is generally given by $f_i^* \leq \min [f_i(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}]$. β_i in Eq. (7.1) is a factor that considers the arrangement constraints as an evaluation function. That is, $\beta_i = 0$ is a strong constraint (absolute constraint), and as it approaches $\beta_i = 1$, it becomes a weak constraint (conditional evaluation criteria). The reason for setting this adjustment parameter is that if the arrangement constraint is too strong, difficulties in the layout solution tends to arise, which should be avoided if possible. In the case of an actual design work, even when an arrangement constraint cannot be satisfied, it can nevertheless be bypassed by taking measures such as installing soundproof walls and fireproof walls afterwards.

7.3.2 Algorithm

Referring to Fig. 7.1, we will explain the algorithm of the proposed system. The database shown in the figure was constructed on the company's internal layout design CAD system, and the data within the database corresponds to the system. The algorithm of this system combines a main loop, which conducts the layout optimization computation of Steps 1–6, and a sub-loop, which allows manual layout adjustment in Step 1 and Steps 4–6.

Step 1: The designer sets the plant data (to be described in Section 7.4.1) and registers it into the database. Next, if a multi-objective optimization calculation is to be carried out, the designer advances to Step 2. However, if a manual adjusting layout result is selected in Step 6, the designer compiles the layout design data (installation of coordinates and angles) of the database and then goes back to Step 4.

Step 2: The designer sets the aspiration level values for each of the evaluation functions.

Step 3: A search for the candidates of an optimal layout solution close to the aspiration

level values set in Step 2 is performed through the evolution calculation with GA. At this time, the auxiliary optimization problem of Eq. (7.1) is solved by using the values of the evaluation functions estimated in Step 4.

Step 4: On the basis of the data in the database, each evaluation function value is estimated, and the layout design data and each related evaluation function value are registered into the database.

Step 5: While referring to the database, the system extracts each of the layout results close to each aspiration level value set in Step 2 and displays it as a candidate of the optimal layout solution.

Step 6: The designer evaluates the many layout results displayed in Step 5 from a comprehensive point of view and selects one desirable layout result. Next, he/she judges whether the selected layout result is satisfactory. If the designer is not satisfied, the process returns to Step 1, and then repeats Steps 1–6 (layout optimized calculation) or repeats the loop from Step 1 and Steps 4–6 (manual layout adjustment). If the designer is satisfied, the procedure proceeds to Step 7.

Step 7: The designer makes a final decision on the optimal layout solution from the layout result of Step 6.

7.4 Layout optimization method developed using GA

In this section, the coding method and the genetic computation method based on the suitable GA for the design of a plot plan are examined.

7.4.1 Plant data

Since a plot plan is designed in an upstream design phase, the design data tends to be limited. Therefore, it is important to abstract the design object and be able to present it using simple design data. The required plant data are described below, including the outer shape dimension, the relation of the connections, the group arrangement, and the site condition. Samples of the plant data are shown in Table 7.1. Here, the equipment and building structures in the table are all shown as rectangles.

7.4.1.1 Outer shape dimension

As shown in Fig. 7.2, the equipment and building structures are represented by the simple graphical elements of rectangles or circles and established from the outer shape

Table 7.1 Samples of the plant data

No.	Name	Size [m]		Group	Connecting of pipes, ducts, and cables														
		Length	Width		1	2	3	4	5	6	7	8							
0	Equipment A	12.0	16.0		To														
1	Equipment B	12.0	16.0			To		From											
2	Equipment C	8.0	16.0	Unit A			To												
	Equipment D	8.0	8.0							From									
3	Equipment E	8.0	16.0	Unit B				To											
	Equipment F	8.0	8.0								From								
4	Building A	8.0	8.0							To			From						
5	Building B	56.0	8.0											To				From	
0	Terminal A				From														
1	Terminal B																		To
2	Terminal C						From												
		Inlet connecting port			Center	Center	Center	Center	Center	Center	Center	Center	Center	Center	Center	Center	Center	Rear	Rear
		Outlet connecting port			Left	Right	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Front	Center
	Kind of connection (pipe, duct, or cable)	Number of pipe/duct/cable			Pipe	Pipe	Duct	Duct	Duct	Duct	Duct	Duct	Duct	Duct	Cable	Cable	Cable	Cable	Cable
	Unit cost of pipe/duct/cable [\$/m]	Number of pipe/duct/cable			2	2	1	1	1	1	1	1	1	1	1	1	1	8	8
	Unit cost of pipe/duct/cable [\$/m]	Number of pipe/duct/cable			800	800	800	800	800	800	800	800	800	400	400	400	200	200	200

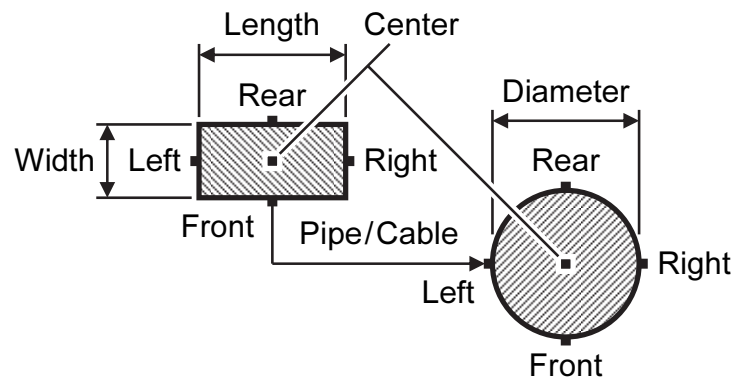


Fig. 7.2 Configuration of equipment/buildings. The connection ports of each graphical element are arranged at five locations in the center, front, rear, left, and right.

dimension, the length and the width (rectangle) or diameter (circle), of each graphical element.

7.4.1.2 Connections of pipes, ducts, and cables

As shown in Fig. 7.2, connecting ports are provided in the center, front, rear, left, and right of each graphical element. Further, from the plant system diagram and skeleton diagram, we can define the connections of pipes, ducts, and cables in the form of “from–to–” relations as shown in Table 7.1, and the positions of the connecting ports of the inlet (from side) and outlet (to side) can be assigned. For example, Table 7.1 shows a total of eight connecting relations. In addition, the type (pipe, duct, or cable), the number of connections, and the cost per unit length of each connection can be assigned. As an example, Fig. 7.2 shows a connecting relation from the front of a rectangular equipment unit (or building structure) to the left of a circular equipment unit (or building structure).

7.4.1.3 Group arrangement

The term “group arrangement” means simultaneously arranging the positions of several pieces of equipment and/or building structures of the plant. This is used in the following three ways:

Group arrangement 1. array arrangement: From the viewpoint of the customer’s usage, similar equipment and/or building structures can mostly be arranged in parallel

arrays or line-symmetrical or point-symmetrical arrays. In addition, plant facilities such as auxiliary equipment constituting several machines that have a mutual positional relationship can mostly be packaged together. For these kinds of equipment and/or building structures, the groups should be arranged such that they can maintain a mutual positional relationship.

Group arrangement 2. connecting arrangement: For example, a steam turbine and a generator are connected with a shaft. For this kind of equipment and/or building structure, the group should be arranged such that its constituent elements can be easily integrated.

Group arrangement 3. arbitrary shape: There are inconvenient situations when the outer shape of the equipment and/or building structure is a rectangle or circle, such as in the case of the items mentioned in Section 7.4.1.1. In these situations, a number of graphical elements are grouped to represent arbitrary shapes.

7.4.1.4 Site condition

The shape and the size of the construction site are set. Next, the access points of the facilities outside of the site, such as the fuel supply points, electric power outlets, water inlets, and water outlets, are set. These are assigned according to the location plan of the electric power company.

7.4.2 Coding method

In this study, the plant layout problem is considered by converting it into a mixed problem of schedule optimization, which determines the installation sequence of equipment and building structures, and combinatorial optimization, which determines the orientation, spacing, and locations of the installations. A high-speed computation can be achieved this way by efficiently seeking solutions in a narrowed design space.

Table 7.2 shows an example of the chromosome of GA. First, the length of the

Table 7.2 Example of the chromosome of GA

Identification number	1	3	4	0	2	5
Direction	2	1	1	0	3	0
Maintenance space	2	1	1	1	0	0
Installation point	0	2	1	2	1	2

chromosome is set to equal the number of equipment units (hereinafter, also including building structures). However, the several pieces of equipment in a group arrangement are considered a single entity, as described in Section 7.4.1.3. Each genetic locus of the chromosome represents the equipment identification number and the direction, maintenance space, and installation point of these equipment units.

The direction gene is provided in the gene in order to obtain the appropriate orientation of the equipment. As illustrated in Fig. 7.3, using the origin as the reference point, gene 0 is defined as 0° ; gene 1, 180° ; gene 2, 90° ; and gene 3, 270° . In addition, by limiting the adjustable value of the gene number corresponding to the equipment, the installation angle can be assigned.

The maintenance space gene is provided in order to maintain appropriate maintenance space among the equipment units. A number of p maintenance space frames are prepared around the equipment, and the distance between maintenance spaces is decided and maintained on all four sides (front, rear, left, and right) of each equipment unit. For example, Fig. 7.4 illustrates a case of three maintenance space frames where gene 0 has no maintenance space (i.e., the outer shape of the equipment),

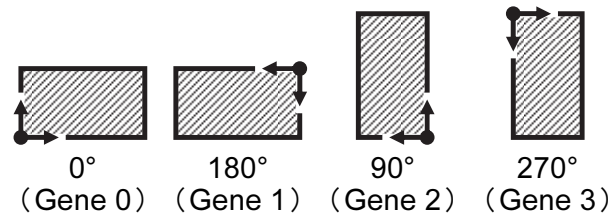


Fig. 7.3 Directions of equipment/building. Installation angles are four directions (0° , 90° , 180° , and 270°).

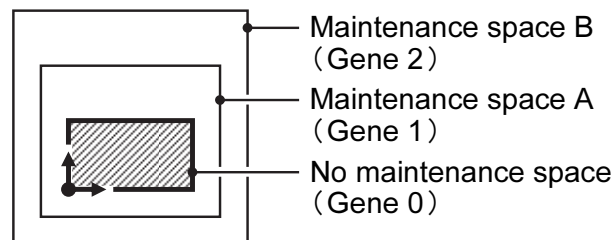


Fig. 7.4 Maintenance spaces around equipment/building. For example, this figure illustrates a case of three maintenance space frames.

gene 1 defines the area of maintenance space A, and gene 2 defines the area of maintenance space B.

The installation point gene is provided in order to obtain the appropriate location of the equipment. The installation point can be set at any location on the site, but when the access points described in Section 7.4.1.4 are considered, a reasonable arrangement can be easily obtained. The q installation points correspond to gene 0 to gene $q - 1$, and the equipment are installed such that the distance between the installation point indicated by the corresponding gene number and the central point of the equipment is the shortest. The example shown in Fig. 7.5 shows the case of installing an equipment unit with a direction of 0° in the maintenance space A nearest to installation point A (gene 0).

Next, the base number array is used as a rule to convert the chromosome into a phenotype. The base number array is any array of natural numbers corresponding to the identification numbers of the equipment and determines the order of installations of the equipment inside the site. The array of the identification numbers of the equipment is designated as the gene array of the chromosomes. The number of the base number array that is in the same rank as the number that corresponds to the order of this series is removed; it replaces the line of the gene array of the phenotype. For example, the gene array in Table 7.2 is (1, 3, 4, 0, 2, 5). Hence, when the base number array is (0, 1, 2, 3, 4, 5), the gene array of the phenotype is (1, 3, 4, 0, 2, 5); when the base number array is (5, 4, 3, 2, 1, 0), the gene array of the phenotype is (4, 2, 1, 5, 3, 0); and when the base number array is (3, 2, 5, 0, 4, 1), the gene array of the phenotype is (2, 0, 4, 3, 5, 1). The rules of installing equipment with this phenotype are explained below:

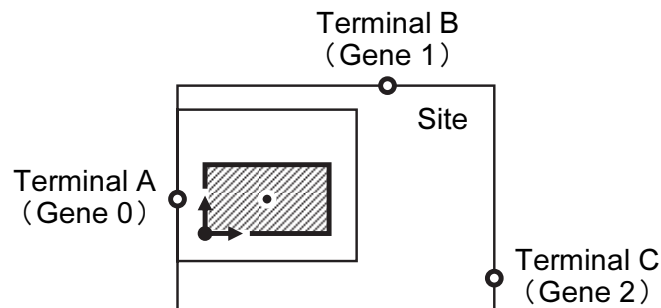


Fig. 7.5 Installation points with respect to equipment/building. For example, this figure illustrates a case of three installation points.

Rule 1: According to the sequence of the gene array of the phenotype, set the equipment of the corresponding identification numbers in the site with the direction, maintenance space, and installation point held by each genetic locus.

Rule 2: When setting the equipment according to Rule 1, avoid overlap between the areas of the maintenance space corresponding to the gene numbers of the equipment already set and the area of the maintenance space corresponding to the gene number of the equipment to be set. If there is no space to set the equipment inside the site, it should be set protruding outside of the site such that the area of the protruding part is minimized.

Rule 3: While following the rule regarding the maintenance space, as described in Rule 2, installations are to be done in such that the distance from the installation point corresponding to the gene number of the equipment to be installed to the central point of the equipment is the shortest. If there are multiple positions with the shortest distance, randomly select one from among these. Furthermore, when making arrangements of a group, calculate the distance of each equipment unit in the group and designate the shortest one as the representative distance.

7.4.3 Considering the arrangement constraints

In the coding method described in the previous section, some facilities cannot satisfy the arrangement constraints. Therefore, in order to be able to preserve the arrangement constraints, constraint conditions are added. Arrangement constraints can be classified into three types as described below. Moreover, the spacing distance of each arrangement and constraint type corresponding to all four directions (front, rear, left, and right) of each equipment unit (hereinafter, also including building structure) is set. When there are multiple constraints belonging to the same type of arranged constraint, the largest spacing distance is considered the representative distance. Figure 7.6 shows the types of overlap between the area (dotted line frame) of each arrangement constraint type of equipment A and equipment B on the boundary of the site.

Type A arrangement constraint: As illustrated in Fig. 7.6a, this area is not allowed to protrude out of the site but permits other equipment inside. This type is seen in the case of environmental regulations of noise and the like.

Type B arrangement constraint: As illustrated in Fig. 7.6b, this area is not allowed to

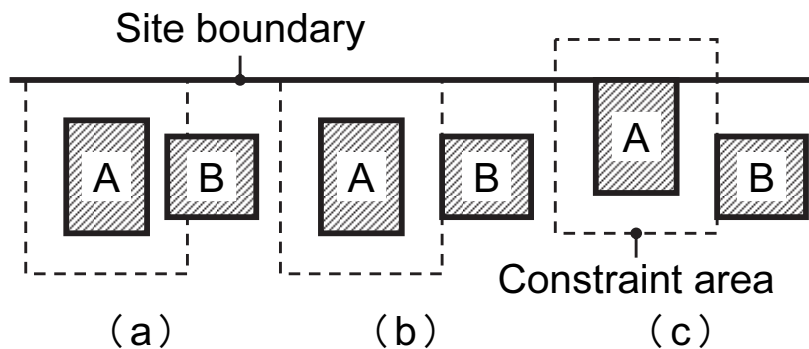


Fig. 7.6 Types of arrangement constraints. (a) Type A area is not allowed to protrude out of the site but permits other equipment inside. (b) Type B area is not allowed to protrude out of the site nor are other equipment allowed inside. (c) Type C area is allowed to protrude out of the site but does not allow other equipment inside.

protrude out of the site nor are other equipment allowed inside. This type is seen in the case of safety regulations pertaining to dangerous and combustible materials, etc.

Type C arrangement constraint: As illustrated in Fig. 7.6c, this area is allowed to protrude out of the site but does not allow other equipment inside. This type is seen in the case of the safety rules of doors and passages, etc.

Next, with respect to the arrangement constraints, the procedure for installing equipment can be explained as follows:

Step 1: Based on the arrangement constraints of the equipment already installed and the arrangement constraints of the equipment to be installed, select possible installation areas from the site according to the arrangement constraints. Install the equipment in these possible installation areas according to Rules 1–3 described in the previous section.

Step 2: If no possible installation area can be found by following the arrangement constraint in Step 1, re-select from the same areas by removing the arrangement constraints of the equipment to be installed, and install the equipment according to Rules 1–3 described in the previous section.

Step 3: If no possible installation area can be found in Step 2, remove all of the arrangement constraints of the equipment already installed, and install the equipment according to Rules 1–3 as described in the previous section.

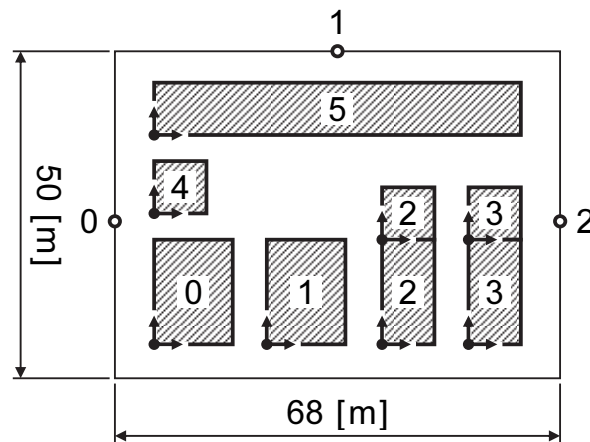


Fig. 7.7 Design data of the equipment, building structures, and the site condition, referring to the plant data in Table 7.1.

Step 4: If there is no place for an installation inside the site even in Step 3, the equipment is to be set protruding outside of the site such that the protruding area is minimized.

As an example, referring to the plant data in Table 7.1, the phenotype arrangements based on the design data of the equipment, building structures, and the site condition shown in Fig. 7.7 are explained. However, the equipment units with identification numbers 2 and 3 are to be arranged in a group and the spacing distances of maintenance spaces A and B are set as 4 m and 8 m in all directions, respectively, with respect to all equipment and building structures. Further, the spacing distance for the Type A arrangement constraint is set to be 6 m in all four directions for the equipment units with the identification numbers 0 and 1. The spacing distance for the Type B arrangement constraint is set to be 2 m in all four directions for equipment units with the identification numbers 0 and 1, and building structure with the identification number 4. Moreover, the spacing distance for the Type B arrangement constraint is set to be 4 m on the front side and 2 m on the rear, left, and right sides for the equipment units with the identification numbers 2 and 3. The spacing distance for the Type C arrangement constraint is set to be 2 m in all four directions of the building structure with the identification number 5. Here, when (0, 1, 2, 3, 4, 5) is the base number array, the installation process followed by the chromosome mentioned in Table 7.2 [with gene array (1, 3, 4, 0, 2, 5) of the phenotype] is shown in Figs. 7.8a–7.8f. The solid line

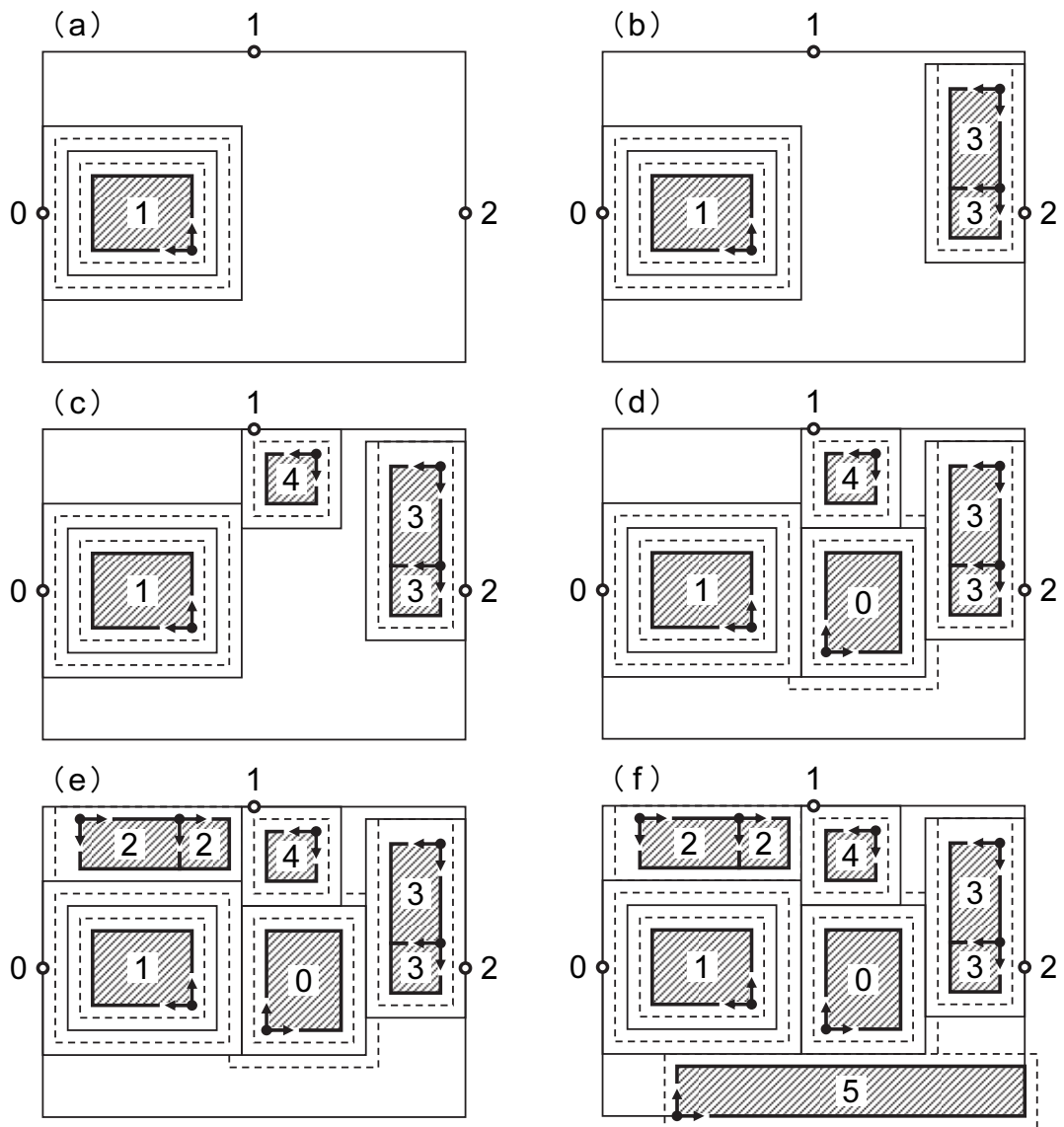


Fig. 7.8 Coding method and phenotype of plant layout. This figure shows the installation process followed by the chromosome mentioned in Table 7.2 with gene array (1, 3, 4, 0, 2, 5) of the phenotype.

frames around the installed equipment and building structures are areas for maintenance space, and the dotted line frames represent the areas of each type of arrangement constraint.

7.4.4 Evaluation functions

The evaluation function generalizes the five evaluation indicators mentioned in

Section 7.2, combined with the eight items formulated below.

Evaluation indicator of the cost of pipes and ducts: The reduction of the total cost of pipes and ducts. The cost breakdown is divided into pipes/ducts (item ①) and cables (item ②) because the extent of the price risk of both differs depending on the construction project.

① **Total cost of pipes and ducts:** The length of the pipes and ducts is displayed in Manhattan distances, and the total cost is calculated by multiplying the total length with the cost per unit. In this study, we deal with a plane arrangement, but by setting the height positions of the connecting ports, the difference in elevation can also be considered.

Evaluation indicator of the cost of cables: The reduction of the total cost of cables.

② **Total cost of cables:** The length of the cables is displayed in Manhattan distances, and the total cost is calculated in the same way as that mentioned in item ①.

Evaluation indicator of the maintainability and expendability of plant facilities:

A reduction in the overlap in the maintenance space, as described in Section 7.4.2.

③ **Interference area of maintenance space:** Considering the outermost area of the maintenance space, calculate the total area occupied by other equipment and building structures. There are times when the total area of the part protruding out of the site is added because of the design.

Evaluation indicator of the ease of construction works: A reduction in the degree of difficulty of construction works.

④ **Degree of difficulty of construction works:** With respect to main equipment such as boilers and turbines that affect the cost and duration of construction works, if necessary, items ①–③ are individually evaluated, and the degree of difficulty of construction works is calculated.

Overall evaluation indicator: The reduction of the area of the equipment and building structures outside the site. Moreover, for this proposed method, the equipment and building structures do not overlap.

⑤ **Area of the equipment and building structures outside the site:** The total area of the part of the equipment and building structures protruding out of the site is calculated.

Evaluation indicators of legality and safety: The reduction of the violated area of the arrangement constraint corresponding to each arrangement constraint type listed in Section 7.4.3.

⑥ **Violating area according to Type A arrangement constraint:** Considering the area corresponding to the Type A arrangement constraint, the total area of the part protruding out of the site is calculated.

⑦ **Violating area according to Type B arrangement constraint:** Considering the area corresponding to the Type B arrangement constraint, the total area of the part protruding out of the site and the part overlapping with other equipment and building structures is calculated.

⑧ **Violating area according to Type C arrangement constraint:** Considering the area corresponding to the Type C arrangement constraint, the total area of the part interfering with other equipment and building structures is calculated.

In this study, the β_i value calculated using Eq. (7.1) is a strong constraint when $\beta_5 = 0.1$ for item ⑤; items ⑥–⑧ are weak constraints when $\beta_i = 0.5$ ($i = 6, 7, 8$), and all other items ①–④ are adjusted to the objective functions $\beta_i = 1.0$ ($i = 1, 2, 3, 4$).

7.4.5 Genetic computation method

The procedure for seeking a solution with a GA using the proposed method is explained. First, randomly generate N initial parent generation individuals that have already been established ($N = 30$ in this study) and base number arrays. At this time, try to avoid overlapping the identification numbers of the equipment and building structures, and seek a quasi-optimal layout solution through an evolutionary calculation of evaluation and selection, crossover, mutation, and mutation of foreign species, which are explained below. Note that crossover, mutation, and mutation of foreign species are methods that prevent the repetition of the identification numbers of equipment and building structures in order to avoid the generation of lethal genes. In addition, to speed up the calculations, when an individual that is the same phenotype of the gene array is formed, the layout solution from the past calculation records are referred to, and the calculation of that individual is abbreviated.

7.4.5.1 Evaluation and selection

The fitness function, F_j , of each individual is calculated by scalarizing the multiple evaluation functions of the previous section (see Section 7.4.4) from Eq. (7.1) and utilizing the sigmoid function to scale the non-negative functions with the following formula. Further, when selecting the survival of the fittest, a roulette selection using the

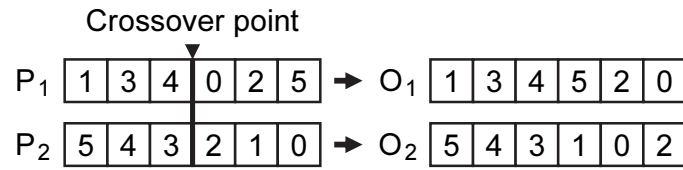


Fig. 7.9 Crossover. This shows the crossover method of the gene array for the installation order of the equipment and building structures.

fitness proportional system and the elite retention selection that passes on the best individual to the next generation are used together. Moreover, N individuals (next generation parent individuals) are selected from among parent individuals and child individuals.

$$F_j = 100 - \frac{95}{\exp[-a(S_j - b)]}, \quad (7.3)$$

where S_j is the scalarization function value given by Eq. (7.1), whereas a and b are variable parameters adapted to individual distribution. Subscript j is the number of the individual.

7.4.5.2 Crossover

The subjects of the crossover, the gene array for the installation order of the equipment and building structures, and other gene arrays (direction, maintenance space, and installation point), are each considered at a specified crossover rate. The crossover method of the former is shown in Fig. 7.9, and the single crossover point is decided randomly. Next, the child individual O_1 inherits the order of the gene numbers from the parent individual P_1 until the crossover point, and after that, the child inherits the gene numbers that have already been chosen in the order that they have been removed from the parent individual P_2 . Similarly, the child individual O_2 inherits the order of the gene numbers from the parent individual P_2 until the crossover point, and after that, the child inherits the gene numbers that have already been chosen in the order that they have been removed from the parent individual P_1 . The latter crossover method decides multiple crossover points randomly and sets the multiple point crossovers that exchange the gene numbers at the boundary of the crossover point.

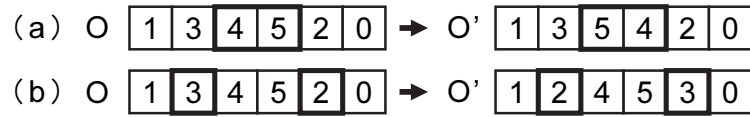


Fig. 7.10 Mutations. This shows the two mutation methods of the gene array for the installation order of the equipment and building structures.

7.4.5.3 Mutation

The subjects of mutation, the gene array with respect to the installation order of the equipment and building structures, and other gene arrays (direction, maintenance space, and installation point), each occur at a specified mutation rate. The former is mutated by the two methods as shown in Fig. 7.10. Figure 7.10a shows how the contiguity order of the gene number's arbitrary position can be exchanged, while Fig. 7.10b shows how the two randomly chosen gene numbers are exchanged. The latter mutation is to be a normal mutation that changes the gene number of the element that is chosen randomly.

7.4.5.4 Mutation of foreign species

When dealing with the general arrangement problem for power plants, even with the same degree of fitness function values, a great difference in layout solutions exist; hence, there is a problem of premature convergence phenomena depending on the setting of a base number array. In this study, a new invention is carried forward where during the evolution of dozens of generations; the base number array is replaced with the best individual of the phenotype of the gene array discovered up until then. Next, the best individual chromosome is adapted to the new base number array, and the remaining individuals are used for choosing different phenotypes even if they are the same chromosome. According to this method, with a large-scale optimal solution, the entire gene array of the chromosome lines up to become $(0, 1, 2, \dots, m-1)$, which shows the rationality of the results as well as the expectation of achieving various layout solutions. Here, m is the number of equipment and building structures.

7.5 Application results

7.5.1 Application results for trial design problem

Utilizing the example of Fig. 7.7 shown in Section 7.4.3 (refer to the plant data in

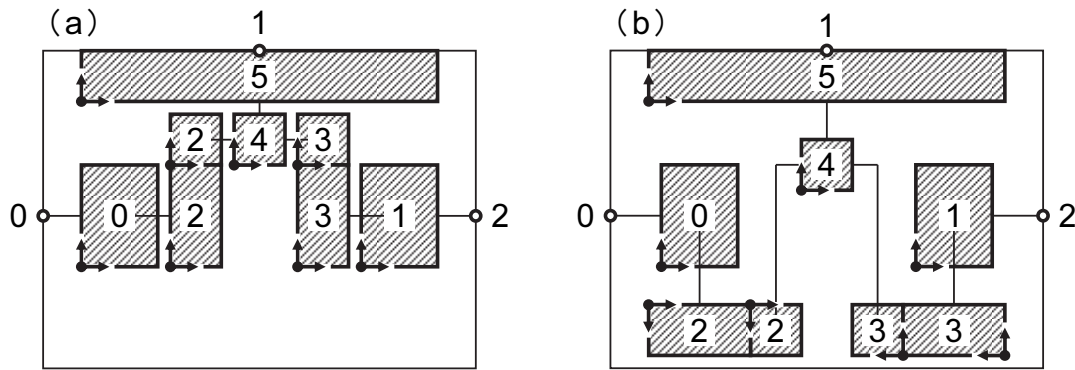


Fig. 7.11 Results for each aspiration level. (a) Prioritizing pipe/duct and cable costs (Case 1). (b) Prioritizing maintenance space (Case 2).

Table 7.1), we calculate a multi-objective optimal layout solution. The total number of combinations possible with a GA using the proposed method is $(4 \times p \times q) \times m!$, wherein m is the number of equipment and building structures, p is the number of genes per maintenance space, q is the number of genes per installation point, and four is the number of genes per direction. In the example shown in Fig. 7.7, the number of equipment and building structures is six, the number of genes per maintenance space is three, and the number of genes per installation point is three, making the total number of combinations 25 920. Even though this is a small-scale layout problem, calculating an optimal layout solution corresponding to the aspiration level of the designer by trial and error requires a fair amount of time and effort.

The computation conditions include Case 1, where the total cost of pipes and ducts, f_1 , and the total cost of cables, f_2 , are the priorities, and Case 2 where the interference area of the maintenance space, f_3 , is the priority. In particular, the values of each aspiration level are $(\overline{f_{1,1}}, \overline{f_{2,1}}, \overline{f_{3,1}}) = (\$32\,000, \$8000, 1344\text{ m}^2)$ for Case 1, and $(\overline{f_{1,2}}, \overline{f_{2,2}}, \overline{f_{3,2}}) = (\$41\,600, \$35\,200, 200\text{ m}^2)$ for Case 2. Next, a series of trial calculations with various random numbers was conducted 10 times (initial individuals also differ because of the random numbers). Furthermore, the various parameters of a GA are individual numbers of 30, crossover rate of 60%, and mutation rate of 10%, while adding a mutation of foreign species every 15 generations and carrying on a search that can achieve an optimal layout solution in a certain number of generations.

Trial computations were conducted 10 times for both Case 1 and Case 2. On average, Case 1 took 72 (smallest 53, largest 90) generations, and Case 2 took 136 (smallest 85,

largest 172) generations to obtain the optimal layout solution where each reached the aspiration level. The optimal layout solution obtained for Case 1 is shown in Fig. 7.11a and that obtained for Case 2 in Fig. 7.11b. Both cases satisfied all arrangement constraints and such examples where Case 1 showed that it has a more compact layout than Case 2, displayed that the appropriate optimal layout solution was obtained in proportion to the aspiration levels. Furthermore, the calculation time per generation was only 4 minutes when a Toshiba notebook PC dynabook™ SS RX2 (processor operating frequency: 1.20 GHz) was used.

7.5.2 Application results for thermal power plants

The application of the proposed method for a general arrangement problem of a gas-turbine combined-cycle power plant (total output: 280 MW) is described as an example. An alternative general arrangement plan drawn with the former design plan by a skilled designer with 20 years of experience is compared to a layout optimized calculation done by a GA. Figure 7.13a presents an onsite layout diagram, wherein a skilled designer lays out equipment and building structures by a conventional design procedure. Hereinafter, each evaluation function value of the layout results obtained by the skilled designer (Fig. 7.13a) is standardized as 100. The calculation conditions here are as follows: the aspiration level in Eq. (7.1) with respect to each evaluation function

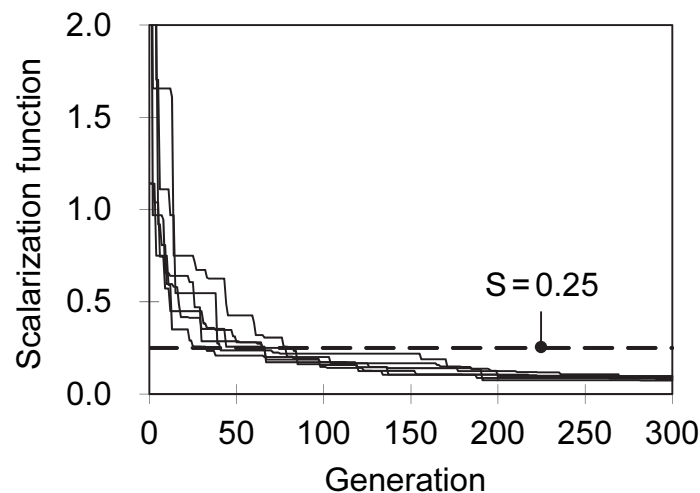


Fig. 7.12 Convergence history of the proposed GA. This shows the calculation history of the scalarization function value of the best individual of each generation in each of the six trial calculations.

Table 7.3 Application results through six trial calculations by the proposed GA

No.	f_1	f_2	f_3	S
1	92.9	91.1	92.5	0.0734
2	93.8	93.3	86.3	0.0957
3	93.9	93.3	93.8	0.0970
4	93.2	91.2	93.1	0.0806
5	93.5	89.1	87.5	0.0872
6	93.5	92.5	86.9	0.0872

is 90, and the ideal point in Eq. (7.2) is designated as a value that equals 50. Furthermore, each of the GA parameters is as follows: the individual numbers are 30; the generation number is 300; the crossover rate is 60%; the mutation rate is 10%; and the mutation of foreign species is introduced every 15 generations. Further, in order to understand the tendency of the solution search, six trial calculations with changing random numbers were conducted (initial individuals differ because of the random numbers).

Figure 7.12 shows the history of the GA's solution search by overlapping the scalarization function value [obtained using Eq. (7.1)] of the best individual of each generation in each of the six trial calculations. Further, on average, a better layout solution than the one provided by the skilled designer (scalarization function value of under 0.25, as illustrated by the dotted lines in the figure) was reached at generation 59 (smallest 33, largest 83). Each subsequent trial calculation reached near-optimal layout solutions in only 200 generations. This showed that a relatively superior layout solution could be quickly found with a GA using the proposed method. Table 7.3 shows each evaluation function value (total cost of pipes and ducts, f_1 , total cost of cables, f_2 , and the interference area of maintenance space, f_3) of the best individual obtained after 300 generations out of the six trial computations and the scalarization function value, S . Although the trial computations satisfied all of the arrangement constraints and obtained good layout solutions not inferior to practical applications, they did not reach completely identical layout solutions. The reason for this is discussed below. The minimization of the scalarization function of Eq. (7.1) is equivalent to a maximal component minimization method (min-max solution). In Table 7.3, f_1 becomes the largest component in each of the trial calculations, and because of the problem of the

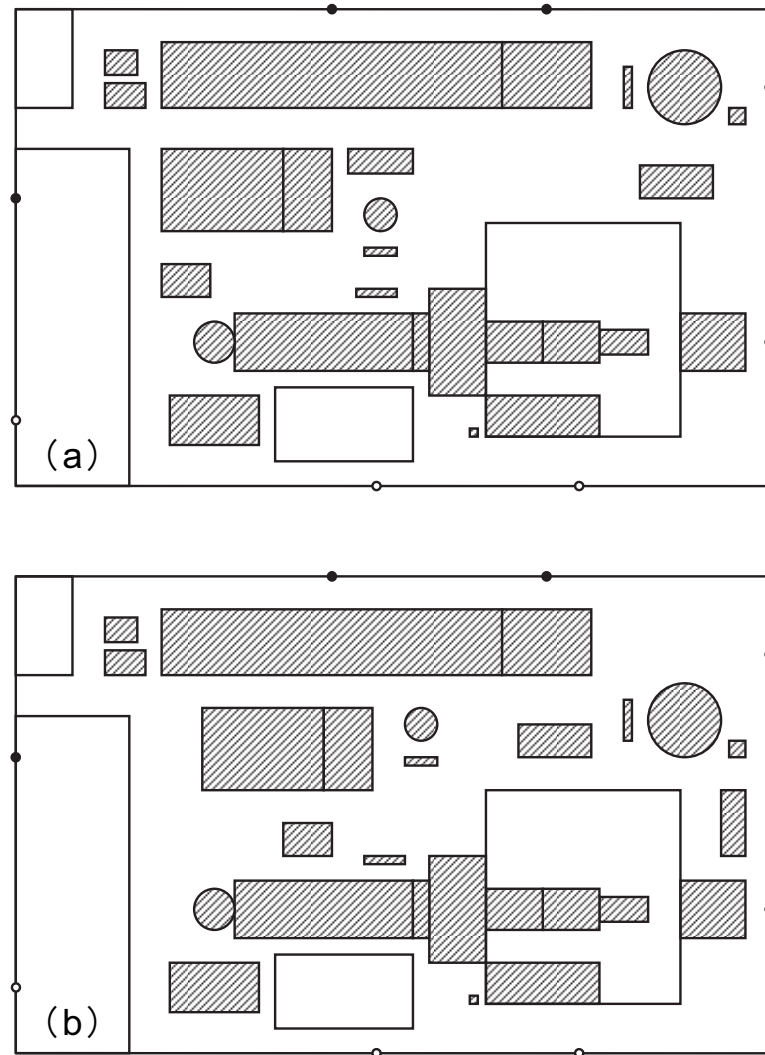


Fig. 7.13 Design equipment layout of the plot plan for a gas-turbine combined-cycle power plant. A comparison between layouts created by (a) a skilled designer and (b) the proposed GA method.

GA having the ability to do a fine solution search, a variation of approximately 1% (from 92.9 to 93.9) occurs. Further, since the multiple weakly Pareto-optimal solutions that exists within this variation are obtained, some variations in f_2 and f_3 can be observed.

Figure 7.13b shows the optimal layout solution obtained by the first trial calculation shown in Table 7.3. Figure 7.13b (layout result obtained by the proposed GA) and the layout result of the skilled designer (Fig. 7.13a) are similar, but as shown in Table 7.3, each evaluation function value of the layout result obtained by a GA is superior.

Furthermore, there is sufficient space among the equipment and building structures, and there are no problems related to the completion of the construction work. According to the skilled designer, the arrangement of the layout result by a GA needs to be adjusted so that it considers the laying routes for roads, pipe racks, and trenches, and the aesthetics of the arrays of the equipment and building structures, but otherwise, an appropriate layout solution is obtained. As described in Section 7.3.2, this suggests that this system can produce efficient layout designs by the combined use of a layout optimized calculation and a manual layout adjustment.

7.6 Conclusions

In this study, a plant layout design was considered to be a multiple-criteria decision-making problem, and an interactive design support method that exploits both a skilled designer's expertise and a computer's processing ability for obtaining an optimal layout solution that the designer desired was proposed. In other words, a multi-objective optimization system for the plant layout design with a satisficing trade-off method as the framework was constructed, and a layout optimization method based on a GA suited for the basic planning and design of a plot plan was proposed. Next, by applying this method to the general arrangement problem of an actual gas-turbine combined-cycle power plant, we verified the usefulness of the proposed method.

The results of this study are summarized as follows:

1. We proposed a novel interactive layout solution method based on the coordination between a designer and a computer, and developed a multi-objective optimization system that supports plant layout design for a plot plan.
2. We proposed a layout optimization method using a GA that can generate various layout solutions at high speed.
3. We applied this to the layout design problem of an actual gas-turbine combined-cycle power plant installation, and verified the usability of the proposed method by comparison with the layout results of a skilled designer.

Finally, some issues that will be considered in the following chapters are as follows:

1. The proposed method can obtain only quasi-optimal solutions in a design space that depends on the coding method of the GA. Although the fact that a manual layout

adjustment carried out by a designer is necessary in the final stage in each case, it does not immediately impair the usefulness of the proposed method, but rather, the establishment of a more flexible design support method is expected to improve a designer's workability.

2. Further, when basic planning and design of a plot plan, not only the layout of the equipment and building structures but also the route design of the pipe conduits and electric circuits is important.

The above two issues are considered in Chapters 8 and 9 of Part III, respectively.

References

- Castell, C. M. L., Lakshmanan, R., Skilling, J. M. and Bañares-Alcántara, R., Optimisation of process plant layout using genetic algorithms, *Computers and Chemical Engineering*, Vol. 22, Supplement 1, (1998), pp. S993–S996.
- Fuchino, T., Itoh, T. and Muraki, M., Arrangement of process equipment modules with consideration of plant safety, *Journal of Chemical Engineering of Japan*, Vol. 30, No. 5, (1997), pp. 896–901.
- Fujita, K., Akagi, S., Hase, H., Nakatogawa, T. and Takeuchi, M., Plant layout design system using a hybrid approach with a constraint-directed search and a mathematical optimization technique, *JSME International Journal, Series C, Dynamics, Control, Robotics, Design and Manufacturing*, Vol. 36, No. 4, (1993), pp. 562–568.
- Fujita, K., Akagi, S. and Doi, H., Automated acquisition of constraints in plant layout design problems, *JSME International Journal, Series C, Dynamics, Control, Robotics, Design and Manufacturing*, Vol. 38, No. 2, (1995), pp. 334–341.
- Guirardello, R. and Swaney, R. E., Optimization of process plant layout with pipe routing, *Computers and Chemical Engineering*, Vol. 30, No. 1, (2005), pp. 99–114.
- Mierswa, I., Incorporating fuzzy knowledge into fitness: multiobjective evolutionary 3D design of process plants, *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation (GECCO2005)*, (2005), pp. 1985–1992, Washington, D.C., the United States.
- Nakayama, H., Aspiration level approach to interactive multi-objective programming and its applications, In: *Advances in Multicriteria Analysis* (Pardalos, P. M.,

- Siskos, Y. and Zopounidis, C., eds.), (1995), pp. 147–174, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Patsiatzis, D. I., Knight, G. and Papageorgiou, L. G., An MILP approach to safe process plant layout, *Chemical Engineering Research and Design*, Vol. 82, Issue 5, (2004), pp. 579–586.
- Penteado, F. D. and Ciric, A. R., An MINLP approach for safe process plant layout, *Industrial and Engineering Chemistry Research*, Vol. 35, Issue 4, (1996), pp. 1354–1361.
- Suzuki, A., Fuchino, T., Muraki, M. and Hayakawa, T., An evolutionary method of arranging the plot plan for process plant layout, *Journal of Chemical Engineering of Japan*, Vol. 24, No. 2, (1991a), pp. 226–231.
- Suzuki, A., Fuchino, T., Muraki, M. and Hayakawa, T., Equipment arrangement for batch plants in multi-floor buildings with integer programming, *Journal of Chemical Engineering of Japan*, Vol. 24, No. 6, (1991b), pp. 737–742.

Chapter 8

Application of particle swarm optimization to automatic placement and adjustment of plant systems

Abstract

It is difficult for a designer to optimize a plant layout design because the related decision-making involves multiple objectives and constraints such as safety, economic costs, maintainability, and construction term of works. In order to support the designer's decision-making, we have developed a multi-objective optimization system for plant layout design; this system involves an effective interaction between the designer and the computer. In this study, we have proposed a hybrid optimization technique for plant layout design using a genetic algorithm (GA) and particle swarm optimization (PSO). In the first step, the designer globally searches for a layout solution on the basis of a coding rule of the GA proposed in Chapter 7. However, the obtained layout solution in this step is rough and is not adequately optimal because the GA is a combinatorial optimization algorithm that uses discrete design variables. Then, in the second step, the obtained layout solution is automatically corrected toward a better position by using a modified PSO algorithm for continuance design variables. The proposed method is applied to the layout design problem of an actual thermal power plant in order to demonstrate the validity of this approach.

Keywords

Design engineering, Engineering optimization, Genetic algorithm, Layout design, Multi-objective optimization, Particle swarm optimization, Plant layout, Power plant, Satisficing trade-off method, Systems engineering

8.1 Introduction

Layout design of systems with many interrelated components often occurs in industrial settings, e.g., layout of components for applying electronic parts on

substrates, component parts of internal combustion engines, production lines within factories, or plant facilities in the heavy chemical industry. In this kind of layout designs, many evaluation indices such as functionality, economy, safety, maintainability, and designability interact in complicated ways, and the evaluation indices may be different depending on what is to be designed. In other words, a basic difficulty of system layout design lies in the multiple-criteria decision-making that a designer faces when optimizing multiple conflicting evaluation indices.

Therefore, we have chosen a general arrangement problem (also called plot plan) for a thermal power plant as a design objective, and developed a multi-objective optimization system for plant layout design using an interactive multi-objective programming technique. In Chapter 7, we showed the structure of this system and proposed a layout optimization method using a genetic algorithm (GA) appropriate for the basic planning and design of the plot plan. However, the layout optimization method using this GA has the practical problem that only quasi-optimal solutions can be obtained within the design space, depending on the genetic code that is used. The reason is that, whereas the real design space for the layout design has installation coordinates and installation angles with continuous values (to be precise, they are discontinuous because there are arrangement constraints), in the GA these are put into a combined rule base optimization problem and solutions are sought in a design space of discrete quantities. As a result, not only does the obtained layout solution not correspond to the true global optimal solution (or a corresponding suboptimal solution) but it also is possible that in some cases it will not achieve an excellent layout solution that can sufficiently satisfy the designer. This type of phenomenon is a problem that commonly occurs in layout optimized calculations with GAs. For example, it is also seen in the research of Yoshikawa et al. into printed-circuit-board automatic layout using a GA (Yoshikawa et al., 1994). Yoshikawa et al., from a practical viewpoint, present a plan that increases the information (design rules) that is allocated to genes and increase the resolution of the design space. However, the ability to search for a solution worsens, so that the intervention of a human (the designer) is necessary to eliminate overlapping of electronic parts (Yoshikawa et al., 1995).

The purpose of this study is to establish a more practical design support method that compensates for the above limitations found in the layout optimized calculation with GA. Specifically, we present a hybrid multi-objective optimization method for plant layout design that first uses the layout optimized calculation according to the GA of the

previous study (see Chapter 7 for details) to globally search the design space for a solution. Then, it uses particle swarm optimization (PSO) (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 2001) to approach the true global optimal solution (or a corresponding suboptimal solution) near a layout solution obtained by the GA.

The structure of this chapter is as follows. First, Section 8.2 summarizes the multi-objective optimization system for plant layout design of the previous study (see Chapter 7), and its problems. Next, Section 8.3 provides a summary of PSO, and Section 8.4 explains an automatic layout adjustment method using PSO. Then, Section 8.5 verifies the usability of this proposed method with an application example. Finally, Section 8.6 presents our conclusions.

8.2 Multi-objective optimization system for plant layout design

Here we give a simple description of the multi-objective optimization system for plant layout design of the previous study (see Chapter 7). We also describe the practical problems of this system.

8.2.1 Overview of the previous study

In Chapter 7, we constructed an interactive design support system that incorporates many evaluation indices related to plant layout design and finds an optimal layout solution based on cooperation between the designer and the computer. This system applies the designer's multiple-criteria decision-making process to the results of a satisficing trade-off method (Nakayama, 1995) to create an interactive layout solution method. This method uses a GA to obtain multiple layout solutions that become solution candidates for the decision-making. The layout optimization method using GA in Chapter 7 treats the plant layout problem as a mixed problem of schedule optimization, which determines the installation sequence of the equipment (hereinafter, also including building structures), and combinatorial optimization, which determines the directions, maintenance spaces, and installation points when installing the equipment. The chromosome of the GA is expressed as sets of equipment numbers ($i = 1, 2, \dots, m$), installation angles ($0^\circ, 90^\circ, 180^\circ, \text{ and } 270^\circ$), p maintenance space frames (A_1, A_2, \dots, A_p) that define the spacing distances from equipment perimeters, and q installation points (P_1, P_2, \dots, P_q) on the site, for the respective equipment. The length of the chromosome is equal to the number of equipment units.

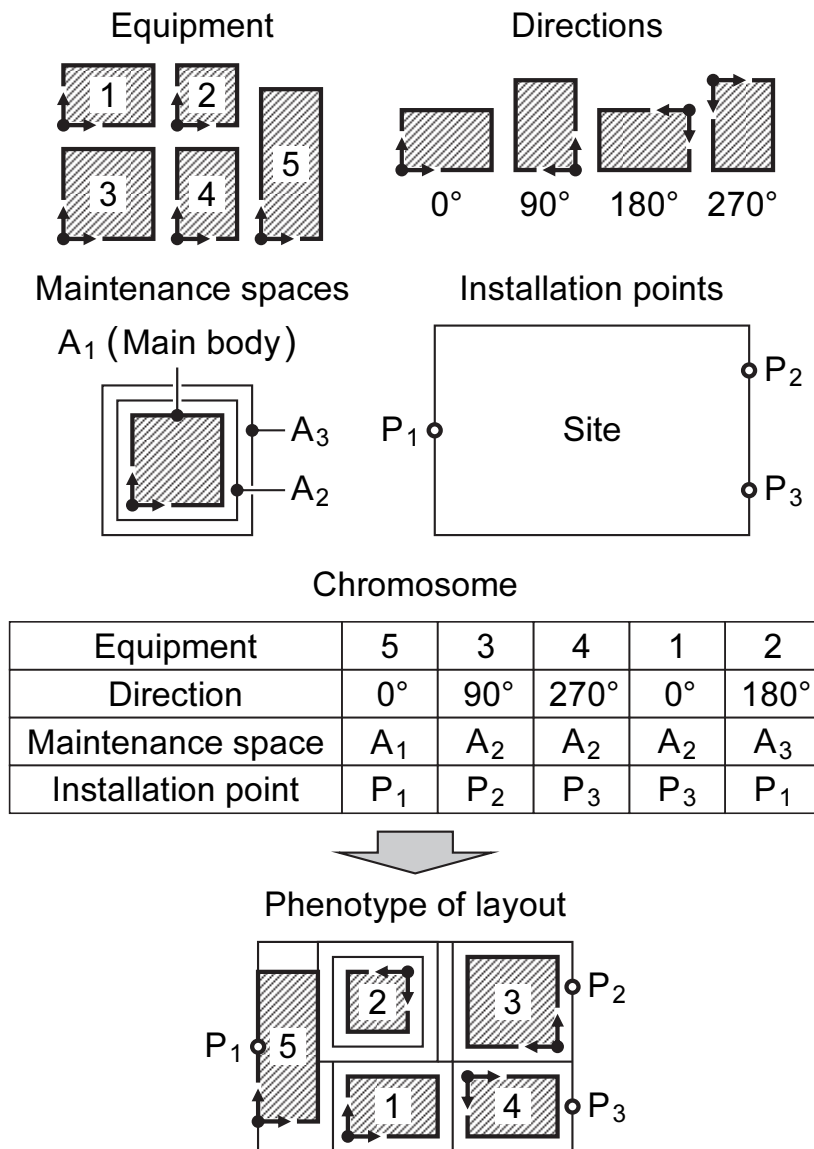


Fig. 8.1 GA coding method. This figure shows the equipment layout of a gene sequence that represents five equipment units, as an example of a layout optimization method using the GA in Chapter 7.

Through a genetic calculation, the appropriate equipment installation sequence and the appropriate directions, maintenance spaces, and installation points are decided. For example, the equipment layout for the gene sequence of the representation shown in Fig. 8.1 can be explained as follows. First, equipment 5 is installed, with an installation angle of 0° , so that the maintenance space frame A_1 (the outer shape of the equipment 5) does not extend outside the site and the center point of equipment 5 is as close as

possible to the installation point P_1 . Next, equipment 3 is installed, with an installation angle of 90° , so that the maintenance space frame A_2 does not extend outside the site, does not interfere with the maintenance space frame of equipment unit that is already installed (here, the maintenance space frame A_1 of equipment 5), and the center point of equipment 3 is as close as possible to the installation point P_2 . Then, equipment 4 is installed, with an installation angle of 270° , so that the maintenance space frame A_2 does not extend outside the site, does not interfere with the maintenance space frames of equipment units that are already installed (here, the maintenance space frames A_1 of equipment 5 and A_2 of equipment 3), and the center point of equipment 4 is as close as possible to the installation point P_3 . Then, equipment 1 is installed, with an installation angle of 0° , so that the maintenance space frame A_2 does not extend outside the site, does not interfere with the maintenance space frames of equipment units that are already installed (here, the maintenance space frames A_1 of equipment 5, A_2 of equipment 3, and A_2 of equipment 4), and the center point of equipment 1 is as close as possible to the installation point P_3 . Finally, equipment 2 is installed, with an installation angle of 180° , so that the maintenance space frame A_3 does not extend outside the site, does not interfere with the maintenance space frames of equipment units that are already installed (here, the maintenance space frames A_1 of equipment 5, A_2 of equipment 3, A_2 of equipment 4, and A_2 of equipment 1), and the center point of equipment 2 is as close as possible to the installation point P_1 . After all the equipment units are installed, they are joined by pipes, ducts, and cables based on previously defined connection relationships among the equipment units. Finally, in the previous study (see Chapter 7, Section 7.4.1.3 and Section 7.4.3), we also proposed items such as installation methods that consider equipment group layout (array arrangement, connecting arrangement, and arbitrary shape) and various types of arrangement constraints.

We understand plant layout design as a multi-objective optimization problem that also has multiple evaluation functions (hereinafter, also including arrangement constraints). Based on a satisficing trade-off method, we substitute into the minimization problem of the scalarized function shown in the following formula to determine the optimal layout solution desired by the designer. This method corresponds to a maximal component minimization method (min-max solution), and it becomes a better layout solution, the more the item that is the maximum component within the evaluation function is minimized.

$$\begin{cases} \min_{\mathbf{x}, z} z + \alpha \sum_{i=1}^r w_i [f_i(\mathbf{x}) - \bar{f}_i], \\ \text{subject to} \begin{cases} w_i [f_i(\mathbf{x}) - \bar{f}_i] \leq \beta_i z, & i = 1, 2, \dots, r, \\ \mathbf{x} \in \mathbf{X}, \end{cases} \end{cases} \quad (8.1)$$

where f_i is an evaluation function we shall describe in Section 8.5.1, \bar{f}_i is the aspiration level (designer's target value), and $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the design variable vector. \mathbf{X} indicates the feasible region of \mathbf{x} , α is a small positive value (that is, 10^{-6}), and w_i is the weight factor and is automatically determined by Eq. (8.2).

$$w_i = \frac{1}{f_i - f_i^*}, \quad (8.2)$$

where f_i^* is an ideal point and is generally taken so that $f_i^* \leq \min [f_i(\mathbf{x}) | \mathbf{x} \in \mathbf{X}]$. In Eq. (8.1), β_i expresses the relative importance of the objectives and constraints in f_i , adjusted to be within the interval $[0, 1]$. That is, if $\beta_i = 1$ then the objective functions are more important, and as we approach $\beta_i = 0$ then the constraints become more important.

8.2.2 Problems in the previous study

The optimal layout solution obtained from the layout optimized calculation using the GA described in Chapter 7 depends on the values that are allocated to the direction, maintenance space, and installation point genes. The reason is that continuous quantities such as spacing distances and installation point coordinate locations are being allocated as discrete. Moreover, as the number of maintenance space frames and installation points increase, they approach continuous quantities. The total number of combinations increases remarkably, and that worsens convergence in the search for a solution; thus, it becomes difficult to obtain an optimal layout solution within a practical calculation time. In other words, narrowing down to a discrete design space with a scale that is appropriate for searching an optimal solution is necessary from the viewpoint of computational cost. However, doing this to express layout results for an originally continuous design space will not necessarily be optimal. Still, it can be understood that, for a rough equipment layout with a layout optimized calculation using GA, an approximately satisfactory layout solution is obtained, based on which, it is

possible to correct the equipment layout.

In the present work, to reduce the effort of the designer in correcting the equipment layout, we investigate a design support method using automatic layout adjustment. The basic planning and design of the plot plan for a thermal power plant must efficiently evaluate various layout proposals in a comparatively short design time. Thus, it is extremely important to reduce the computational cost of the developed interactive design support system.

8.3 Summary of PSO

PSO is an optimization method conceived by Eberhart and Kennedy through a simulation of a simplified society model (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 2001). It is a heuristic model that imitates the swarm intelligence of living organisms (birds or fish) searching for a solution. The following are the reasons for choosing PSO for this study:

1. In a layout optimization problem, it is difficult to express the evaluation functions as functions of design variables, for reasons such as the need to exchange data with a computer-aided design (CAD) system. In addition, the evaluation functions, in general, have multiple peaks. Therefore, PSO is applicable because it does not require function sensitivity (a gradient), but rather can find a global optimal solution by searching multiple points simultaneously.
2. The above reasons are similar to the features of GAs, and therefore, there is high affinity with them, such as the ability to share the layout optimized calculation and data structures with the GA of the previous study (see Chapter 7). Further, because PSO is good at optimizing continuous variables, it has a complementary relationship with GA, which is good at optimizing discrete variables, and thus, it fits with the purpose of this study.

Recently, PSO has had refinements from many researchers. Here we will give an overview of the standard PSO and the agent-type PSO (one type of PSO refinement) that is used in this study.

8.3.1 Standard PSO

PSO has multiple search points that are called “particles,” corresponding to

individual organisms, and the search for a solution is performed as the particles move within the design space while forming swarms and sharing profitable information. The motion of each particle is found by updating its position and velocity with an iterative calculation. In the search, using the position e_{ij}^k and velocity v_{ij}^k of each particle ($j = 1, 2, \dots, N$) in each dimension ($i = 1, 2, \dots, m$) at time k , the position e_{ij}^{k+1} and velocity v_{ij}^{k+1} at time $k + 1$ are updated by using the following equations:

$$e_{ij}^{k+1} = e_{ij}^k + v_{ij}^{k+1}, \quad (8.3)$$

$$v_{ij}^{k+1} = wv_{ij}^k + c_1 r_{1ij} (p_{ij}^k - e_{ij}^k) + c_2 r_{2ij} (g_i^k - e_{ij}^k), \quad (8.4)$$

where p_{ij}^k is, for each particle j , its own best position information (personal best: *p-best*) that has been found until now, and g_i^k is the best position information (global best: *g-best*) shared by the entire particle swarm. r_{1ij} and r_{2ij} are uniformly distributed random numbers within the interval $[0, 1]$. w , c_1 , and c_2 are adjustment parameters (in particular, w is called the inertia coefficient), and they are specified as follows:

$$\begin{cases} 0 \leq w \in R^1, \\ 0 \leq c_1 \in R^1, \\ 0 \leq c_2 \in R^1. \end{cases} \quad (8.5)$$

Equation (8.4), for updating velocity, is a linear composition of three vectors, namely, the velocity of the particle at the immediately preceding time (inertia), direction toward the particle's own *p-best*, and direction toward the *g-best* that the particle shares with all the particles of the swarm.

8.3.2 Agent-type PSO

For the standard PSO of the previous section, it is widely known that the performance of the solution search greatly changes according to the settings of the three adjustment parameters w , c_1 , and c_2 . Thus, it is necessary to perform appropriate adjustment of w , c_1 , and c_2 by trial and error depending on the design problem. Based on various types of benchmark calculations, c_1 and c_2 are typically chosen so that $c_1 + c_2 = 4$ (in particular, $c_1 = c_2 = 2$ is widely used), and an adjustment method in

which the inertia coefficient w goes down from 0.9 to 0.4 with increase in the time k (the number of iterations) has been recommended. Such a method of adjusting the parameters can be viewed in the following way from Eqs. (8.3) and (8.4):

1. In the beginning of the search for a solution, by making w relatively large compared to c_1 and c_2 , the particles tend to have divergent motions, which increases the power of global search and suppresses the power of local search.
2. Conversely, near the end of the search for a solution, by making w relatively small compared to c_1 and c_2 , the particle motions tend to be focused, which increases the power of local search and suppresses the power of global search.

However, with the above method, regardless of whether the behavior of the particle swarm tends to be divergent or focused, or whether the g -best discovers a sufficiently good solution, it is necessary to first schedule the inertia coefficient w , and there are problems such as premature convergence phenomena depending on the design problem. Therefore, in the present study, to do both global and local searching, and to reduce the cost of the complicated parameter adjustment operations, we use an agent-type PSO. The following are the refinements over the standard PSO of the previous section:

1. The three adjustment parameters w , c_1 , and c_2 can be set differently for each particle.
2. As shown below, w , c_1 , and c_2 are randomly decided for each number of iterations k within ranges having upper and lower bounds that are different for each particle ($j = 1, 2, \dots, N$) in each dimension ($i = 1, 2, \dots, m$).

$$\begin{cases} w_{ij, \min} \leq w_{ij}^k \leq w_{ij, \max} , \\ c_{1ij, \min} \leq c_{1ij}^k \leq c_{1ij, \max} , \\ c_{2ij, \min} \leq c_{2ij}^k \leq c_{2ij, \max} , \end{cases} \quad (8.6)$$

$$\text{where } \begin{cases} w_{ij, \min} \in R^1 , & w_{ij, \min} < w_{ij, \max} \in R^1 , \\ c_{1ij, \min} \in R^1 , & c_{1ij, \min} < c_{1ij, \max} \in R^1 , \\ c_{2ij, \min} \in R^1 , & c_{2ij, \min} < c_{2ij, \max} \in R^1 . \end{cases} \quad (8.7)$$

The refinement described above applies to the PSO particle swarm the idea that a group of organisms with various types of individuality has more resistance to

environmental change than a uniform group of organisms. Arakawa et al. have previously performed benchmarks to verify that this kind of agent-type PSO, which allows setting the parameters for different purposes within the particle swarm, has a good ability to search for solutions (Arakawa et al., 2006).

8.4 Proposed method

8.4.1 Design variables

Possible design variables for the plant layout design are the installation coordinates and the installation angles of all the equipment units. Among these, with regard to the installation angles, in general, the only installations are a horizontal layout and a vertical layout with an angle of rotation of 90° , so we assume that this is already optimized in the layout results using the GA of the previous study (see Chapter 7). As for the installation coordinates, in this study, the objective is a plane configuration layout, so the position is on a xy -plane (two dimensions) within the site. Therefore, if m is the number of equipment units to be installed, then the dimension of the design space for this design problem is $2 \times m$.

A simple idea would be to use the installation coordinates of the origins of the equipment directly as the design variables to be optimized with PSO. Nonetheless, in previous investigations by us with this approach, it was not possible to obtain a satisfactory solution, because in a thermal power plant, it is necessary to lay out much equipment with complicated interactions within a limited site. Therefore, in the majority of the regions within the design space, problems such as overlapping of equipment and violations of arrangement constraints occur. If penalties are added to eliminate such regions from the solution search, then the solution space (the evaluation function space) becomes strongly multi-peaked, giving rise to more local solutions. Then, owing to strong focusing, which is a drawback of PSO, one becomes entangled with local solutions and extrication is impossible. Therefore, we have taken the key idea as finding an approximately optimal solution starting with a layout solution found by the layout optimized calculation using the GA of the previous study (see Chapter 7), and we proceed as follows. First, as shown in Fig. 8.2, the installation coordinates x_{di} of the origin of each equipment unit ($i = 1, 2, \dots, m$), are defined as the movement of a displacement amount u_{di} from the installation coordinates $x_{di, tbest}$ of the best solution information (total best: $tbest$) found until now, expressed by the following

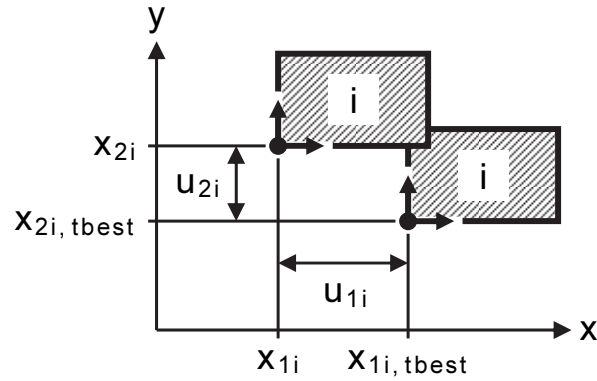


Fig. 8.2 Design variables. The installation coordinates x_{di} of the origins of the equipment are defined as the displacement amount u_{di} from the installation coordinates of the best solution information (t -best) that was found so far.

equation:

$$x_{di} = x_{di, tbest} + u_{di} , \quad (8.8)$$

where d is the coordinate axis on the xy -plane, namely $d = 1$ is the x -coordinate axis, and $d = 2$ is the y -coordinate axis. Then, by setting the displacement amounts u_{di} as the design variables to be optimized by PSO, and successively correcting the u_{di} in the PSO iterative calculation, we get an algorithm to find an optimal layout solution. Then, in Eqs. (8.3) and (8.4), the position and velocity update formulas (time $k + 1 \leftarrow k$) of each particle ($j = 1, 2, \dots, N$) can be substituted into the following equations:

$$u_{dij}^{k+1} = u_{dij}^k + v_{dij}^{k+1} , \quad (8.9)$$

$$v_{dij}^{k+1} = w_{dij}^k v_{dij}^k + c_{1dij}^k r_{1dij} (p_{dij}^k - u_{dij}^k) + \frac{1}{2} c_{2dij}^k r_{2dij} (g_{di}^k - u_{dij}^k) + \frac{1}{2} c_{2dij}^k r_{2dij} (0 - u_{dij}^k) . \quad (8.10)$$

Here, the fourth term on the right side of Eq. (8.10) expresses the direction toward the t -best, where the displacement amount would always be “0 (zero).” The g -best position g_{di}^k in the third term on the right side of Eq. (8.10) is updated with each iteration of the

calculation and, as a result, it is possible that the power of the local search is reduced and the focusing of the solution search is worsened. Therefore, by adding the direction toward *t-best*, the focusing of the solution search is improved.

8.4.2 Algorithm

The following is the algorithm of this proposed method. Hereinafter, subscripts d , i , and j are the coordinates number ($d = 1, 2$), the equipment number ($i = 1, 2, \dots, m$), and the particles number ($j = 1, 2, \dots, N$), respectively.

Step 1. Preparation: Set the number of particles $2 \leq n \in R^1$ and the maximum number of iterations k_{\max} . Select a candidate for the optimal layout solution from the layout results using the GA of the previous study (see Chapter 7), and set the installation coordinates $x_{di, GA}$ of the origin of each equipment unit i as the installation coordinates $x_{di, tbest}$ of the best solution information (*t-best*) found so far, as follows:

$$x_{di, tbest} = x_{di, GA} . \quad (8.11)$$

Next, set the evaluation function value for the *t-best* layout solution to S_{tbest} . The following evaluation function value S uses the scalarizing function value of Eq. (8.1):

$$S_{tbest} = f(x_{di, tbest}) . \quad (8.12)$$

Step 2. Initialization: Randomly generate the initial position u_{dij}^0 of each particle j within the interval $[u_{d,L}, u_{d,U}]$:

$$u_{d,L} \leq u_{dij}^0 \leq u_{d,U} , \quad (8.13)$$

and set the initial velocity v_{dij}^0 of each particle j to be 0:

$$v_{dij}^0 = 0 . \quad (8.14)$$

Set the initial position u_{dij}^0 of each particle j the initial position p_{dij}^0 of the best solution information (*p-best*) of that particle j :

$$p_{dij}^0 = u_{dij}^0, \quad (8.15)$$

and set the initial position g_{di}^0 of the best solution information (*g-best*) shared by all the particles in the swarm to 0:

$$g_{di}^0 = 0. \quad (8.16)$$

Therefore, the velocity v_{dij}^1 of each particle j at time $k = 1$ is updated as follows from Eq. (8.10):

$$v_{dij}^1 = -c_{2dij}^0 r_{2dij} u_{dij}^0. \quad (8.17)$$

Then, for time $k = 1$, go to Step 3.

Step 3. Position update: Update the position u_{dij}^k of each particle j from Eq. (8.9) by using the following equation:

$$u_{dij}^k = u_{dij}^{k-1} + v_{dij}^k. \quad (8.18)$$

However, impose the following upper and lower limits:

$$u_{d,L} \leq u_{dij}^k \leq u_{d,U}. \quad (8.19)$$

Step 4. Evaluation: Convert from the position u_{dij}^k of each particle j to the installation coordinates x_{dij}^k of the origin of each equipment unit i :

$$x_{dij}^k = x_{di,tbest} + u_{dij}^k. \quad (8.20)$$

Calculate the evaluation function value S_j^k of each particle j from the installation coordinates x_{dij}^k of the origin of each equipment unit i :

$$S_j^k = f(x_{dij}^k), \quad (8.21)$$

and then, update the *p-best* position p_{dij}^k by using the following equation:

$$P_{dij}^k = \begin{cases} u_{dij}^k, & j \in \mathbf{J}, \\ p_{dij}^{k-1}, & j \notin \mathbf{J}, \end{cases} \quad (8.22)$$

$$\text{where } \mathbf{J} = \left(j \mid S_j^k \leq S_{j, pbest}^{k-1}, \quad j = 1, 2, \dots, N \right). \quad (8.23)$$

For each equipment unit i , the installation coordinates $x_{dij, pbest}^k$ of the origin and the evaluation function value $S_{j, pbest}^k$ for the p -best position p_{dij}^k are given by the following equations:

$$x_{dij, pbest}^k = x_{di, tbest} + p_{dij}^k, \quad (8.24)$$

$$S_{j, pbest}^k = f\left(x_{dij, pbest}^k\right). \quad (8.25)$$

Next, the g -best position g_{di}^k is updated with the following equation:

$$g_{di}^k = u_{dij_g}^k, \quad (8.26)$$

$$\text{where } j_g = \arg \min_j S_j^k. \quad (8.27)$$

Here, “arg min” means the value of the variable that gives the minimum value.

For each equipment unit i , the installation coordinates $x_{di, gbest}^k$ of the origin and the evaluation function value S_{gbest}^k for the g -best position g_{di}^k are given by the following equations:

$$x_{di, gbest}^k = x_{di, tbest} + g_{di}^k, \quad (8.28)$$

$$S_{gbest}^k = f\left(x_{di, gbest}^k\right). \quad (8.29)$$

Step 5. Velocity update: Update the velocity v_{dij}^k of each particle j from Eq. (8.10) by using the following equation:

$$\begin{aligned}
v_{dij}^{k+1} &= w_{dij}^k v_{dij}^k + c_{1dij}^k r_{1dij} (p_{dij}^k - u_{dij}^k) \\
&+ \frac{1}{2} c_{2dij}^k r_{2dij} (g_{di}^k - u_{dij}^k) + \frac{1}{2} c_{2dij}^k r_{2dij} (0 - u_{dij}^k).
\end{aligned} \tag{8.30}$$

Step 6. Calculate the optimal layout solution: Save the t -best installation coordinates $x_{di, tbest}$ by using the following equation:

$$x_{di, tbest} = \begin{cases} x_{di, gbest}^k, & \text{if } S_{gbest}^k \leq S_{tbest}, \\ x_{di, tbest}, & \text{otherwise.} \end{cases} \tag{8.31}$$

If the t -best gets updated with Eq. (8.31), then both the position u_{dij}^k and the p -best position p_{dij}^k of each particle j are corrected to match the change in the t -best installation coordinates $x_{di, tbest}$:

$$u_{dij}^k = \begin{cases} u_{dij}^k - g_{di}^k, & \text{if } S_{gbest}^k \leq S_{tbest}, \\ u_{dij}^k, & \text{otherwise,} \end{cases} \tag{8.32}$$

$$p_{dij}^k = \begin{cases} p_{dij}^k - g_{di}^k, & \text{if } S_{gbest}^k \leq S_{tbest}, \\ p_{dij}^k, & \text{otherwise.} \end{cases} \tag{8.33}$$

The t -best evaluation function value S_{tbest} is expressed by the following equation:

$$S_{tbest} = f(x_{di, tbest}). \tag{8.34}$$

Step 7. Mutation: The velocity update in Eq. (8.10) adds a direction toward the t -best, for which the displacement amount is always 0, to increase the power of local search, and as a result there is the possibility of early focusing. Accordingly, in this study, as a way of preventing premature convergence phenomena we include a mutation rule in PSO. During a given number of iterations if there is no change in t -best (if the update amount is within a given range) then, with a given mutation probability, choose particles and randomly change their positions u_{dij}^k within the intervals $[u_{d,L}, u_{d,U}]$ ($d = 1, 2$; $i = 1, 2, \dots, m$; $j = 1, 2, \dots, N$). Also, set the p -best position p_{dij}^k to the position u_{dij}^k after the change. Except, save the g -best particle as an elite particle, and

remove it from the mutation targets.

Step 8. Stop decision: The decision to stop the calculation depends on the number of iterations. When the time $k = k_{\max}$ is satisfied, finalize the installation coordinates for the optimal layout solution as $x_{di, tbest}$ ($d = 1, 2; i = 1, 2, \dots, m$), and its evaluation function value as S_{tbest} , and terminate. Otherwise, return to Step 3 for the time $k = k + 1$.

According to the above algorithm, the *t-best* optimal solution information is updated by an iterative calculation; therefore, the equipment layout is gradually corrected, and the direction of the correction is transmitted to the particle swarm, and thus, this should be effective for PSO. Furthermore, in this proposed method, the search for a solution is performed while performing a motion, similar to circular motion that is always based on the *t-best*, which is an optimal layout solution candidate. Therefore, it is easy to avoid the design space regions in which problems arise, such as equipment overlap and violations of arrangement constraints. As an optimal layout solution is reached, the positions u_{dij}^k and velocities v_{dij}^k of all the particles ($j = 1, 2, \dots, N$) converge to $u_{dij}^k = 0$ and $v_{dij}^k = 0$.

8.5 Application results

We explain the use of the proposed method through the example of the general arrangement for a supercritical-pressure coal-fired thermal power plant installation (total output: 600 MW×2 generators). In this example, the number of equipment and building structures is 53 (as described in Section 8.5.2, 21 of these equipment and building structures are considered to have fixed installation locations) and, considering the main connection relationships among the equipment and building structures, the number of pipe and duct connections is 120, and the number of cable connections is 42.

8.5.1 Evaluation functions

Using the previous study (see Chapter 7, Section 7.4.4) as a reference, the evaluation functions, f_i , in Eq. (8.1) are considered for each of the seven items listed below.

① **Total cost of pipes and ducts:** express the lengths of pipes and ducts that connect equipment and building structures as Manhattan distances; for each connection,

multiply the length times the unit cost of pipe or duct; reduce the total value, which is the total cost.

② **Total cost of cables:** express the lengths of cables that connect equipment and building structures as Manhattan distances; for each connection, multiply the length times the unit cost of cable; reduce the total value, which is the total cost.

③ **Interference area of maintenance space:** within the area of the regions of the maintenance space frames that are furthest on the outside, reduce the total area of the portions that interfere with intruding equipment and building structures, and the portions that extend outside the site.

④ **Movement distance of people and vehicles:** considering the convenience of the operators and workers at the power plant, reduce the total movement distance of people and vehicles from the entrances and exits of the power plant to the equipment and building structures that they access. In this example, weights are assigned according to the hypothesized amount of movement of people and vehicles, considering the 13 most important routes, such as from the main entrance to the management building, from the management building to the central control room, and from the repair factory and warehouse to the turbine building.

⑤ **Area where equipment and building structures overlap:** within the area of the outer shape of an equipment or building structure, reduce the total area of the portions that overlap with other equipment and building structures.

⑥ **Area of the equipment and building structures outside the site:** within the area of the outer shape of an equipment or building structure, reduce the total area of the portion that extends outside the site.

⑦ **Areas that violate arrangement constraints:** reduce the total area of portions that violate the arrangement constraints, corresponding to each of the types of arrangement constraints (areas of regions that show the various types of arrangement constraints) discussed in Chapter 7, Section 7.4.3.

In this study, the β_i values in Eq. (8.1) are adjusted to the strong constraints $\beta_5 = \beta_6 = 0.1$ for items ⑤ and ⑥, to the weak constraint $\beta_7 = 0.5$ for item ⑦, and to the objective functions $\beta_i = 1.0$ ($i = 1, 2, 3, 4$) for the other items ①–④.

8.5.2 Design conditions

Figure 8.3 is a general arrangement diagram created by a skilled designer with over 20 years of experience, using a conventional design procedure (manual procedure). In

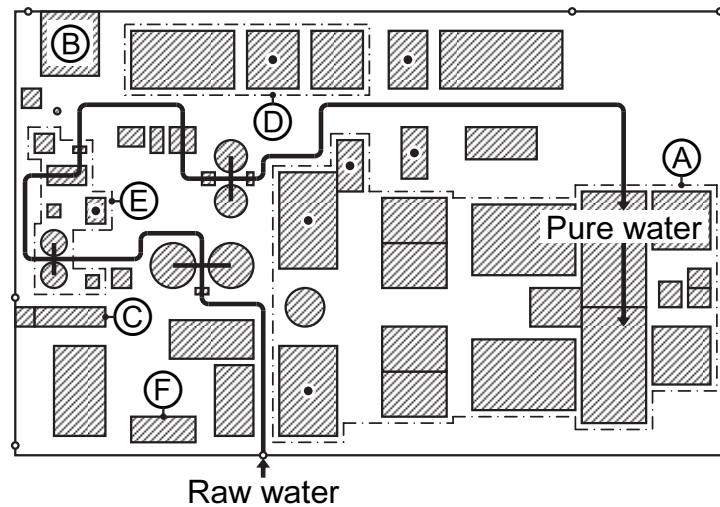


Fig. 8.3 Layout by a skilled designer. This is the site layout map created by the skilled designer with over 20 years of experience using a conventional design method (manual process). The bold line in the figure shows the piping system for the process of water quality adjustment that eliminates impurities from the obtained raw water to create pure water that is appropriate for a boiler.

Fig. 8.3, the plant facilities surrounded by the broken line A are the main facilities that constitute the boiler and turbine. For the layout of these facilities, the installation locations are determined in priority order, such as from the position of the power-sending end. In the layout optimized calculation using GA and PSO (to be described in the next section), the positions of the layout results of the skilled designer (the broken line A in Fig. 8.3) are considered fixed. The recirculating water pump equipment (the pump that sends the cooling water of the condenser, B in Fig. 8.3) and the main gate and bicycle parking area (C in Fig. 8.3) are installed in locations indicated by the customer, so those positions are considered fixed. In other words, only the layout of the other facilities is to be optimized, including ancillary facilities, such as combustion, ash-processing, wastewater processing, and pure water filtering facilities, and buildings and the vehicle garage, such as the management building and the repair factory and warehouse (F in Fig. 8.3). The following facilities use a two-step layout in which the design conditions are updated during the layout optimized calculation processes using GA and PSO, but considering the practical design sequence of the skilled designer. For the layout of the wastewater processing facilities surrounded by

the broken line D in Fig. 8.3, the installation location is adjusted after the layout of the other facilities is decided to match them. Therefore, in the layout optimized calculation using GA, they are provisionally located at the positions that are the layout results of the skilled designer (broken line D in Fig. 8.3). Then, in the layout optimized calculation using PSO, the optimal layout is found by moving the installation coordinates. Regarding the pure water filtering facilities, which are surrounded by the broken line E in Fig. 8.3, their layout details are determined after the entire installation is assured. Accordingly, in the layout optimized calculation using GA, while preserving the positional relationships of the layout results of the skilled designer (the broken line E in Fig. 8.3), the facilities are laid out together using the group layout discussed in Chapter 7, Section 7.4.1.3. In the layout optimized calculation using PSO, the optimal layout is found by dividing these facilities into individual pieces of machines and updating their various installation coordinates.

8.5.3 Optimization results

In this study, we perform automatic layout adjustment applying the PSO of the proposed method, based on the optimal layout solution obtained by the layout optimized calculation using GA of the previous study (see Chapter 7). In the following discussion, we show each of the objective function values in the layout results (Fig. 8.3) of the skilled designer as standardized to the value 100. The aspiration levels in Eq. (8.1) for each of the objective functions are set as in Table 8.1. That is, the aspiration levels for the total cost of the pipes and ducts (f_1), total cost of the cables (f_2), and interference area of the maintenance spaces (f_3) are set to values corresponding to 90, and the aspiration level for the movement distance of people and vehicles (f_4) is set to a value corresponding to 100. However, laying out all the objective functions in priority order is too strong a constraint, so we deactivate f_4 by omitting it from the evaluation functions that must be no greater than the aspiration level. The ideal point in Eq. (8.2) is set to values corresponding to 50 for all the objective functions.

First, we run the layout optimized calculation with the GA of the previous study (see Chapter 7). Here, the parameters of the GA are 30 individuals, a crossover rate of 60%, and a mutation rate of 10%, and we added external types of mutations (mutation of foreign species) every 15 generations, as described in Chapter 7, Section 7.4.5.4. We continued the solution search to the 500th generation and verified that, with the GA, we had sufficiently converged to an optimal layout solution within the solution search.

Table 8.1 Application results through six trial calculations by the proposed PSO

		f_1	f_2	f_3	f_4
Skilled designer		100.0	100.0	100.0	100.0
Aspiration level		90.0	90.0	90.0	100.0
GA		93.3	87.1	87.5	98.2
PSO	#1	87.9	84.8	87.9	97.9
	#2	87.7	84.5	87.3	98.6
	#3	87.2	84.4	87.2	98.5
	#4	87.3	84.0	87.2	98.2
	#5	87.3	84.6	87.3	98.6
	#6	87.3	84.2	87.3	99.3

Figure 8.4 shows the optimal layout solution with GA as that obtained in the 435th generation. The layout results with GA do not have any equipment or building structures extending outside the site, and all the arrangement constraints are satisfied. Moreover, in the layout optimization method with the GA of the previous study (see Chapter 7) no overlap occurs among the equipment and building structures. The bold lines in Figs. 8.3 and 8.4 show an overview of the piping system corresponding to the process that removes impurities from the raw water that is received and adjusts the water quality to pure water, which can be supplied to the boiler. This is just one example of how, in the layout optimized calculations with GA, it is possible for the pipe length to be significantly shortened and the pipe cost reduced. Table 8.1 shows the objective function values for the optimal layout solution using GA (Fig. 8.4). The layout solution using GA has all the objective function values no greater than 100, so it is better than the layout results of the skilled designer. However, f_1 is 93.3, a result that does not reach the aspiration level of 90.

Next, we run the automatic layout adjustment with the proposed PSO, starting with the layout result with the GA of Fig. 8.4. Here the PSO parameters are number of particles $N = 30$ and maximum number of iterations $k_{\max} = 1000$. The upper and lower bounds in the position update of Eq. (8.19) are $\pm 10\%$ for each of the total length (x -coordinate axis) and total width (y -coordinate axis) of the site. With regard to mutations (Step 7 in Section 8.4.2), in an interval of 10 iterations, if the update amount of the scalarization function value of Eq. (8.1) for t -best is less than 0.001, then

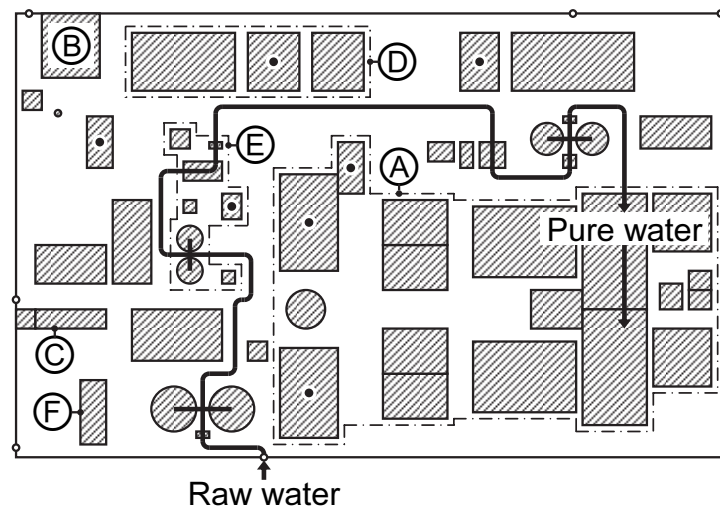


Fig. 8.4 GA layout. This figure shows the optimal layout solution using the GA in Chapter 7. The layout result using GA has no equipment and building structures extending outside the site, and satisfies all the arrangement constraints.

mutations are generated with a rate of 50%, and an interval of at least 50 iterations is left until the next mutations. To verify reproducibility of the solution search, we ran six trial calculations by changing random numbers (the initial particles were also different, depending upon random numbers). Table 8.1 shows the objective function values for the optimal layout solutions with PSO for the six trial calculations. The layout results using PSO have all the objective function values no greater than the aspiration levels, so they are better in general than the layout results using GA. We also see that the optimal layout solutions that are obtained are better balanced. The ranges of the objective function values in the six trial calculations (the difference between the largest and the smallest values) are small, where f_1 is 0.779, f_2 is 0.784, f_3 is 0.721, and f_4 is 1.409, and all the trial calculations converged in on optimal layout solutions similar to that of Fig. 8.5, to be explained next. Here we add an explanation for why the range of f_4 is slightly larger. As explained above, f_4 is omitted from the evaluation functions that must be no greater than the aspiration level. This enables that the vehicle garage (F in Fig. 8.5), which depends only on f_4 , can be freely moved in a range that does not affect the layout of the other equipment and building structures, and in a range that does not exceed the aspiration level values. Therefore, variation occurs.

Figure 8.5 shows an optimal layout solution (third trial calculation) using PSO. The

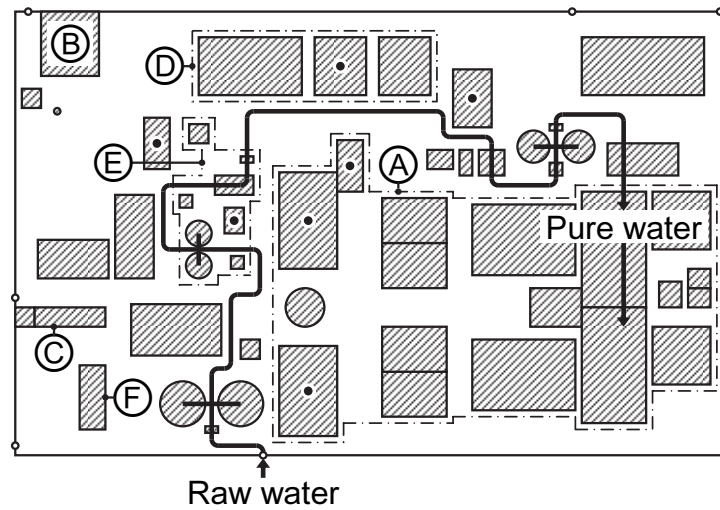


Fig. 8.5 Layout by the proposed PSO. The figure shows the optimal layout solution obtained by performing automatic layout adjustment with the PSO of this proposed method, based on the layout results using GA shown in Fig. 8.4.

layout result using PSO, similar to the layout result using the GA mentioned above, has no neither overlapping facilities, nor it has equipment or building structures extending outside the site, and satisfies all the arrangement constraints. The global solution search is performed well by the layout optimized calculation using GA mentioned above; thus, in general, the positions of the equipment and building structures are the same in Figs. 8.4 and 8.5. However, there are facilities that have been moved in a relatively large distance, such as the wastewater processing facilities (broken line D in Fig. 8.5). Here, the seven facilities marked by black dots in Figs. 8.3, 8.4, and 8.5 should be observed. These facilities are intimately related to the process of desulfurizing flue-gas from the boiler (the “desulfurization-related facilities”), and collecting them close to each other can reduce the piping cost. In the layout result of the skilled designer (Fig. 8.3) the desulfurization-related facilities (black dots in Fig. 8.3) are laid out to be collected into a comparatively small area. However, in the layout results using GA (Fig. 8.4), items such as the piping system (bold line in Fig. 8.4) for the water treatment processing from raw water to pure water, which has a relatively large effect on reducing the piping cost, have priority. Therefore the desulfurization-related facilities (black dots in Fig. 8.4) are likely to be scattered over a relatively large area. In the layout results using PSO (Fig. 8.5), while the piping lengths are kept short, such as in the piping system (bold line in

Fig. 8.5) for water treatment processing, it can be seen that the desulfurization-related facilities (black dots in Fig. 8.5) are laid out to be collected together, similarly to the layout of the skilled designer (Fig. 8.3). This is just an example of how, with automatic layout adjustment using PSO, the installation coordinates of the equipment and building structures are corrected to be more optimal, and how a total layout that is a little more compact is achieved. For the skilled designer, adjusting the arrangement by hand while understanding the connection relationships among the many pipes, ducts, and cables requires a certain skill and effort, so making use of this proposed method can lead to a layout plan that contributes to effectively reducing the construction cost.

Finally, we explain the solution search using the PSO of this proposed method. Figure 8.6 shows the calculation history (sixth trial calculation) of the scalarization function value of Eq. (8.1) for *g-best* (broken line) and *t-best* (bold line), and mutations occurred at the number of iterations indicated by the white circles on the line. For comparison, the thin line in the figure shows the calculation history for the case with the same random numbers (initial particles are also the same) but omitting the mutations. In this trial calculation, there are equipment and building structure overlaps and arrangement constraint violations in all the initial particles; thus, penalties are assigned, and *g-best* has a large initial value of 555 for the scalarization function value.

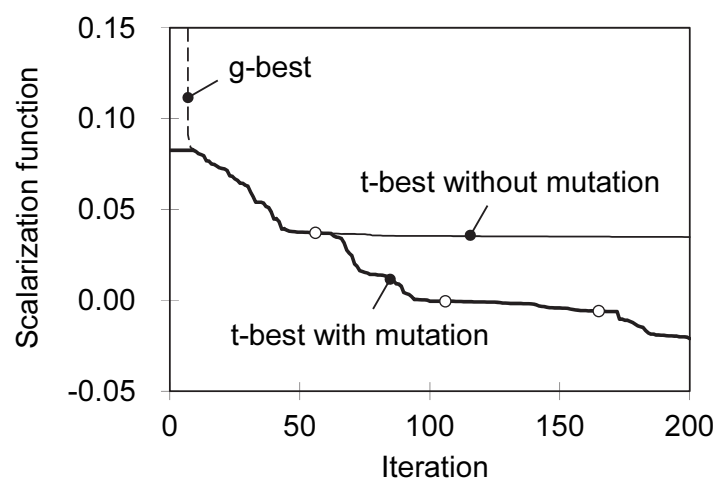


Fig. 8.6 Convergence history of the proposed PSO. This shows the calculation history of the scalarization function values of *g-best* (broken line) and *t-best* (bold line), where mutations are generated at the iteration numbers indicated by the white circles on the lines. For comparison, the thin line shows the calculation history if the mutations are removed.

Even from such inferior conditions, in the 10th iteration it is updated to 0.0825, which is the scalarization function value for *t-best* corresponding to the optimal layout solution using GA, and thereafter *g-best* and *t-best* usually move together as the solution search progresses. With no mutations (thin line), at close to 40 iterations the particle swarm has excessively concentrated around the *t-best* layout solution, so that the speed of the solution search remarkably slows, and there is premature convergence phenomena close to the 100th iteration. On the other hand, with mutations (bold line), the solution search is promoted to avoid premature convergence phenomena, so it does not fall into a local solution, and it arrives at an optimal layout solution after several rounds of mutations.

8.5.4 Discussions

In this proposed method, by combining the layout optimization methods using GA and PSO we have tried to improve the optimization of the layout solution more than in the layout optimization method of the previous study (see Chapter 7), using GA alone. However, to obtain a true global optimal solution (or a corresponding suboptimal solution) using this proposed method it is necessary for the true global optimal solution to exist near the layout results using GA. How close it needs to be probably depends on the design problem being attempted, but from the search history of Fig. 8.6 we see that the PSO of this proposed method has the ability to deal with the existence of several surrounding local solutions. This suggests that one can expand the nearby region to some extent, and we can expect that, without increasing the information allocated to the GA genes, we can obtain an optimal layout solution to satisfy the designer within a practical computation time. The calculation time per iteration, using the Toshiba notebook PC dynabook™ SS RX2 (processor operating frequency: 1.20 GHz), is approximately 1.5 s (not including the time to calculate the evaluation function values).

8.6 Conclusions

In this study, we proposed a multi-objective optimization method for a practical plant layout design. It combines layout optimized calculation using a GA and, to complement it, automatic layout adjustment using PSO. We applied this method to the general arrangement problem of an actual thermal power plant, and verified its usability.

We believe that the proposed method can be widely used in ways other than those mentioned above. For example, after choosing a general layout plan by a layout optimized calculation using GA, it is possible to determine a construction method, routes for bringing in or shipping out materials, or site routes for roads, pipe racks, or trenches, and then perform automatic layout adjustment. Alternatively, after a general layout design performed by hand by a designer, automatic layout adjustment can be performed.

The results of this study are summarized as follows:

1. We proposed a multi-objective optimization method for hybrid plant-layout designs that combines GA and PSO.
2. We proposed an automatic layout adjustment method by using a modified PSO algorithm to approach the true global optimal solution (or a corresponding suboptimal solution) near a layout solution obtained by the GA.
3. We applied this to the layout design problem of an actual supercritical-pressure coal-fired thermal power plant installation, and verified the usability of the proposed method by comparison with the layout results of a skilled designer.

References

- Arakawa, M., Kitayama, S., Yamazaki, K. and Yamakawa, H., Development of agent based PSO, Proceedings of the 4th China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems (CJK-OSM4), (2006), pp. 409–414, Kunming, China.
- Eberhart, R. C. and Kennedy, J., A new optimizer using particle swarm theory, Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS'95), (1995), pp. 39–43, Nagoya, Japan.
- Kennedy, J. and Eberhart, R. C., with Shi, Y., Swarm Intelligence, (2001), Morgan Kaufmann Publishers, Inc., San Francisco, CA, the United States.
- Nakayama, H., Aspiration level approach to interactive multi-objective programming and its applications, In: Advances in Multicriteria Analysis (Pardalos, P. M., Siskos, Y. and Zopounidis, C., eds.), (1995), pp.147–174, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Yoshikawa, T., Furuhashi, T. and Uchikawa, Y., A study on application of genetic algorithm to automatic placement of parts on printed circuit boards, IEEJ

Transactions on Industry Applications, The Institute of Electrical Engineers of Japan, Vol. 114-D, No. 4, (1994), pp. 387–392 (in Japanese).

Yoshikawa, T., Furuhashi, T. and Uchikawa, Y., Coding methods for automatic placement of parts on printed circuit boards, IEEJ Transactions on Industry Applications, The Institute of Electrical Engineers of Japan, Vol. 115-D, No. 5, (1995), pp. 642–651 (in Japanese).

Chapter 9

Interactive multi-objective optimization technique for pipe routing design

Abstract

Plot plan drawings in the process and power industries are considered key decision-making documents for projects and are normally initiated in the pre-contract, conceptual, and developmental stages of a proposal. These drawings indicate the locations of the main equipment and building structures at a plant that contain pipes and cable connections between equipment units. This study proposes a multi-objective pipe and cable routing optimization method for a thermal power plant based on effective interaction between a designer and a computer. We applied an interactive satisficing trade-off method and genetic algorithms to perform combinatorial optimization with multiple objectives. To obtain pipe and cable route candidates for multiple-criteria decision-making, we applied an ant colony optimization algorithm to a rule-based automatic layout of pipe racks. The features of this method can be optimized simultaneously not only for pipe and cable routes, but also for the placement of pipe racks among multiple conflicting evaluation functions. These functions may include the total cost of the pipes and cables, total length of the pipe racks, number of roads straddled by pipe racks, and total number of pipes and cables on a pipe rack. Finally, the validity and usefulness of this method is demonstrated by solving a layout design problem involving pipe and cable routing for an actual gas-turbine combined-cycle power plant. Furthermore, this method is premised on its incorporation into the multi-objective optimization system for plant layout design proposed in Chapters 7–8.

Keywords

Ant colony optimization, Design engineering, Genetic algorithm, Multi-objective optimization, Pipe routing, Plant layout, Power plant, Routing design, Satisficing trade-off method, Systems engineering

9.1 Introduction

The main equipment in a thermal power plant includes boilers, turbines, and generators; the auxiliary equipment includes heat exchangers, pumps, and fans; the incidental facilities are utilized for fuel, water treatment, and power transmission and transformation, and the onsite buildings include control rooms, electrical rooms, and service buildings. The equipment and buildings are connected by pipes, ducts, and cables arranged in an overarching system structure. For example, a 1000-MW-output class thermal power plant represents a complex large-scale system with a total of at least 20 km of piping and at least 3500 km of cabling. Plant layout design for pipes and cables is coordinated in conjunction with many different design departments; examples of such departments include mechanical, electrical, control and instrumentation, civil and construction departments. In addition, layout plans must be developed to allow for easy plant operation, management, maintenance, and inspection, while not wasting energy or driving up construction costs. Furthermore, the plant layout must be multi-objective based, meaning that it must simultaneously account for multiple evaluation indices, such as aesthetics and the ease of laying the foundation in addition to the construction costs and duration.

Optimization methods used for the automatic support of a variety of pipe routing designs have previously been proposed and have been intended for a wide variety of designs, such as chemical plants, power plants, waterborne ships, and aircraft. One example of such an optimization method is a technique considered for batch plant chemical processes, in which the optimum equipment installation and the shortest pipe routes are found through mixed-integer linear programming (MILP) (Guirardello and Swaney, 2005). In addition, there is a famous optimization method utilized to search for the shortest routes using Dijkstra's algorithm (Dijkstra, 1959), which is a graph theory, and several methods have been proposed to improve calculation efficiency by combining the algorithm with heuristic searching (Hart et al., 1968; Yamada and Teraoka, 1998; Wang and Liu, 2011; Liu and Wang, 2015). However, these methods are single-objective optimization methods used for determining the shortest pipe routes, and to the best of our knowledge, there are no useful examples of the multi-objective pipe routing scenario described above. In recent years, there has been active research efforts conducted on optimization methods for pipe routing problems using evolutionary computation. Examples include a proposed pipe routing search method

using a genetic algorithm (GA) (Ito, 1999; Ren et al., 2014) and a pipe routing search method using particle swarm optimization (PSO) (Liu and Wang, 2011; Dong and Lin, 2017). In these methods, a design space is divided into a lattice, and the presence of the pipe elements passing through the lattice is sought through evolutionary computation, making it possible to easily avoid obstructions or pipe interference. In addition, a method has been proposed to search for lattice routes by using ant colony optimization (ACO) in the design of engine rooms for waterborne ships and aircraft (Fan et al., 2007; Jiang et al., 2015; Qu et al., 2016). However, these methods are basically single-objective optimization methods used to search for the shortest routes while accounting for constraining conditions. The problem is that to handle large-scale design scenarios, such as that of a thermal power plant, computing time significantly increases depending upon the number of lattice divisions. In addition, although a method has been reported to optimize pipe branching point positions by applying a multi-evolutionary computation (Niu et al., 2016), it is believed that searching for a Pareto-optimal solution for all pipe routes in a plant is difficult to achieve within a reasonable amount of time. In addition, a knowledge-based expert system has been proposed for pipe routing design when designing chemical plants and ship engine rooms (Richert and Gruhn, 1999; Park and Storch, 2002). The benefit of these methods is that the computational complexity is kept relatively low, but it is believed that a significant amount of labor is required to first enter the design knowledge into a computer database to enable multi-objective evaluation. Furthermore, some commercial CAD (computer-aided design) tools have automatic support functions for pipe and cable routing design. However, these functions only aim at semi-automation based on some design rules for the pipes and cable connections between equipment units after the plant layout, including a plot plan, was determined.

Therefore, we have chosen to address the design of thermal power plant layouts (plot plans), and have developed a multi-objective optimization system for plant layout design using a satisficing trade-off method (Nakayama, 1995). This method is an interactive multi-objective programming technique that acts as the framework for the optimization effort. In Chapter 7, an interactive design support method was proposed to realize the optimal layout solution intended by a designer through effective interaction between a designer and a computer. Furthermore, a layout optimization method was proposed using a GA that was ideal for planning and designing plot plans capable of producing various layout solutions. This method featured a relatively short computing

time and simple design data. In addition, in Chapter 8, a layout optimization method was proposed that was capable of efficiently searching for a solution by combining a global search using a GA and an automatic layout adjustment using PSO. These methods are characterized as satisfying safety criteria such as safety distance and space, while achieving optimal layout solutions through by simultaneously considering multiple evaluation indices. These indices include how cost-effective the pipes, ducts, and cables are, the maintainability of the plant facilities, the ease of the construction work, and worker convenience. However, previously, because more emphasis was placed on equipment layout design than pipe routing design, the length of the pipes and cables is expressed in Manhattan distance for the sake of convenience, as described in Chapters 7–8. Actual thermal power plant pipes and cables, from the viewpoint of maintainability and aesthetics, are not connected in a scattered manner by the shortest routes. Pipes and cables are concentrated several at a time on pipe racks, and then are connected to individual facilities with which they have an input–output relationship, while avoiding equipment and building structures.

The purpose of this study is to propose an optimization method to search for an optimal route for pipes and cables based upon multi-objective evaluation, in addition to the optimal layout solutions obtained by previously studied optimization methods (see Chapters 7–8). Specifically, it proposes the following three-stage optimization method. Initially, a pipe rack is positioned based on a design rule. Next, candidate route solutions for individual pipes and cables are extracted using ACO (Dorigo and Stützle, 2004). Finally, a satisficing trade-off method and GA are used to combine all optimal routes for pipes and cables, and to solve multi-objective combinatorial optimization problems. The features of this method can be optimized simultaneously not only for pipe and cable routes, but also for the placement of pipe racks among multiple conflicting evaluation functions. Furthermore, this proposed method is premised on its incorporation into the previously studied multi-objective optimization system for plant layout design (see Chapters 7–8 for details) and thus has been devised so it may share designs and other data.

9.2 Plot plan design and multi-objective optimization system

A plot plan is a site layout drawing that shows the locations of equipment, building structures, roads, pipe racks, and the other elements that comprise a plant, while

accounting for the geographical conditions of a construction site. During thermal power plant construction projects, basic planning and design is carried out according to customer inquiries, and then a design quote is submitted as a basis for computing construction costs. Specifically, a heat balance diagram defining plant performance, a plot plan realizing plant construction, and other plans are drafted, and then construction costs are computed based upon these design drawings. Accurate and appropriate decision-making is sought when designing plot plans to minimize cost overruns due to design changes made after receiving orders. However, designs made for quotes tend to be different from detailed designs for actually constructing plant facilities, with an emphasis on short-term construction cost estimates in upstream design processes using limited design data. In this sense, the purpose of this study is not accurate pipe routing design involving the branching and joining of pipes, but support of designer decision-making based upon multi-objective evaluation to see if such pipe routing is logical.

We shall explain the multi-objective optimization system for plant layout design developed uniquely by us (see Chapters 7–8). In this system, we will use a satisficing trade-off method to solve multi-objective optimization problems by converting them to auxiliary optimization problems using the scalar function minimization in Eq. (9.1) to search for the Pareto-optimal solution intended by a designer.

$$\left\{ \begin{array}{l} \min_{\mathbf{x}, z} z + \alpha \sum_{i=1}^r w_i [f_i(\mathbf{x}) - \bar{f}_i], \\ \text{subject to } \begin{cases} w_i [f_i(\mathbf{x}) - \bar{f}_i] \leq \beta_i z, & i = 1, 2, \dots, r, \\ \mathbf{x} \in \mathbf{X}, \end{cases} \end{array} \right. \quad (9.1)$$

where f_i is an evaluation function we shall describe later, \bar{f}_i is the aspiration level (designer's target value), and $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the design variable vector. \mathbf{X} represents the feasible region of \mathbf{x} , α is the minute positive value (that is, 10^{-6}), and w_i is the weight and is automatically determined by Eq. (9.2).

$$w_i = \frac{1}{f_i - f_i^*}, \quad (9.2)$$

where f_i^* is an ideal point and is generally provided by $f_i^* \leq \min [f_i(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}]$. β_i in Eq. (9.1) is a factor that incorporates layout constraints as an evaluation function.

That is, there is a strong constraint (absolute constraint) when $\beta_i = 0$ and a weak constraint (conditional evaluation criteria) as it approaches $\beta_i = 1$.

In this study, there are five evaluation functions, f_i , accounted for in Eq. (9.1): ① the total cost of pipes, f_1 , ② the total cost of cables, f_2 , ③ the total length of pipe racks, f_3 , ④ the number of roads straddled by pipe racks, f_4 , and ⑤ the total number of pipes and cables on a pipe rack, f_5 . However, prioritizing all evaluation functions makes the constraint conditions too strong, and thus f_4 and f_5 become inactive below the aspiration level and are not available as evaluation functions. In addition, β_i in Eq. (9.1) modifies function ④ (f_4) into a weak constraint, $\beta_4 = 0.7$, function ⑤ (f_5) into a strong constraint, $\beta_5 = 0.3$, and other functions ① (f_1), ② (f_2), and ③ (f_3) into objective functions, $\beta_i = 1.0$ ($i = 1, 2, 3$). A pipe rack is a steel frame used to support multiple pipes and cables that connect equipment. Furthermore, cables run through ducts on the pipe racks in a similar manner to the pipes themselves. By bunching pipes and cables together and passing them through the pipe racks to as large a degree as possible, it is possible for them to share support members, thus enabling efficient utilization of plant space. In other words, laying pipe racks out in a logical manner contributes to a more economical arrangement through the sharing of support members, leading to improved maintainability and aesthetics through effective use of plant space. However, when f_3 is reduced, then pipes and cables tend to run about and lengthen. In addition, when pipe racks straddle roads, they need to be elevated (by constructs built high above ground) or buried underneath so they do not interfere with the movement of vehicles and heavy machinery. That is, when f_4 is reduced, workability and maintainability may be expected to improve because there are fewer elevated or subterranean workspaces. However, when f_4 was reduced, pipes and cables are locally concentrated, and f_5 increases. Furthermore, when the number of pipes and cables laid on a pipe rack considerably increase, the rack width may become too large, making it difficult to support the pipes and cables. In this scenario, the number of pipe rack levels and earthquake resistance must be increased. In other words, decreasing f_5 is beneficial for pipe rack earthquake resistance. However, when f_5 is decreased, the pipe and cable routes become more scattered and f_3 increases. Furthermore, f_1 and f_2 are strongly dependent not only on the numerical value of f_3 , but also on the layout of the pipe racks. In this manner, each evaluation function is in a complex, mutually competitive trade-off relationship.

In design procedures for plot plans in actual design settings, problems are

large-scale and the evaluation indices to consider are wide-ranging, so equipment layout design and pipe routing (hereinafter, also including cable) design are kept separated and performed hierarchically. Correspondingly, the optimization method proposed herein sequentially deals with the equipment layout optimization in Chapter 7 and the pipe routing optimization in this study. Pipe routing optimization according to this design procedure makes the output results of equipment layout optimization a precondition, and thus sometimes equipment layout must be corrected upon determining pipe routing. In such cases, it is possible to improve the overall layout from a comprehensive perspective through the automatic layout adjustment in Chapter 8. In this study, we focus on pipe routing optimization; Fig. 9.1 shows an applicable system configuration. This system follows the interactive layout solution method proposed in Chapter 7, which involves the fusion of computer intelligence and the designer's knowledge. Below, we shall explain a computational algorithm referring to Fig. 9.1.

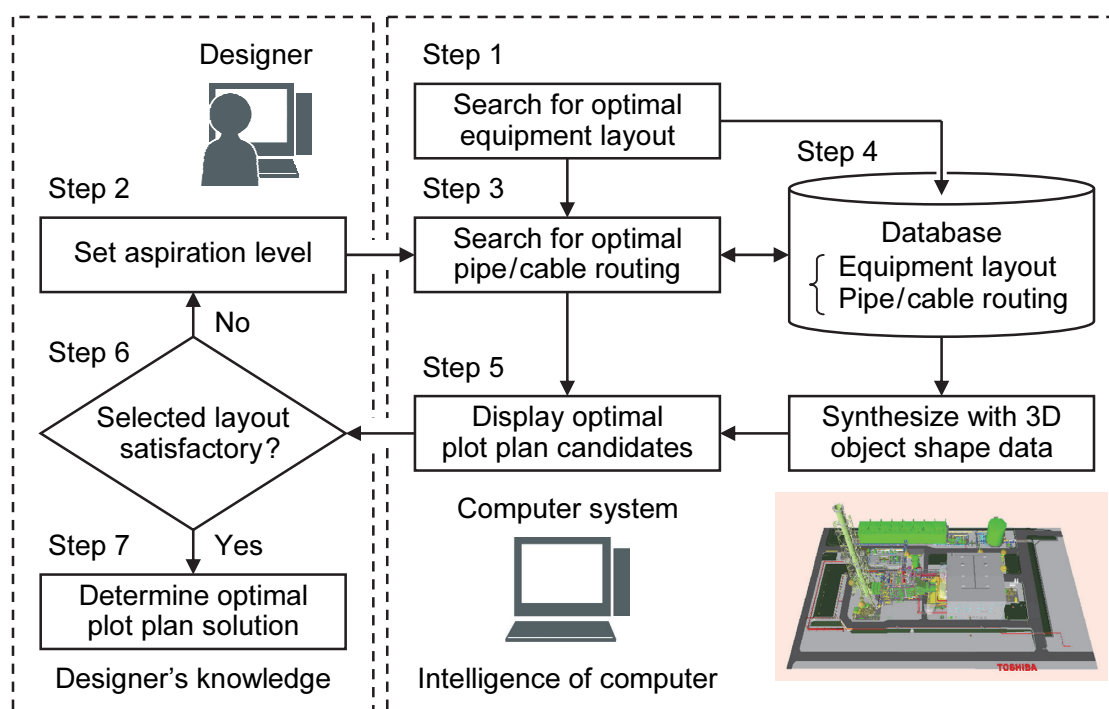


Fig. 9.1 Flowchart showing the functional structure of the proposed pipe routing design system. An optimal equipment layout (Step 1) is obtained by the multi-objective optimization system for the plant layout design in Chapter 7. This figure draws only the section of piping route optimization that focused on the entire system.

Step 1: Search for an optimal equipment layout solution based on the equipment layout optimization method introduced in Chapter 7.

Step 2: The designer sets the aspiration level values for each of the evaluation functions described above.

Step 3: Search for the optimal pipe routing solution candidates approaching each aspiration level (set in Step 2) through the pipe routing optimization method (to be described in Section 9.3) for the optimal layout solution (obtained in Step 1).

Step 4: Associate equipment layout data and pipe routing data and record in database. Furthermore, it is possible to synthesize plot plan layout data and a solid model of all facilities (equipment, building structures, and pipe racks), and then render a conceptual drawing (bird's eye view) on three-dimensional CAD. This is conducted to investigate evaluation indices for which it is difficult to formulate the aesthetics of power plants, their harmony with the surrounding environment, and other conditions.

Step 5: Extract multiple pipe routing results approaching each aspiration level and display them as optimal layout solution candidates, while referring to the database. That is, the plot plan and its evaluation function values are shown to the designer.

Step 6: The designer evaluates the multiple plot plans presented in Step 5 from a comprehensive perspective, and then selects the most ideal plot plan. Then, the designer determines whether the selected plot plan is satisfactory. If the designer is not satisfied, the process returns to Step 2, and then repeats Steps 2–6 in a loop. If the designer is satisfied, the procedure proceeds to Step 7.

Step 7: The designer makes a final decision on the plot plan selected in Step 6 for an optimal layout solution.

9.3 Proposed method

Pipe routing for thermal power plants involves countless combinations with a high degree of design freedom. Thus, it is difficult to search for a Pareto-optimal solution set that is an optimal combination, in a short amount of time, using conventional multi-objective evolutionary computation methods. Therefore, the purpose of this proposed method is to simplify pipe routing design and develop an initial temporary layout of a pipe rack based on a design rule. Next, using ACO, which is a technique for evolutionary computation, we will extract several route solution candidates for each pipe and cable connection on a temporary pipe rack layout. Finally, from amongst the

route solution candidates per each pipe and cable connection, we will search for the optimal combination based upon the multi-objective evaluation presented in Section 9.2. Subsequently, the pipe rack layout is finalized together with route solutions for all pipes and cables. By using this three-stage optimization method, it is possible to narrow the design freedom step by step and efficiently search for a solution within a short period. We shall explain the three-stage optimization method in detail below.

9.3.1 Temporary pipe rack layout according to a design rule

We held a hearing on pipe rack layout designs with skilled designers. The following points were found to be the most important takeaways from the hearing:

1. In principal, pipe racks are horizontally or vertically arranged across a plant site from the viewpoint of plant facility maintainability and workability; they are not laid out diagonally.
2. As much as possible, pipe racks are laid out along equipment from the viewpoint of pipe support being earthquake resistant.
3. As much as possible, pipe racks are laid out in places in long continuous straight lines from the viewpoint of workability and aesthetics. In doing so, places with bending pipes (elbow pipes) and pipe pressure loss is reduced, thus improving plant performance.

Therefore, in this proposed method, a temporary pipe rack layout is determined by the following rule-based procedure.

Step 1: Pipe racks for all equipment (including building structures that do not allow pipes to pass through) are thinly laid out in straight lines in four directions (front, rear, left, and right), reaching the plant site boundary or other equipment.

Step 2: Priority is given to the longest possible pipe racks in Step 1. First, pipe racks shorter than a prescribed length in the horizontal direction (x_{\min}) and vertical direction (y_{\min}) are removed. Furthermore, parallel pipe racks with spaces narrower than a prescribed width (w_{\min}) are removed. Here, longer pipe racks are left, and shorter pipe racks are removed. If pipe racks lengths are the same, then pipe racks along equipment previously defined as high in priority are left, and others are removed.

Step 3: Pipe rack connection points are determined for all equipment. First, equipment nodes are placed at equipment center points. Next, connection nodes are set at points

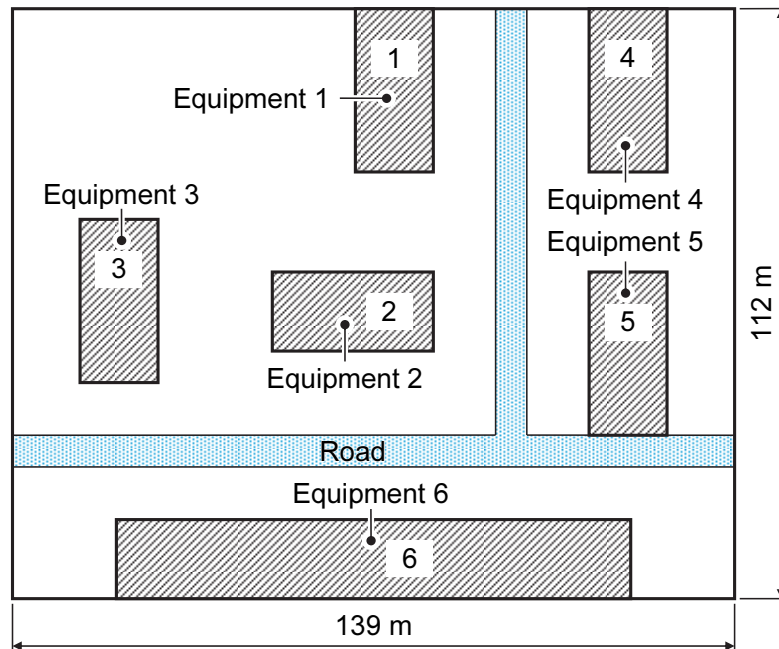


Fig. 9.2 Equipment layout of the site in the trial design problem. There are six equipment units, and the connection of pipes and cables between the six equipment units refers to Table 9.A in the Appendices. Furthermore, there is a three-way intersection where a junction between three road segments.

intersecting pipe racks within the range of a prescribed radius (r_{\min}) in four directions (front, rear, left, and right) from the equipment nodes. If equipment and pipe racks are not adjacent, then additional pipe racks are set between equipment and connection nodes.

Step 4: Finally, relay nodes are set to intersect with pipe racks in the vertical and horizontal directions. The coordinates of each node (equipment nodes, connection nodes, and relay nodes) and node numbers of the relation which each node adjoins are stored in a database as preparation for ACO computation in the next section.

Here, we will address the trial design problem for the equipment layout results shown in Fig. 9.2. There are six equipment units in this problem. The connection relationship between equipment units is shown by the line list in Table 9.A in the Appendices, wherein there are 10 pipe connections and five cable connections. We shall explain each of the steps described while referring to Fig. 9.3. First, Fig. 9.3a assumes a layout of pipe racks arranged to the front, rear, left, and right of the six equipment

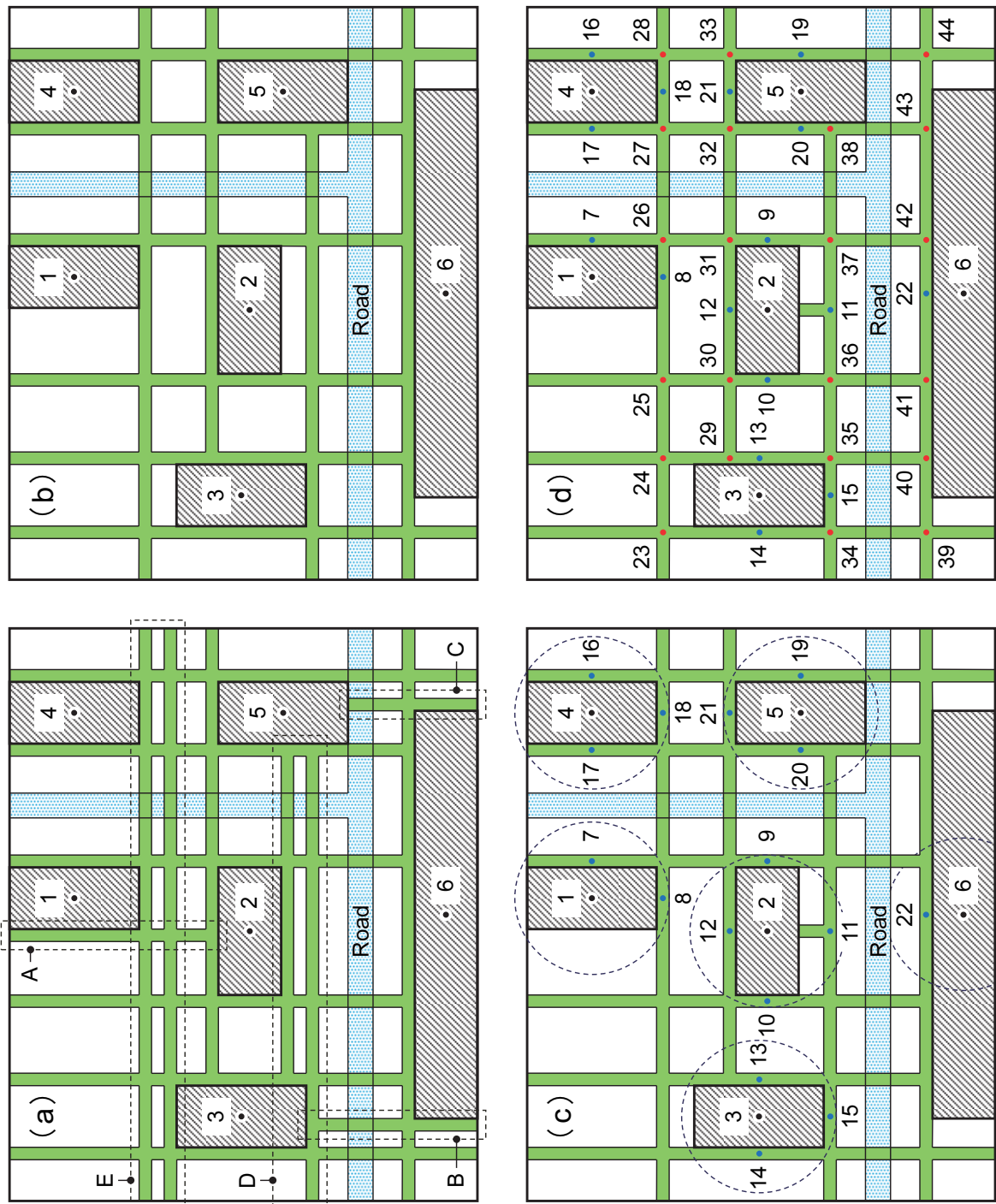


Fig. 9.3 Temporary placement of the pipe rack for the trial design problem in Fig. 9.2 based on the design rule. (a) Pipe racks are constructed at the front, rear, right, and left of each equipment unit (Step 1). (b) Pipe racks are thinned out based on a design rule (Step 2). (c) Equipment nodes (black points) and connection nodes (blue points) with equipment units are plotted (Step 3). (d) Relay nodes (red points) are plotted (Step 4).

units according to Step 1. Here, the center points of each unit of equipment are assumed to be equipment nodes 1–6 in Fig. 9.3a (black points in Fig. 9.3). Thus, the number of pipe racks is too large, so we try to reduce them according to Step 2. Furthermore, the priority of the equipment increases as the equipment node number increases. Here, we assume $x_{\min} = y_{\min} = 50$ m and $w_{\min} = 6$ m, and $r_{\min} = 18.5$ m. The short pipe racks enclosed by the A, B, and C dotted lines in Fig. 9.3a are of length y_{\min} or less, and thus are removed. The two adjacent pipe racks enclosed by dotted line D in Fig. 9.3a have spaces of w_{\min} or less, and thus the long pipe rack in the front is kept, and the short pipe rack in the rear is removed. Similarly, the two pipe racks enclosed by dotted line E in Fig. 9.3a with spaces of w_{\min} or less are of the same length, but equipment 4 is higher in priority than equipment 3, so the pipe rack to the rear connected to equipment 4 is kept, and the pipe rack to the front connected to equipment 3 is removed. By doing this, a temporary pipe rack layout with little redundancy is determined as shown in Fig. 9.3b. Next, we determine the connection points of the pipe racks to the six equipment units according to Step 3. Connection nodes 7–22 in Fig. 9.3c (blue points in Fig. 9.3) are found as intersection points within a circle of radius r_{\min} from the center points of each unit of equipment. Here, because connection node 11 is not adjacent to equipment 2, a pipe rack is added between equipment 2 and connection node 11 (see Figs. 9.3b and 9.3c). Finally, relay nodes are determined as the intersection points of the pipe racks in the horizontal and vertical directions, according to Step 4. In doing so, relay nodes 23–44 in Fig. 9.3d (red points in Fig. 9.3) are found. According to the above, in this problem, there are totally 44 node points: 6 equipment node points, 16 connection node points, and 22 relay node points.

9.3.2 Extraction of pipe routing solutions according to ACO

ACO, which was proposed in the 1990s by Dorigo et al., is an algorithm that hints that ants can find the shortest route between two points via pheromones (Dorigo et al., 1996). It has previously been applied to many combinatorial optimization problems that search for the shortest route, such as the traveling salesman problem (TSP) (Dorigo and Gambardella, 1997). In addition, many modified ACO algorithms have been proposed to address designs. The ACO algorithm used in this study references past research (Jiang et al., 2015). In other words, we apply a multi-ACO algorithm for which it is easy to find routes through which pipes may be collectively run by increasing the amount of pheromones in places where multiple pipes are passing

through, instead of searching for the shortest route per individual pipe. Jiang et al. have proposed a method in which optimal route solutions for multiple pipes are sought according to ACO using a single-objective evaluation function based on a linear weighted sum. However, the method proposed by Jiang et al. has the assignments. That is, it is necessary to adjust the weights of the evaluation functions depending on a design problem. Moreover, because ACO is prone to fall into a local solution of a large-scale problem (Dorigo and Stützle, 2004), there is a disadvantage in that the calculation amount increases. Thus, if the pipe number increases, the solution search performance will decrease. Therefore, to avoid these assignments, we use ACO to list a variety of route solution candidates, including the shortest routes. Assuming a hierarchical optimization method to search for a final optimal route solution from among these route solution candidates using a GA, decreased solution search performance owing to ACO is avoided. In addition, in the multi-objective optimization by the satisficing trade-off method explained in Section 9.2, there is no need for complicated adjustments, such as weighting between evaluation functions. Below, we shall explain the ACO algorithm used in this study.

Step 1. Initialization: Assuming generation number $t = 0$, an initial pheromone amount is provided to all routes (a, b) as in Eq. (9.3) ($\tau_0 = 100$ in this study), where route (a, b) is the route of the motion from node a to adjacent node b . $J(a)$ is a node set adjacent to node a , and N_d is the number of nodes (for example, in the trial design problem in Fig. 9.3, $N_d = 44$). Then, we set the number of individual agents (ants), N_{aco} , and the largest generation number, T ($N_{aco} = 20$ and $T = 500$ in this study), and proceed to Step 2. Although N_{aco} and T are dependent on the design problem, $(N_{aco} \times T) \geq 6000$ is recommended in our experience.

$$\tau_{a,b}(0) = \tau_0, \quad a = 1, 2, \dots, N_d, \quad b \in J(a). \quad (9.3)$$

Step 2. Solution search: The probability, $P_{a,b}^h(t)$ ($h = 1, 2, \dots, N_{aco}$), that the agents at node a will move to adjacent node b is presented as follows:

$$P_{a,b}^h(t) = \frac{\tau_{a,b}(t)}{\sum_{S \in J_h(a)} \tau_{a,S}(t)}, \quad (9.4)$$

$$a = 1, 2, \dots, N_d, \quad b \in J_h(a), \quad h = 1, 2, \dots, N_{aco}, \quad t = 1, 2, \dots, T,$$

where $J_h(a)$ represents the set of nodes through which agent h at node a can move through (nodes that have not been passed through), and $\tau_{a,b}(t)$ represents the amount of pheromones in a route (a, b) .

The number of pipe connections, N_p (hereinafter, also including cables), is provided from a line list (for example, $N_p = 15$ in Table 9.A). The N_{aco} agents ($h = 1, 2, \dots, N_{aco}$) for each pipe connection ($\ell = 1, 2, \dots, N_p$) search for routes. Agents h , according to pipe ℓ input and output information (for example, from-to- in Table 9.A), select likely connection nodes to move through with the probability in Eq. (9.4) from among the set of connection nodes with input equipment (from side). They then select relay nodes to move through with the probability in Eq. (9.4) from among the set of nodes adjacent to the selected connection nodes, and then sequentially select relay nodes to move through. If the selected relay nodes arrive at the connection nodes with output equipment (to side), then the route search is successful. However, agents remember the first node they passed through and then eliminate them from the available node set so they do not pass through it again. Through this removal process, some agents (fatal individuals) unable to arrive at the output equipment (to side) will arise, but it will be possible to remove circuitous routes unsuitable for pipe connection. Furthermore, as the generation number grows, the amount of pheromones in routes through which agents successful in route searching have passed increases, so the probability of succeeding in the route search becomes higher. If the agents are successful, then the node number line and the length of the successful route are stored in the database, and the global cumulated total is taken for the number of individual agents that reached the same successful route (number of times route reached).

Step 3. Pheromone update: The amount of pheromones is updated using Eq. (9.5). That is, the first term on the right-hand side of Eq. (9.5) imitates pheromone volatilization, and the amount of pheromones, $\tau_{a,b}(t)$, in route (a, b) for generation t are uniformly weakened by evaporative factor ρ ($\rho = 0.9$ in this study). The second term on the right-hand side of Eq. (9.5) imitates pheromone adhesion due to the passage of agents, and the more agents passing through a route, the more the amount of pheromones increases. In this manner, the amount of pheromones for generation $t + 1$ is updated by $\tau_{a,b}(t + 1)$ owing to pheromone volatilization and adhesion.

$$\tau_{a,b}(t+1) = \rho \tau_{a,b}(t) + \sum_{\ell=1}^{N_p} \left[\sum_{h=1}^{N_{aco}} \Delta \tau_{a,b}^{\ell,h}(t) \right], \quad (9.5)$$

$$a = 1, 2, \dots, N_d, \quad b \in J_h(a),$$

where $\Delta \tau_{a,b}^{\ell,h}(t)$ is the amount of pheromones newly adhering to a route (a, b) owing to agent h in pipe ℓ , which becomes larger in value as the distance across this individual circuit $C^{\ell,h}$ shortens. In this study, we compute $\Delta \tau_{a,b}^{\ell,h}(t)$ using Eq. (9.6).

$$\Delta \tau_{a,b}^{\ell,h}(t) = \begin{cases} 9 \times \left[\frac{L_{\max}^{\ell}(t) - L^{\ell,h}(t)}{L_{\max}^{\ell}(t) - L_{\min}^{\ell}(t)} \right] + 1, & \text{if } (a, b) \in C^{\ell,h}, \\ 0, & \text{otherwise,} \end{cases} \quad (9.6)$$

where $L^{\ell,h}(t)$ is the distance (pipe length) of a circuit of agent h in pipe ℓ . $L_{\max}^{\ell}(t)$ and $L_{\min}^{\ell}(t)$ are the largest and shortest successful routes among agents ($h = 1, 2, \dots, N_{aco}$) in pipe ℓ , respectively.

Below, Steps 2–3 are repeated until a final condition ($t \geq T$) is satisfied for $t \leftarrow t + 1$.

Step 4. Extraction of route solution candidates: Sort route solution candidates in order of shortest distance for each pipe connection ($\ell = 1, 2, \dots, N_p$), and extract the top n_L . Sort the remaining route solutions candidates in order of highest number of times, and then extract the top n_F . That is, a combined optimization problem for routes per $N_p(n_L + n_F)$.

9.3.3 Combinatorial optimization of pipe routing according to GA

The costs for the route solution candidates extracted by ACO in the previous section are computed by multiplying the distance and unit value (for example, unit cost in Table 9.A) for each pipe connection ($\ell = 1, 2, \dots, N_p$). Then, we sort and number them in the order of routes with the lowest cost.

The combinations of optimal route solutions across all pipes ($\ell = 1, 2, \dots, N_p$) are sought for using binary code GA. That is, in binary code, for example, route candidate solution 1 for pipe ℓ is expressed by gene 0000, route candidate solution 2 is expressed by gene 0001, and route candidate solution 8 is expressed by gene 0111, and the optimal

gene sequence across all pipes is sought. Below, we shall explain the GA for this proposed method.

Step 1. Initialization: Generate N_{ga} initial parent generation individuals ($N_{ga} = 30$ in this study). Route solutions in which f_1 and f_2 are the small costs (shortest distances) from search results according to ACO in the previous section are self-evident. Therefore, place individuals that are shortest route solutions into one initial parent generation (for this individual, the gene sequences are all zero). The remaining individuals are random gene sequences. Then, compute the fitness function value, F_j , of parent individuals using Eq. (9.7) described below.

Step 2. Crossover: Crossover pair parent individuals are selected from parent individuals by a prescribed crossover rate, P_c ($P_c = 60\%$ in this study). Subsequently, the selection pressure is proportional to the fitness, and the selection probability rises with the conforming parent individual. The crossover method randomly stipulates multiple crossover points, with multi-point crossovers exchanging gene bits at the crossover points, and two child individuals generated per crossover pair. Then, the fitness function value, F_j , of child individuals is calculated using Eq. (9.7) described below.

Step 3. Evaluation and selection: Compute the fitness function value, F_j , of each individual by scalarizing the multiple evaluation functions in Section 9.2 using Eq. (9.1). Then, as in Eq. (9.7), use a cubic function to scale them. Then, for survival of the fitness selection, combine the roulette selection using a fitness proportionate selection, and elite preservation selection leaving the best individuals to the next general, where, N_{ga} individuals (next generation parent individuals) are selected from among parent individuals and child individuals.

$$F_j = 90 \times \left(\frac{S_{\max} - S_j}{S_{\max} - S_{\min}} \right)^3 + 10, \quad (9.7)$$

where S_j is a scalarized function value of Eq. (9.1) for each individual, and S_{\max} and S_{\min} are scalarized function values for the worst individuals and best individuals, respectively. Subscript j is the number of the individual.

Step 4. Mutation: The mutation method is used to invert the gene bits of each individual by a prescribed mutation rate, P_m ($P_m = 3\%$ in this study). Furthermore, if

the scalarized function value, S_{\min} , for the best individuals has not been updated within the past 50 generations, then to prevent premature convergence phenomena, large mutations are generated at mutation rate $P_m = 50\%$. Then, calculate the fitness function value, F_j , of the mutated individuals using Eq. (9.7). Furthermore, the best individuals shall not mutate, from the viewpoint of consistency with elite preservation selection described above.

Step 5. End condition: Repeat Steps 2–5 and proceed with the next generation until the scalarized function value, S_{\min} , converges for the best individuals. Furthermore, in this study, if the scalarized function value, S_{\min} , for the best individuals has not been updated within the past 500 generations, then it is regarded as converged and the calculations have ended. Furthermore, the layout for the pipe racks is found by leaving the routes through which one or more of pipes ($\ell = 1, 2, \dots, N_p$) are passing through, and removing routes where no pipes are passing through.

9.4 Application results

9.4.1 Application results for trial design problem

The purpose of this section is to examine the performance of the proposed method. Here, we shall multi-objectively determine an optimal route solution using the trial design problem in Fig. 9.3 explained in Section 9.3.1. In the problem in Fig. 9.3, the number of equipment units is six, and from the connection relationships between equipment shown by the line list in Table 9.A in the Appendices, the number of pipe connections is 10, and the number of cable connections is five.

9.4.1.1 ACO search results

First, the ACO in the proposed method was used to search for route solution candidates for a total of 15 pipes and cables. Here, the ACO parameters are $N_{aco} = 20$ and $T = 500$, so route searching is by $20 \times 500 = 10\,000$ individuals (10 000 times) per each pipe and cable connection. Then, we extracted a total of 16 routes: eight routes in the order of shortest distance from the searched routes, and eight routes in the order of most number of individuals (number of times). Figure 9.4 shows the search results of ACO per each connection of pipes and cables. The horizontal axis of the graph in Fig. 9.4 is distance, and the vertical axis is the number of times. Data marked with ● represents 16 extracted routes, and data marked with × represents the remaining routes

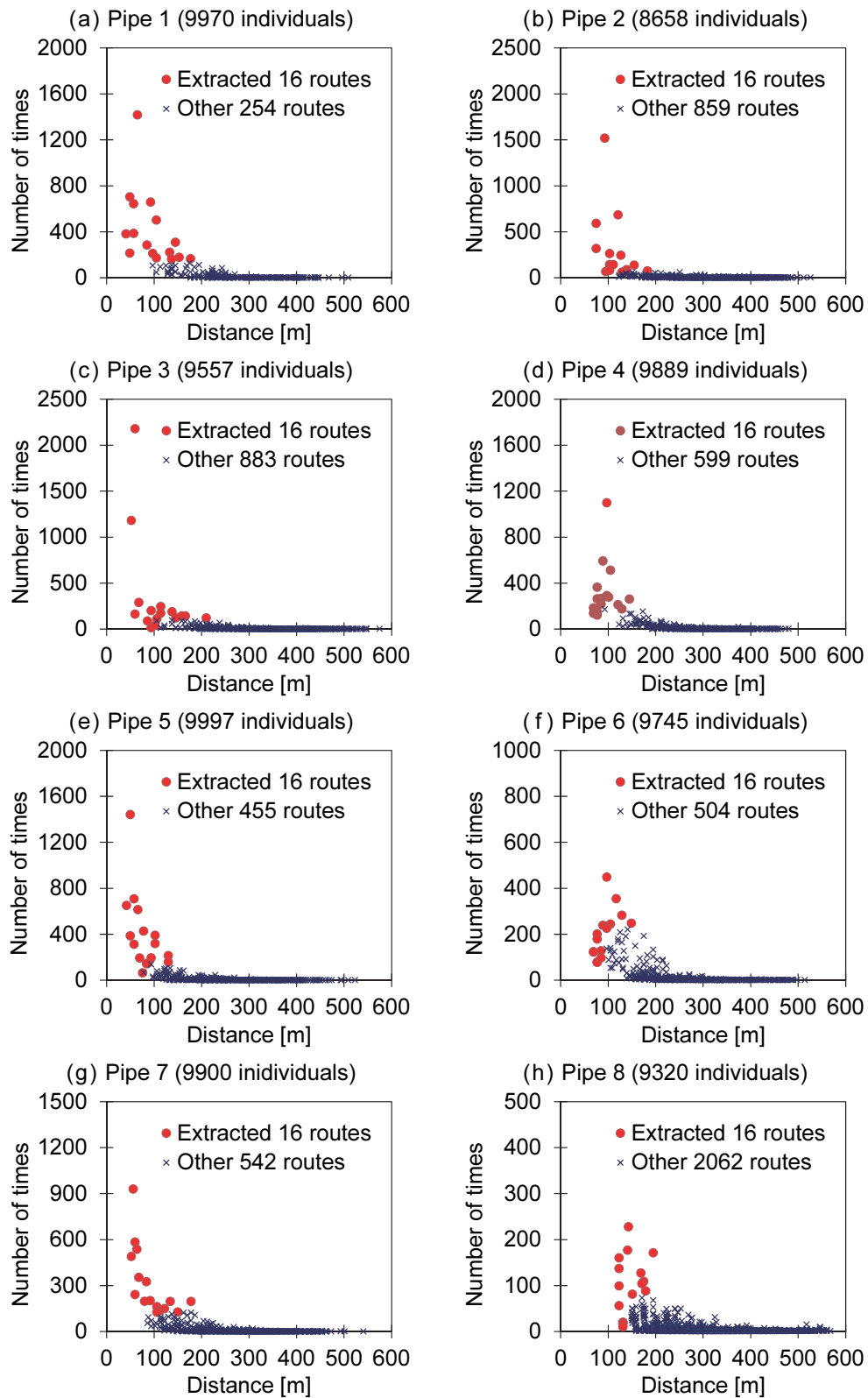


Fig. 9.4 Search results for each connection of pipes and cables by the proposed ACO (Figs. 9.4a–9.4h).

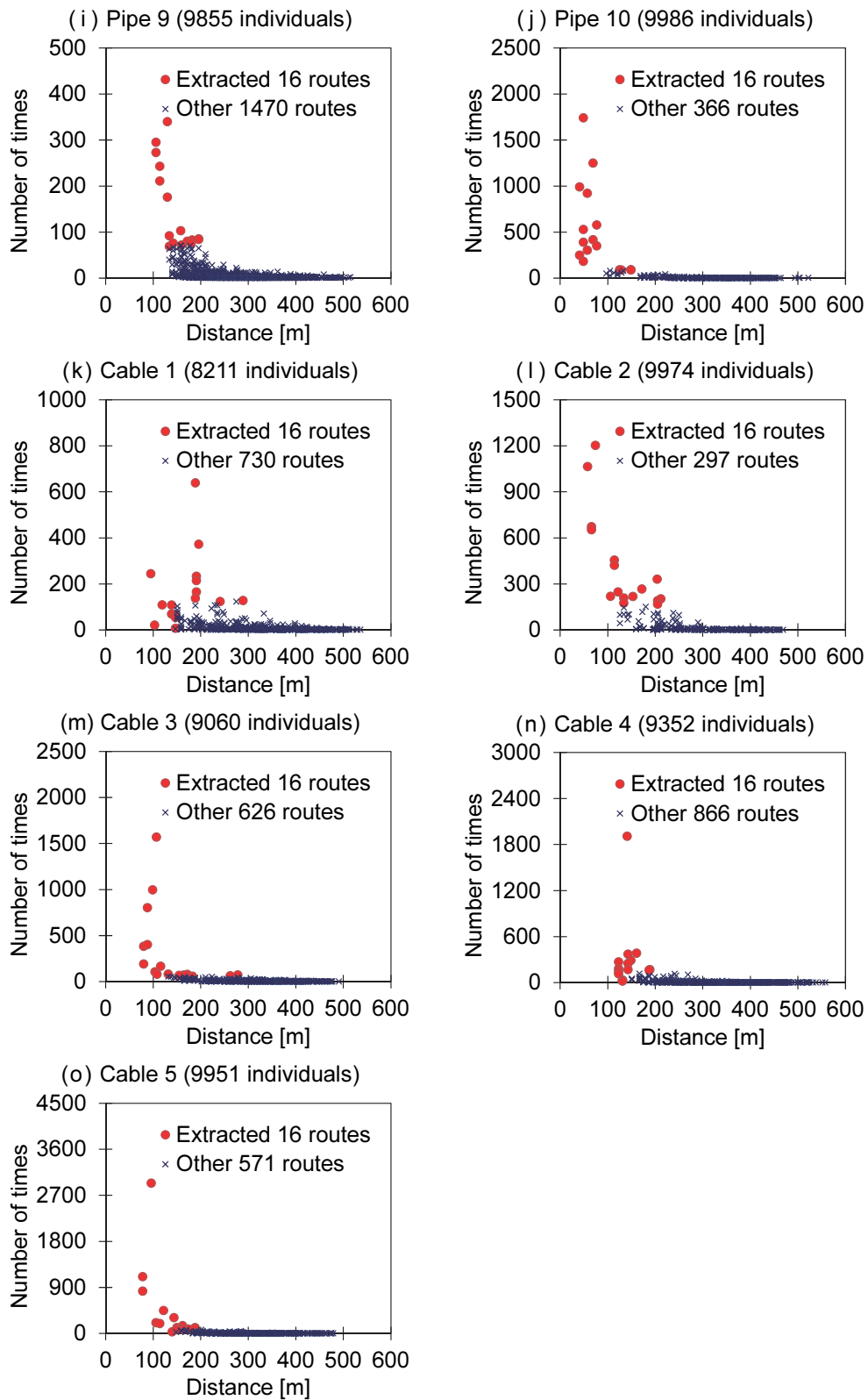


Fig. 9.4 Search results for each connection of pipes and cables by the proposed ACO (Figs. 9.4i–9.4o).

that were not extracted. For example, in pipe 2 in Fig. 9.4b, 8658 out of 10 000 individuals were successful in route searching (the remaining 1342 individuals failed in route searching); this represents a success rate of 86.58%. Thus, they present a relationship between distance and number of times for a total of 875 routes (16 extracted routes marked with ● and the remaining 859 routes marked with ×) found from a total of 8658 individuals successful in route searching. Similarly, in cable 3 in Fig. 9.4m, 9060 individuals out of 10 000 individuals are successful in route searching (the remaining 940 individuals failed in route searching), representing a success rate of 90.60%. Thus, they present a relationship between the distance and number of times for a total of 642 routes (16 extracted routes marked with ● and the remaining 626 routes marked with ×) found from a total of 9060 individuals successful in route searching. We find that the success rate for route searches by ACO is quite high. The highest success rate is 99.97% for pipe 5 in Fig. 9.4e, and the lowest is 82.11% for cable 1 in Fig. 9.4k, averaging 95.62%. This is because as ACO searching proceeds, the amount of pheromones in good routes found to be successful in searches gradually becomes higher, and the outlook for route searching becomes better. In addition, we find that a significant number of routes are found. The highest number of routes is 2078 routes for pipe 8 in Fig. 9.4h, and the lowest is 270 routes for pipe 1 in Fig. 9.4a, averaging 755 routes. Calculating the total number of combinations from these search results, there are 1.640×10^{42} routes, and despite such a small-scale routing problem, it is difficult to find the suitable route solution intended by the designer through trial and error. The 16 extracted routes in the individual pipes and cables tend to be short in distance and high in number of times. However, the shortest distance route does not necessarily indicate the highest number of times. This suggests that diverse route solution candidates are sought out by ACO in the proposed method. In addition, long-distance routes greatly detour and meander, and are unsuitable for connecting plant pipes and cables. Such routes tend to be quite small in the number of times and are found to be reasonably removed from solution candidates.

9.4.1.2 Multi-objective optimization

Next, we used the GA in the proposed method to solve combinatorial optimization problems for each of the 16 routes for a total of 15 pipes and cables. The total number of combinations from the GA according to the proposed method is $16^{15} = 1.153 \times 10^{18}$ routes. Therefore, the total number of combinations decreases from 1.640×10^{42} to

Table 9.1 Application result for the trial design problem in Fig. 9.3

		f_1	f_2	f_3	f_4	f_5
(a)	Initial route solution	\$405 400	\$155 000	566 m	6	6
(b)	Aspiration level #1	\$397 292	\$151 900	405 m	6	10
	Optimal solution #1	\$405 400	\$155 000	397 m	5	6
(c)	Aspiration level #2	\$430 644	\$182 580	235 m	3	10
	Optimal solution #2	\$422 200	\$179 000	240 m	3	10
(d)	Aspiration level #3	\$444 516	\$182 580	210 m	2	12
	Optimal solution #3	\$435 800	\$179 000	214 m	2	12

1.153×10^{18} routes according to the extraction of solution candidates using the ACO described above and decreases the calculation load for the solution search according to the GA. Five evaluation functions were explained in Section 9.2. The calculation conditions here represent three cases: a first aspiration level prioritizing f_1 and f_2 , second aspiration level prioritizing f_3 and f_4 , and third aspiration level alleviating f_5 constraints. We performed five trial calculations for each aspiration level, changing random numbers (initial individuals also different depending upon random numbers). The ideal point in Eq. (9.2) is conveniently placed at $(f_1^*, f_2^*, f_3^*, f_4^*, f_5^*) = (\$0, \$0, 0 \text{ m}, 0, 0)$. From the search results obtained using the ACO described above, the route solutions where f_1 and f_2 are smallest are easily determined by combining the routes with the shortest distances. However, when there are multiple routes with the shortest distance, we adopt routes with a higher number of times. The route solution determined in this manner is referred to as an initial route solution, and each evaluation function value is $(f_1, f_2, f_3, f_4, f_5) = (\$405\,400, \$155\,000, 566 \text{ m}, 6, 6)$ in Table 9.1. The pipe rack layout for this initial route solution is as shown in Fig. 9.5a, and Table 9.Ba in the Appendices shows each route for pipes and cables. For example, pipe 2 reaches equipment 3 from equipment 1 via relay nodes 8, 25, 24, 29, and 13. Similarly, cable 3 reaches equipment 3 from equipment 6 via relay nodes 22, 41, 40, 35, and 15.

The initial route solution is one feasible solution, but is not a Pareto-optimal solution. This is because it is possible to improve one or more evaluation functions (f_3 , f_4 , and f_5), excluding f_1 and f_2 , through a combination of other shortest distances. Therefore, the first aspiration level is set to $(\overline{f_{1,1}}, \overline{f_{2,1}}, \overline{f_{3,1}}, \overline{f_{4,1}}, \overline{f_{5,1}}) = (\$397\,292,$



Fig. 9.5 Pipe rack placement results of the trial design problem in Fig. 9.3 obtained for each aspiration level together with an initial route solution. (a) Initial route solution obtained by the proposed ACO. (b) First optimal solution. (c) Second optimal solution. (d) Third optimal solution.

(\$151 900, 405 m, 6, 10) in Table 9.1. Thus, when a solution search was made putting the initial route solution into one GA initial individual, each of the five trial calculations converged on the first optimal solution $(f_{1,1}, f_{2,1}, f_{3,1}, f_{4,1}, f_{5,1}) = (\$405\,400, \$155\,000, 397\text{ m}, 5, 6)$ in Table 9.1 by the 299th generation on average (smallest 154, largest 645). This first optimal solution pipe rack layout is as shown in Fig. 9.5b, with Table 9.Bb in the Appendices showing each route for pipes and cables. However, because several equivalent solutions with different pipe and cable routes exist, Fig. 9.5b and Table 9.Bb present only the first trial calculation result. For example, pipe 2 is changed to a route of the same distance reaching equipment 3 from equipment 1 via relay nodes 8, 25, 30, 29, and 13, so the pipe racks between relay nodes 30 and 29 are shared with other routes, and the total length of the pipe rack is shortened. Similarly, cable 3 is changed to a route of the same distance reaching equipment 3 from equipment 6 via relay nodes 22, 41, 36, 35, and 15, so the pipe racks between relay nodes 36 and 35 are shared with other routes, and the total length of the pipe rack is shortened. We find that with the first optimal solution in Table 9.1, regardless of the fact that f_1 , f_2 , and f_5 are of the same numerical value compared to the initial route solution, the numerical values of f_3 and f_4 improve, and a Pareto-optimal solution is obtained by maximally shortening the pipe rack distance with the smallest costs for pipes and cables. This suggests that in optimization methods in past studies addressing only shortest route problems, the differentiation of the initial route solution and first optimal solution is not clear, and a satisfactory route solution is not obtained. However, in the proposed method, route solution candidates are differentiated based upon multi-objective evaluation, so such problems may be overcome.

Next, it is assumed that a designer would want to further improve f_3 and f_4 . The first optimal solution is already one of the Pareto-optimal solutions, so it is not possible to improve all evaluation function values. Therefore, we compromise on f_1 and f_2 , and set the second aspiration level value to $(\overline{f_{1,2}}, \overline{f_{2,2}}, \overline{f_{3,2}}, \overline{f_{4,2}}, \overline{f_{5,2}}) = (\$430\,644, \$182\,580, 235\text{ m}, 3, 10)$ in Table 9.1. Thus, when the first optimal solution was put into one GA initial individual, each of the five trial calculations converged on the second optimal solution $(f_{1,2}, f_{2,2}, f_{3,2}, f_{4,2}, f_{5,2}) = (\$422\,200, \$179\,000, 240\text{ m}, 3, 10)$ in Table 9.1 by the 390th generation on average (smallest 36, largest 604). The second optimal solution pipe rack layout is as shown in Fig. 9.5c, with Table 9.Bc in the Appendices showing each route for pipes and cables. However, because several equivalent solutions with different pipes and cable routes

exist, Table 9.Bc presents only the first trial calculation result. For example, to shorten the total length of the pipe rack, pipe 2 is changed to a detour route reaching equipment 3 from equipment 1 via relay nodes 8, 26, 31, 12, 30, 29, and 13. Similarly, cable 3 is changed to a detour route reaching equipment 3 from equipment 6 via relay nodes 22, 42, 37, 9, 31, 12, 30, 29, and 13. These pipe racks are shared with other routes, and the total length of the pipe rack is shortened.

The relay nodes enclosed by mark \bigcirc in Figs. 9.5a, 9.5b, 9.5c, and 9.5d show the places where f_5 is largest. The initial route solution (Fig. 9.5a) has six f_5 passing through relay nodes 26. Similarly, the first optimal solution (Fig. 9.5b) has six f_5 passing through relay nodes 8, 9, and 27. However, the second optimal solution (Fig. 9.5c) has 10 f_5 passing through relay node 31, reaching the upper limit value of $\overline{f_5} = 10$ set by the aspiration level value. Therefore, we relax it to $\overline{f_5} = 12$, and set the third aspiration level value to $(\overline{f_{1,3}}, \overline{f_{2,3}}, \overline{f_{3,3}}, \overline{f_{4,3}}, \overline{f_{5,3}}) = (\$444\,516, \$182\,580, 210\text{ m}, 2, 12)$ in Table 9.1. Thus, when a solution search was made by putting the second optimal solution into one GA initial individual, each of the five trial calculations converged on the third optimal solution $(f_{1,3}, f_{2,3}, f_{3,3}, f_{4,3}, f_{5,3}) = (\$435\,800, \$179\,000, 214\text{ m}, 2, 12)$ in Table 9.1 by the 553rd generation on average (smallest 154, largest 1053). The pipe rack layout for the third optimal solution is as shown in Fig. 9.5d, and Table 9.Bd in the Appendices shows each route for the pipes and cables. This third optimal solution is thought to be a design limit Pareto-optimal solution, wherein f_3 is smallest, and there are 12 f_5 passing through relay node 31. As described above, the total number of pipes and cables on a pipe rack (f_5) becomes the maximum value on the relay node. However, the total number of parallel pipes and cables passing through the path between the relay nodes is smaller. For example, the total number of parallel pipes and cables is eight between relay nodes 31 and 32 in Fig. 9.5d.

9.4.1.3 Computing time

In this manner, we find that diverse optimal route solutions intended by a designer are obtained through interactive operations of the designer and computer according to the proposed method. Furthermore, the calculating time, using a personal computer (IntelTM CoreTM i7-3770S CPU @ 3.10 GHz), is approximately 3 minutes per 10 generations for ACO unit calculation processing and approximately 3 minutes per 10 generations for GA unit calculation processing.

9.4.2 Application results for thermal power plants

We shall explain a case applying the proposed method to the plant layout problem of a gas-turbine combined-cycle power plant (total output: 280 MW). In this case, there are 30 equipment and building structures, and the main connection relationships between the equipment and building structures are considered. There are 38 pipe connections and 28 cable connections.

9.4.2.1 Optimization result

Figure 9.6a presents an onsite layout diagram, wherein a skilled designer with 20 years or more of continuous work experience lays out equipment and building structures by a conventional design procedure and then lays out pipe racks based on the route of the pipes and cables. As in the previous section, there are five evaluation functions explained in Section 9.2. In this study, the pipe route optimized calculation is performed by the proposed method for the optimal layout solution obtained by the equipment layout optimized calculation in Chapter 7. The aspiration level value for each evaluation function is set to the same value as in the layout result of a skilled designer (Fig. 9.6a), and the ideal point in Eq. (9.2) is assumed to be a value half of that of the aspiration level value. First, we extracted a total of eight routes: four routes in order of the shortest distance per pipe and cable connection from the ACO search results, and four routes in order of the highest number of times. Then, according to GA search results, route solutions even better than those obtained by the skilled designer were obtained by the 3405th generation, and largely converged at optimal route solutions by approximately 7000 generations. Figure 9.6b is a plant layout diagram for an optimal route solution obtained by the 7691th generation of a GA.

9.4.2.2 Discussions

Table 9.2 compares the evaluation function values from the layout result of a skilled designer (Fig. 9.6a) and the layout result from the proposed method (Fig. 9.6b). However, it presents each evaluation function value normalized to 100 in the skilled-designer layout result (Fig. 9.6a). As shown in Table 9.2, the proposed method compared to a skilled designer is superior in each objective function value (f_1 , f_2 , and f_3), regardless of each constraining condition value (f_4 and f_5) being the same. The layout developed by the proposed method in Fig. 9.6b and that for equipment and building structures developed by a skilled designer in Fig. 9.6a resemble each other, but

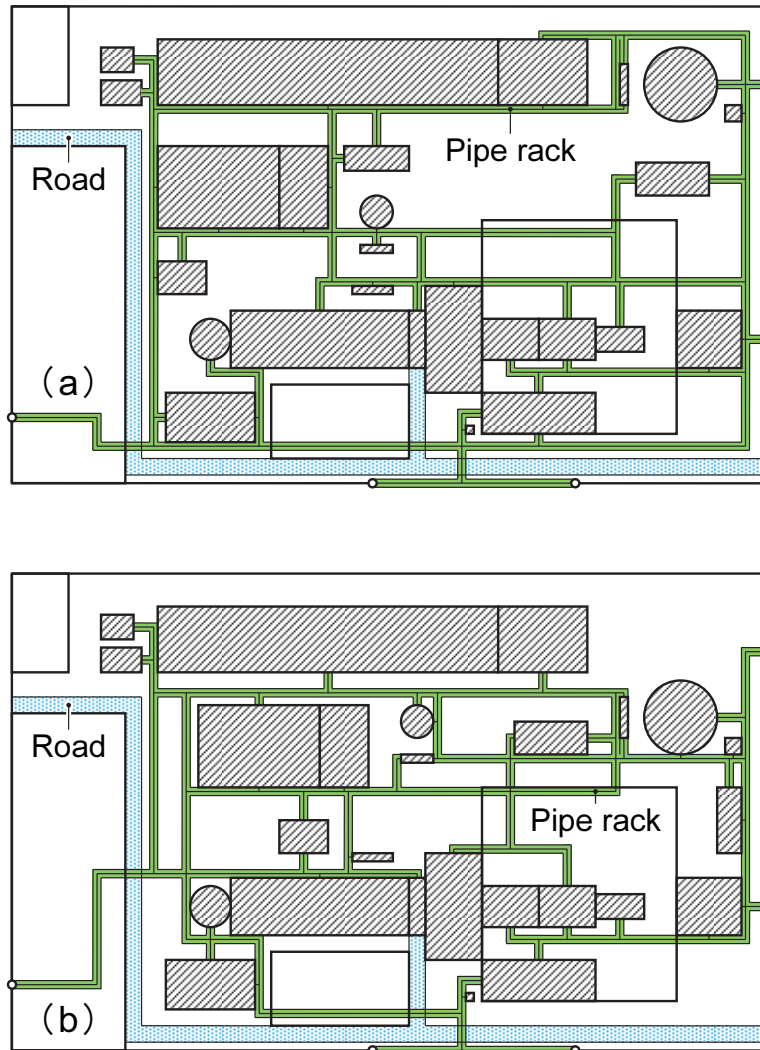


Fig. 9.6 Design pipe routing of the plot plan for a gas-turbine combined-cycle power plant. A comparison between layouts created by (a) a skilled designer and (b) the proposed method. Although white blank facilities permit the passage of a pipe rack, hatching facilities do not permit passage of a pipe rack.

the pipe rack layout is slightly different. The skilled designer shows a tendency to lay out the pipe racks to the periphery of the plant site from a viewpoint of the ease of construction work, while the proposed method shrinks the length of pipes and cables by generally laying them out between equipment on the inner side. In particular, as the pipe racks at three corner parts (the upper right, lower right, and lower left) of the plant site are removed in Fig. 9.6b, f_3 is reduced when compared with Fig. 9.6a. Moreover, as

Table 9.2 Application result for a gas-turbine combined-cycle power plant

		f_1	f_2	f_3	f_4	f_5
(a)	Skilled designer	100.0	100.0	100.0	100.0	100.0
(b)	Proposed method	89.7	71.0	92.4	100.0	100.0

the layout result from the proposed method (Fig. 9.6b) can shorten the length of the pipes and cable connections between equipment units, it can decrease f_1 and f_2 when compared with a skilled-designer layout result (Fig. 9.6a). In addition, as a piping designer takes charge of the design of a pipe rack, a skilled designer tends to give priority the placement of pipe racks to the pipe routing design and has a tendency to neglect the cable routing design. On the other hand, as the proposed method can lay out the pipe racks taking pipe and cable routing into consideration simultaneously, f_2 is relatively more improved rather than f_1 . Further, f_4 and f_5 are kept the same in both the proposed method and the layout developed by the skilled designer. The total number of parallel pipes and cables is also comparable. That is, the layout result from the proposed method has satisfied the constraining conditions. According to the skilled designer, the layout result by the proposed method may need to be adjusted somewhat accounting for problems in pipe installation and aesthetics, but the optimal route solution that can reduce the construction costs more is obtained. This suggests the usefulness of design support methods through multi-objective optimization systems for plant layout design.

9.5 Conclusions

In this study, we addressed multiple-criteria decision-making problems for pipe routing design and proposed an interactive multi-objective optimization method that uses designer and computer interactions to simultaneously consider not only pipe and cable routes but also pipe rack layouts intended by a designer. Specifically, we constructed a multi-objective optimization system for pipe routing design in which a satisficing trade-off method was used as a framework and additionally generated diverse pipe routing solution candidates through ACO. Then, we applied the method to plant layout problems in an actual gas-turbine combined-cycle power plant and

confirmed the usefulness of the proposed method.

The results of this study are summarized as follows:

1. We developed a multi-objective optimization system capable of emulating designer–computer interactive layout solution methods in Chapters 7–8 and seamless automatic support of both equipment layout design and pipe routing design in plot plans.
2. We proposed a route optimization method for generating diverse pipe routing solution candidates at high speeds by referencing past research and adopting a multi-ACO algorithm for the parallel processing of multiple pipe routing.
3. For simplifying the pipe routing design, we proposed a three-stage optimization method in which pipe racks are first temporarily laid out based on a design rule, and then pipe routing solution candidates are extracted by ACO, and then finally pipe racks are laid out while searching for combinations of optimal pipe routing solutions using a GA.
4. We verified the usefulness of the proposed method by applying it to plant layout problems in actual gas-turbine combined-cycle power plants and comparing it against the layout results of a skilled designer.

Appendices

Table 9.A Piping and cabling line list for the trial design problem

(a) Connection relation and unit cost of pipes

(a)	From	To	Unit cost [\$/m]
Pipe 1	Equipment 1	Equipment 2	800
Pipe 2	Equipment 1	Equipment 3	600
Pipe 3	Equipment 1	Equipment 4	400
Pipe 4	Equipment 1	Equipment 5	800
Pipe 5	Equipment 2	Equipment 3	600
Pipe 6	Equipment 2	Equipment 4	400
Pipe 7	Equipment 2	Equipment 5	800
Pipe 8	Equipment 3	Equipment 4	600
Pipe 9	Equipment 3	Equipment 5	400
Pipe 10	Equipment 4	Equipment 5	1000

(b) Connection relation and unit cost of cables

(b)	From	To	Unit cost [\$/m]
Cable 1	Equipment 6	Equipment 1	500
Cable 2	Equipment 6	Equipment 2	400
Cable 3	Equipment 6	Equipment 3	300
Cable 4	Equipment 6	Equipment 4	300
Cable 5	Equipment 6	Equipment 5	300

Table 9.B Pipe and cable routing results of the trial design problem in Fig. 9.3 obtained for each aspiration level together with an initial route solution

(a) Initial route solution obtained by the proposed ACO

(a)	From	Relay nodes								To
Pipe 1	Equipment 1	8	26	31	9					Equipment 2
Pipe 2	Equipment 1	8	25	24	29	13				Equipment 3
Pipe 3	Equipment 1	8	26	27	18					Equipment 4
Pipe 4	Equipment 1	8	26	27	32	21				Equipment 5
Pipe 5	Equipment 2	10	30	29	13					Equipment 3
Pipe 6	Equipment 2	9	31	26	27	18				Equipment 4
Pipe 7	Equipment 2	9	31	32	21					Equipment 5
Pipe 8	Equipment 3	13	29	30	12	31	26	27	18	Equipment 4
Pipe 9	Equipment 3	15	35	36	11	37	38	20		Equipment 5
Pipe 10	Equipment 4	18	28	33	21					Equipment 5
Cable 1	Equipment 6	22	42	37	9	31	26	8		Equipment 1
Cable 2	Equipment 6	22	42	37	9					Equipment 2
Cable 3	Equipment 6	22	41	40	35	15				Equipment 3
Cable 4	Equipment 6	22	42	43	38	20	32	27	18	Equipment 4
Cable 5	Equipment 6	22	42	43	38	20				Equipment 5

(b) First optimal solution

(b)	From	Relay nodes								To
Pipe 1	Equipment 1	8	26	31	9					Equipment 2
Pipe 2	Equipment 1	8	25	30	29	13				Equipment 3
Pipe 3	Equipment 1	8	26	27	18					Equipment 4
Pipe 4	Equipment 1	8	26	27	32	21				Equipment 5
Pipe 5	Equipment 2	10	30	29	13					Equipment 3
Pipe 6	Equipment 2	9	31	32	27	18				Equipment 4
Pipe 7	Equipment 2	9	31	32	21					Equipment 5
Pipe 8	Equipment 3	13	29	30	25	8	26	27	18	Equipment 4
Pipe 9	Equipment 3	15	35	36	11	37	38	20		Equipment 5
Pipe 10	Equipment 4	18	27	32	21					Equipment 5
Cable 1	Equipment 6	22	42	37	9	31	26	8		Equipment 1
Cable 2	Equipment 6	22	42	37	9					Equipment 2
Cable 3	Equipment 6	22	41	36	35	15				Equipment 3
Cable 4	Equipment 6	22	42	37	9	31	32	27	18	Equipment 4
Cable 5	Equipment 6	22	42	37	38	20				Equipment 5

(c) Second optimal solution

(c)	From	Relay nodes								To	
Pipe 1	Equipment 1	8	26	31	9					Equipment 2	
Pipe 2	Equipment 1	8	26	31	12	30	29	13		Equipment 3	
Pipe 3	Equipment 1	8	26	27	18					Equipment 4	
Pipe 4	Equipment 1	8	26	27	32	21				Equipment 5	
Pipe 5	Equipment 2	12	30	29	13					Equipment 3	
Pipe 6	Equipment 2	9	31	32	27	18				Equipment 4	
Pipe 7	Equipment 2	9	31	32	21					Equipment 5	
Pipe 8	Equipment 3	13	29	30	12	31	26	27	18	Equipment 4	
Pipe 9	Equipment 3	13	29	30	12	31	32	21		Equipment 5	
Pipe 10	Equipment 4	18	27	32	21					Equipment 5	
Cable 1	Equipment 6	22	42	37	9	31	26	8		Equipment 1	
Cable 2	Equipment 6	22	42	37	9					Equipment 2	
Cable 3	Equipment 6	22	42	37	9	31	12	30	29	13	Equipment 3
Cable 4	Equipment 6	22	42	37	9	31	32	27	18	Equipment 4	
Cable 5	Equipment 6	22	42	37	9	31	32	21		Equipment 5	

(d) Third optimal solution

(d)	From	Relay nodes								To	
Pipe 1	Equipment 1	8	26	31	9					Equipment 2	
Pipe 2	Equipment 1	8	26	31	12	30	29	13		Equipment 3	
Pipe 3	Equipment 1	8	26	31	32	27	18			Equipment 4	
Pipe 4	Equipment 1	8	26	31	32	21				Equipment 5	
Pipe 5	Equipment 2	12	30	29	13					Equipment 3	
Pipe 6	Equipment 2	9	31	32	27	18				Equipment 4	
Pipe 7	Equipment 2	9	31	32	21					Equipment 5	
Pipe 8	Equipment 3	13	29	30	12	31	32	27	18	Equipment 4	
Pipe 9	Equipment 3	13	29	30	12	31	32	21		Equipment 5	
Pipe 10	Equipment 4	18	27	32	21					Equipment 5	
Cable 1	Equipment 6	22	42	37	9	31	26	8		Equipment 1	
Cable 2	Equipment 6	22	42	37	9					Equipment 2	
Cable 3	Equipment 6	22	42	37	9	31	12	30	29	13	Equipment 3
Cable 4	Equipment 6	22	42	37	9	31	32	27	18	Equipment 4	
Cable 5	Equipment 6	22	42	37	9	31	32	21		Equipment 5	

References

- Dijkstra, E. W., A note on two problems in connexion with graphs, *Numerische Mathematik*, Vol. 1, Issue 1, (1959), pp. 269–271.
- Dong, Z.-R. and Lin, Y., A particle swarm optimization based approach for ship pipe route design, *International Shipbuilding Progress*, Vol. 63, Issues 1–2, (2017), pp. 59–84.
- Dorigo, M., Maniezzo, V. and Colorni, A., Ant system: Optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, Vol. 26, No. 1, (1996), pp. 29–41.
- Dorigo, M. and Gambardella, L. M., Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, (1997), pp. 53–66.
- Dorigo, M. and Stützle, T., *Ant Colony Optimization*, (2004), A Bradford Book, The MIT Press, Cambridge, MA, the United States.

- Fan, X., Lin, Y. and Ji, Z., Ship pipe routing design using the ACO with iterative pheromone updating, *Journal of Ship Production*, Vol. 23, Issue 1, (2007), pp. 36–45.
- Guirardello, R. and Swaney, R. E., Optimization of process plant layout with pipe routing, *Computers and Chemical Engineering*, Vol. 30, Issue 1, (2005), pp. 99–114.
- Hart, P. E., Nilsson, N. J. and Raphael, B., A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, (1968), pp. 100–107.
- Ito, T., A genetic algorithm approach to piping route path planning, *Journal of Intelligent Manufacturing*, Vol. 10, Issue 1, (1999), pp. 103–114.
- Jiang, W.-Y., Lin, Y., Chen, M. and Yu, Y.-Y., A co-evolutionary improved multi-ant colony optimization for ship multiple and branch pipe route design, *Ocean Engineering*, Vol. 102, (2015), pp. 63–70.
- Liu, Q. and Wang, C., A discrete particle swarm optimization algorithm for rectilinear branch pipe routing, *Assembly Automation*, Vol. 31, Issue 4, (2011), pp. 363–368.
- Liu, Q. and Wang, C., A graph-based pipe routing algorithm in aero-engine rotational space, *Journal of Intelligent Manufacturing*, Vol. 26, Issue 6, (2015), pp. 1077–1083.
- Nakayama, H., Aspiration level approach to interactive multi-objective programming and its applications, In: *Advances in Multicriteria Analysis* (Pardalos, P. M., Siskos, Y. and Zopounidis, C., eds.), (1995), pp. 147–174, Kluwer Academic Publishers, Dordrecht, the Netherlands.
- Niu, W., Sui, H., Niu, Y., Cai, K. and Gao, W., Ship pipe routing design using NSGA-II and coevolutionary algorithm, *Mathematical Problems in Engineering*, Vol. 2016, Article number 7912863, (2016), pp. 1–21, Hindawi Publishing Corporation.
- Park, J.-H. and Storch, R. L., Pipe-routing algorithm development: Case study of a ship engine room design, *Expert Systems with Applications*, Vol. 23, Issue 3, (2002), pp. 299–309.
- Qu, Y., Jiang, D., Gao, G. and Huo, Y., Pipe routing approach for aircraft engines based on ant colony optimization, *Journal of Aerospace Engineering*, Vol. 29, Issue 3, (2016), Article number 04015057.
- Ren, T., Zhu, Z.-L., Dimirovski, G. M., Gao, Z.-H., Sun, X.-H. and Yu, H., A new pipe routing method for aero-engines based on genetic algorithm, *Proceedings of the*

- Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, Vol. 228, No. 3, (2014), pp. 424–434.
- Richert, H. and Gruhn, G., A numeric-heuristic system for plant wide pipe routing, Computers and Chemical Engineering, Vol. 23, Supplement 1, (1999), pp. S735–S738.
- Wang, C. and Liu, Q., Projection and geodesic-based pipe routing algorithm, IEEE Transactions on Automation Science and Engineering, Vol. 8, No. 3, (2011), pp. 641–645.
- Yamada, Y. and Teraoka, Y., An optimal design of piping route in a CAD system for power plant, Computers and Mathematics with Applications, Vol. 35, No. 6, (1998), pp. 137–149.

Summary and conclusions

Chapter 10

Conclusions and future directions

10.1 Conclusions

In this thesis, we described an interactive multi-objective optimization system that supports multiple-criteria decision-making for the system design of a power plant through effective interaction between a designer and a computer, especially in plant operation design and plant layout design.

In the first part of the thesis, we discussed the dynamic simulation and controller simulation–testing environment utilized for plant operation design.

In Chapter 2, we derived a mathematical model (dynamic simulation model) that represents the system dynamics of the power plant. In addition, the mathematical formulas of various evaluation functions in plant operation optimization were explained.

In Chapter 3, we presented a new dynamic simulation tool that facilitates plant and control system design of thermal power plants. The issues identified in past research that were addressed by the proposed method are as follows:

1. The author developed a dynamic simulation tool based on object orientation, which was utilized to construct models for various power plant systems without modifying the program. Thus, the tool enables the design engineer to construct a simulation model, perform simulation during the design process, test an actual plant control system, and adjust the control parameters.
2. A hardware-less simulation system was developed using virtualization technology. This system can be employed to carry out design and testing of control software, which is independent of control system hardware. Thus, it is possible to carry out software development in parallel with hardware development, thereby decreasing the lead-time of the entire control system.

In the second part of the thesis, we described the successful application of the interactive multi-objective optimization method to the operational design problem of

the power plant.

In Chapter 4, we proposed an intelligent operation optimization system for a thermal power plant through effective interaction between the designer and the computer. The issues identified in past research that were addressed by the proposed system are as follows:

1. The proposed system can optimize thermal power plant operations based on multiple evaluation criteria within a reasonable computing time and with simple design data.
2. The proposed system can search for flexible start-up schedules through human–computer interactions based on varying daily requirements, that is, the price of electricity as well as fuel, maintenance, and environmental costs.

In Chapter 5, we proposed an intelligent multi-objective model predictive control method through human–computer interactions. The issues identified in past research that were addressed by the proposed method are as follows:

1. The proposed method can be applied to real-time (online) optimal control problems in a dynamic environment with multiple objectives. Subsequently, the method does not require the generation of a Pareto frontier or priorities and weight adjustments for multiple objectives, unlike in existing studies.
2. The proposed method enables flexible real-time operation to respond to unexpected market changes during plant operations, unlike in conventional schedule operation methods in which operations are preplanned.

In Chapter 6, we proposed a multi-objective evaluation method for plant operation using self-organizing maps and data envelopment analysis. The proposed method can be utilized by operators (or designers) of plants for the following real-world applications based on Pareto-optimal solutions obtained from various multi-objective optimization methods:

1. Evaluating the relative positioning of obtained solutions. For example, knowing where to place the weight within evaluation functions clarifies the direction that the user should consider during multiple-criteria decision-making based on tradeoff analytics.
2. Evaluating the relationship between the obtained solution and past operational records. For example, the user can decide not to adopt solutions that are beyond the parameters of the previous operational records. Further, the user may consider errors in modeling between dynamic simulation and the actual plant, or even expect that

there may be some unexpected changes and opt to select a safer operating parameter than the suggested optimal solution.

In the third part of the thesis, we described the successful application of the interactive multi-objective optimization method to the layout design problem of the power plant.

In Chapter 7, we proposed a multi-objective optimization system for plant layout design through effective interaction between the designer and the computer. The issues identified in past research that were addressed by the proposed system are as follows:

1. Unlike expert systems, a general-purpose multi-objective layout optimization system that does not require pre-registration of many layout rules or design knowledge in a computer was developed.
2. The system can search for a quasi-optimal layout solution in a short time and with simple design data; thus, it can handle a large-scale layout design problem such as a thermal power plant.

In Chapter 8, we proposed a hybrid optimization method for plant layout design using a genetic algorithm (GA) and particle swarm optimization (PSO). The issues identified in past research that were addressed by the proposed method are as follows:

1. The flexible multi-objective layout optimization method utilizes rough layout solutions with GA and PSO to perform layout adjustment automatically and achieve a global optimal solution (or a corresponding suboptimal solution).
2. It is not necessary to increase the genetic information of the GA to increase the precision of the layout solution; therefore, it is possible to avoid worsening the solution search capability. Human (designer) intervention is not required for correction of layout errors, such as overlapping machines.

In Chapter 9, we proposed a multi-objective optimization system for pipe and cable routing design through effective interaction between the designer and the computer. The issues identified in past research that were addressed by the proposed system are as follows:

1. The proposed system is capable of simultaneous multi-objective optimization of pipe and cable routes, along with pipe rack layouts that account for multiple evaluation indices. Subsequently, the need to adjust the weights between evaluation functions (order of priority) is eliminated, unlike in existing studies.
2. The proposed system can search for diverse pipe routing solutions within a short

time and with simple design data, thereby facilitating its application to upstream design processes. These processes include planning designs such as plot plans for various plants, and conceptual designs in which several design conditions can be implemented.

10.2 Future directions

In this section, the direction for future research is described based on results presented in this thesis. The engineering design of large-scale systems such as power plants has higher uncertainty of design data at the upstream design processes and higher risk of design changes at the downstream design processes. That is, it is important to support a designer's decision-making considering the uncertainty of design data and the risk of design changes according to the design processes. One possible direction for future research is an interactive multi-objective optimization method with high robustness. An interactive multi-objective optimization method that can extract the value judgement of a designer is also suitable within the specific conditions of the research.

Finally, the future prospects of the research are described in this section. Many industrial products can be considered a multi-domain system, which combines not only mechanical engineering domain but also many engineering domains, such as electrical, chemical, biological, financial, control, information, and electronic communication. Moreover, the scale, sophistication, and functionality of systems are increasing rapidly toward the realization of a "super-smart society" in the near future, supported by the internet of things (IoT) and artificial intelligence (AI), which deal with big data. Although the system design of a power plant was used as an example in this study, the proposed method, which has adaptability and flexibility for rational decision-making through human-computer interactions, can be applied to other complicated systems to mitigate project risks such as cost overruns, reworking of the system design, and time-schedule delay. Particularly in the upstream design phase, the proposed method can improve the applicability of system design by clarifying the relationships between design parameters and multiple conflicting evaluation indices and improving the level of cooperation among project team members.

List of publications by the author

Publications included as part of this thesis

Journal papers

Located in Chapter 2 of Part I,

Masakazu Shirakawa, Masashi Nakamoto, and Shunji Hosaka, Dynamic simulation and optimization of start-up processes in combined cycle power plants, JSME (The Japan Society of Mechanical Engineers) International Journal, Series B, Fluids and Thermal Engineering, Vol. 48, No. 1, (2005), pp. 122–128.

Located in Chapter 3 of Part I,

Masakazu Shirakawa, Development of a thermal power plant simulation tool based on object orientation, Proceedings of the Institution of Mechanical Engineers (IMEchE), Part A: Journal of Power and Energy, Vol. 220, No. 6, (2006), pp. 569–579.

Located in Chapter 4 (excluding Section 4.8) of Part II,

Masakazu Shirakawa, Masao Arakawa, and Hirotaka Nakayama, Intelligent start-up schedule optimization system for a thermal power plant, Journal of Advanced Mechanical Design, Systems, and Manufacturing, JSME Technical Journal, Vol. 1, No. 5, (2007), pp. 690–705.

Located in Chapter 4, Section 4.8 of Part II,

Masakazu Shirakawa, Multi-objective optimization system for a thermal power plant operation, Transactions of the Japan Society of Mechanical Engineers, Series B, Vol. 75, No. 751, (2009), pp. 471–473 (in Japanese).

Located in Chapter 7 of Part III,

Masakazu Shirakawa, Masao Arakawa, and Hirotaka Nakayama, Multi-objective optimization system for plant layout design (1st report, Genetic algorithm approach for designing layout of plot plan for power plants), Transactions of the Japan Society of Mechanical Engineers, Series C, Vol. 76, No. 770, (2010), pp. 2694–2703 (in Japanese).

Located in Chapter 8 of Part III,

Masakazu Shirakawa and Masao Arakawa, Multi-objective optimization system for plant layout design (2nd report, Application of particle swarm optimization to

automatic placement and adjustment of plant systems), Transactions of the Japan Society of Mechanical Engineers, Series C, Vol. 77, No. 776, (2011), pp. 1609–1620 (in Japanese).

Located in Chapter 5 of Part II,

Masakazu Shirakawa, Yeboon Yun, and Masao Arakawa, Intelligent multi-objective model predictive control applied to steam turbine start-up, Journal of Advanced Mechanical Design, Systems, and Manufacturing, Bulletin of the JSME, Vol. 12, No. 1, (2018), DOI: 10.1299/jamdsm.2018jamdsm0007.

Located in Chapter 9 of Part III,

Masakazu Shirakawa and Masao Arakawa, Multi-objective optimization system for plant layout design (3rd report, Interactive multi-objective optimization technique for pipe routing design), Journal of Advanced Mechanical Design, Systems, and Manufacturing, Bulletin of the JSME, Vol. 12, No. 2, (2018), DOI: 10.1299/jamdsm.2018jamdsm0053.

Book chapters

Hiroataka Nakayama, Yeboon Yun, and Masakazu Shirakawa, Multi-objective model predictive control, In: Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems (Ehrgott, M., Naujoks, B., Stewart, T. J. and Wallenius, J., eds.), (2010), pp. 277–287, Springer-Verlag, Berlin, Heidelberg, Germany.

Hiroataka Nakayama, Yeboon Yun, and Masakazu Shirakawa, Multi-objective model predictive control using computational intelligence, In: Computational Intelligence in Expensive Optimization Problems (Tenne, Y. and Goh, C.-K., eds.), (2010), pp. 249–264, Springer-Verlag, Berlin, Heidelberg, Germany.

Expository articles

Masakazu Shirakawa, Power plant systems design using an interactive multi-objective programming technique, Journal of the Institute of Systems, Control and Information Engineers, Systems, Control and Information, Vol. 55, No. 9, (September 2011), pp. 382–388 (in Japanese).

Masakazu Shirakawa, An interactive multi-objective optimization technique for layout

design, *Journal of the Japan Society of Mechanical Engineers*, Vol. 117, No. 1146, (May 2014), pp. 314–317 (in Japanese).

Peer-reviewed international conference papers

Masakazu Shirakawa and Masashi Nakamoto, Start-up schedule optimizing system of a combined cycle power plant, IFAC (International Federation of Automatic Control) Proceedings Volumes, The 5th IFAC Symposium on Power Plants and Power Systems Control 2003, Vol. 36, Issue 20, (September 2003), pp. 261–266, Seoul, the Republic of Korea.

Masakazu Shirakawa, Kensuke Kawai, Masao Arakawa, and Hirotaka Nakayama, Intelligent start-up schedule optimization system for a thermal power plant, IFAC Proceedings Volumes, The 6th IFAC Symposium on Power Plants and Power Systems Control 2006, Vol. 39, Issue 7, (June 2006), pp. 189–194, Kananaskis, Alberta, Canada.

Located in Chapter 6 of Part II,

Masakazu Shirakawa and Masao Arakawa, Multi-objective evaluation method for combined cycle power plant start-up, IFAC-PapersOnLine, Proceedings of the 10th IFAC Symposium on Control of Power and Energy Systems 2018 (CPES 2018), Vol. 51, Issue 28, (September 2018), pp. 19–24, Tokyo, Japan.

Awards related to this thesis

Masakazu Shirakawa, Development of a PC-based thermal power plant simulator (in Japanese), Certificate of Merit for Outstanding Presentation, Power and Energy Systems Division, The Japan Society of Mechanical Engineers, (October 2006).

Masakazu Shirakawa, Multi-objective optimization for a power plant operation (in Japanese), Certificate of Merit for Best Presentation, Design and Systems Division, The Japan Society of Mechanical Engineers, (November 2007).

Masakazu Shirakawa, Intelligent start-up schedule optimization system for a thermal power plant, Manufacturing Systems Academic Achievement Award, Manufacturing Systems Division, The Japan Society of Mechanical Engineers, (March 2010).

Trademarks

Virtual Design and Virtual Design System are registered trademarks of Toshiba Corporation.

MATLAB, Simulink, and Simscape are registered trademarks of The MathWorks, Inc. All other brand and product names appearing in this thesis are registered trademarks or trademarks of their respective holders.