Proposal of Superior Solution Set Search Problem and Construction of Superior Solution Set Search Methods Based on Metaheuristics

Hongran Wang

Tokyo Metropolitan University

A dissertation submitted for the degree of
Doctor of Philosophy

March, 2021

Acknowledgments

As a student of Tokyo Metropolitan University, after graduating from my master's degree, I chose to stay at the graduate school to continue my studies, and I chose to directly pursue a Ph.D. path to realize my ideals and pursuits. I have spent the best 5 years of my life at Tokyo Metropolitan University, and I have also shown my youth and enthusiasm in these 5 years. My essay was devoted to a lot of hard work from my mentor Professor Keiichiro Yasuda. From the topic selection of the thesis, the implementation of the thesis to the finalization of the final paper, Mr. Yasuda gave me careful guidance and support in all aspects.

Over the years, Mr. Yasuda has devoted himself to academic guidance, strict requirements in work, and meticulous care in life, providing me with a good and relaxed learning and working environment. As far as I am concerned, Mr. Yasuda is not only an academic tutor, but also an elder in life. He uses the unique delicacy of Japanese teachers to guide me in all aspects. Mr. Yasuda' s extensive professional knowledge, rigorous academic attitude, selfless work spirit, and honest demeanor have benefited me a lot from my Ph.D. career, and these wealth will surely benefit me throughout my life. Here, I would like to extend my highest respect and most heartfelt thanks to my mentor!

During my Ph.D. study, Assistant Professor Kenichi Tamura put forward many valuable suggestions on my project research plan, the specific implementation of the research work, the difficulties encountered in the experiment process, and the writing of the thesis, which laid the foundation for the smooth development and completion of my thesis. With a good foundation, his focus on scientific research and his unique insights into paper writing have benefited me a lot. Hereby express my heartfelt thanks! The successful completion of this thesis is also inseparable from the great help and support of the reviewers Professor Tochikubo, Associate Professor Souma, and Professor Koakutsu.

Most of the work in this thesis was completed in our laboratory. During the experiment, we received great help and support from classmate in our laboratory. Your valuable suggestions on scientific research have benefited me a lot. Care and love make me feel deeply in my heart, and we have had a deep friendship in the long years of getting along together, and this precious love of classmates will last a lifetime. Thanks to my school senior Seijun Morita and Ryuta Oosumi for their help and concern in my study and life, and thank my friends Xuyuan Li and Zheng Xu for their encouragement, support in their Ph.D. career. Thank you Ph.D. Wataru Kumagai for your guidance in scientific research and care in life. During my Ph.D. study in Mr. Yasuda's research group, other classmates in the laboratory gave me great help in study, work and life. I would like to express my gratitude to them. They are Masatoshi Hashimoto, Yuki Miyake, Ryu Fukushima, and Yuta Obinata and the rest of the classmates who are struggling together who are not mentioned here. It is precisely because of your encouragement, support, company and care along the way that I have persisted along the way.

Here, I would also like to thank my dear relatives, my dear parents, grandma, grandpa, and my girl friend. You unconditionally support any decision I make along the way and give me all the love you can give, especially my parents took on most of the tuition and living expenses for me, but I can only repay very little. In front of my heart, the language is sometimes so pale. Finally, with the most sincere heart, I would like to thank all the teachers, classmates and relatives who have not mentioned here and who have cared and helped me! Thank you for your support, encouragement, care and company, and sincerely wish my loved ones and those who love me every success in their future life!

Hongran Wang

March, 2021

Abstract

Optimization refers to "the act of selecting an option (solution) that produces the optimum result for a purpose from a large number of options under certain constraints". Optimization has been applied in a wide range of fields such as engineering, economics, and sociology, and its importance is widely recognized. Especially in engineering, it is possible to mathematically describe "problems that maximize or minimize the results (objective functions) for the objectives set for each problem" such as planning problems and product design problems as optimization problems. is there. Specific examples include the optimal design problem of dimensions, shape, and topology of industrial products, the optimization problem of factory production plans in the production field, the optimal operation problem of equipment in the energy field, the learning problem of neural networks in the machine learning field, and the financial engineering field. The problem of portfolio optimization is mentioned. These optimization problems are replaced as problems that determine the input (solution) so that the output (objective function) of the system has the optimum value for the purpose. A system refers to "a collection of multiple elements that interact with each other." Optimization for general systems (system optimization) has been established as one of the basic technologies of modern engineering.

In addition, in recent years, in light of the increasing scale and complexity of systems, the sophistication of system design, operation, control, and the sophistication of requirements for the performance of industrial products, new optimization problems and optimization methods with a strong awareness of practical applications have been introduced. Construction has become an important issue. For example, in the shortest path search problem as a practical application of single-objective optimization, it may be required to present not only the optimal solution that gives the shortest path but also multiple alternatives that take into account unexpected situations such as accidents and traffic jams. However, it is difficult to present multiple alternatives because the conventional single-objective optimization aims to search for only one global optimum solution or semi-optimal solution. Further, in the practical application of multi-objective optimization, for example, it may be required to consider the purpose for which it is difficult to evaluate the design, which is the subjective evaluation of the designer in the optimum design. However, conventional multi-objective optimization mainly deals with objectives that can be evaluated objectively, and it is difficult to consider objectives that are difficult to evaluate objectively. Due to the sophistication of requirements for practical applications of such optimization, there are requirements that are difficult to meet sufficiently with conventional optimization.

In order to consider these requirements, we propose and formulate a "superior solution set search problem" that aims to search for a solution set that meets the user's desire level and is composed of various solutions whose properties differ greatly from each other. The desired level in optimization is that the objective function value is superior to a certain level or more, and the difference in the properties of the solution can be evaluated by the degree of difference in the determinants (distance in the solution space). Based on the above, we propose a concept and define it based on it so that a superior solution set is constructed from solutions whose objective function values are superior to the criteria set by the user and whose solutions are far apart from each other. By giving the final solution from this superior solution set, it is expected that the requirements in practical application will be taken into consideration.

On the other hand, in actual optimization, there is a great need to find a solution (quasi-optimal solution) having sufficient optimization within a practical time, rather than finding only an exact optimal solution over a long period of time. Furthermore, in recent years, the dramatic increase in computer power has made a great contribution to numerical calculations such as optimization algorithms and simulations. In this way, the need for practical and new optimization methods is increasing in response to changes in the environment surrounding the optimization field (larger and more complicated actual systems, restrictions on calculation time, development of peripheral technologies). In recent years, metaheuristics (discovery approximate solution method) have been attracting attention as a framework of optimization methods that can respond to the above-mentioned "problems of conventional optimization methods in practical use" and "changes in the environment in the optimization field". There is. Metaheuristics is a framework of a solution direct search method that optimizes using only solution information and the corresponding evaluation value information of the target, and it is possible to obtain an approximate solution corresponding to it within a practical time. Does not require a mathematical model that guarantees differentiability and continuity. In addition, many metaheuristics are developed by taking inspiration from biological and physical phenomena, and are methods for searching for optimal or semi-optimal solutions. For example, Particle Swarm Optimization is a method based on foraging behavior as a school of birds and fish, and Firefly Algorithm (FA) is a method based on the blinking behavior of fireflies. In addition, metaheuristics have adjustable parameters, and the search can be performed efficiently by utilizing the degree of freedom of the parameters and setting them appropriately according to the problem structure and search conditions. In this way, metaheuristics with excellent versatility and search performance are attracting attention as optimization methods with high engineering value. Based on these backgrounds, this paper deals with the development of superior solution set search methods based on metaheuristics.

Based on the above, the optimization method for the superior solution set search problem in this paper is based on (1) the multimodality problem in single-objective optimization, and is a subset of all global and local optimal solutions. Inspired by the approach of indirectly solving the superior solution set search problem by acquiring the superior solution set, and (2) the solution update using the "superior relation" in the research field of multi-objective optimization, the superior solution we have taken the approach of directly acquiring the superior solution set by defining and using the superior relation by the desire level in the set search problem and explicitly including the definition of the superior solution set, and updating the solution using this superior relation. We proposed a superior solution set search method based on these approaches.

This paper is composed of 6 Chapters, and the outline of each chapter and the results obtained are as follows:

In Chapter 1, the background of the need for the proposals in this research, the purpose and outline of this research, and the structure of this paper were described.

In Chapter 2, it was stated that there are requirements in practical applications of optimization that are difficult to fully consider in usual optimization problems. In order to satisfy the above requirements, based on the single-objective optimization problem, a superior solution set is defined by a mathematical formula as a set of local optimal solutions whose evaluation values are superior by a certain amount or more and the distances between the solutions are separated by a certain amount or more. We proposed the "superior solution set search problem" with the goal of discovering this excellent solution set. A superior solution set search problem is novel in that it finds a solution as a set in a single-objecive optimization problem, and is expected to satisfy requirements that cannot be considered or are difficult to consider by usual optimization. It has usefulness in terms of points.

In Chapter 3, we analyzed the search structure of metaheuristics and extracted search strategies common to metaheuristics. Furthermore, while comparing typical metaheuristics, it was clarified that FA, which has a group of search points divided into multiple when applied to a multimodal function, has an affinity with the superior solution set search problem.

Chapter 4 proposes diversification and intensification for the superior solution set search problem in the approach of indirectly solving the superior solution set, and evaluates the realization state of diversification and intensification based on the analysis result of FA. Proposed. Based on diversification and centralization, we proposed an adaptive FA with an adaptive parameter adjustment function to make the evaluation index of the search state follow the preset target value schedule. Numerical experiments using several benchmark functions confirmed excellent search performance compared to the original FA.

In Chapter 5, we analyzed the properties of the superior solution set search problem in the approach of directly acquiring the superior solution set, and pointed out the structural similarities between the superior solution set search problem and the multi-objective optimization problem. We proposed a search strategy based on the "superior relation" that searches for a superior solution set by utilizing the analyzed properties and the user's desire level for the search. The definition of the superior solution set based on the superior relation is explicitly included, and the superior solution set search method is proposed. Then, we conducted numerical experiments on the superior solution set search problem and showed the usefulness of the proposed method while comparing the performance of the proposed method and the original FA.

Chapter 6 is the conclusion of this paper, and summarizes the research results and future research topics obtained in this paper.

Contents

Acknowledgments i

Abstract iii

1 Introduction 1

1.1 Background 1

1.2 Purpose and Positioning of This Paper 6

1.3 Structure of This Paper 8

2 Usual Optimization Problem and Superior Solution Set Search Problem 13

2.1 Introduction 13

2.2 Optimization Problem 13

2.2.1 Formulation of Optimization Problem 13

2.2.2 Classification of Optimization Problem 15

2.3 Usual Optimization Problem 15

2.3.1 Single-Objective Optimization Problem and Multimodal Optimization Problem 15

2.3.2 Multi-Objective Optimization Problem 17

2.4 Superior Solution Set Search Problem 20

2.4.1 Requirements for Practical Application of Optimization 20

2.4.2 Overview of Superior Solution Set Search Problem 22

2.4.3 Proposal of Superior Solution Set Search Problem 24

2.4.4 Comparison of Each Usual Optimization Problem and Superior Solution Set Search Problem 25
2.4.5 Properties of Algorithm Suitable for Searching the Superior Solution Set Problem 27
2.5 Summary 28

3 Metahuristics 29
3.1 Introduction 29
3.2 Metaheuristics Search Structure and Search Strategy 29
3.2.1 Interpretation of Metaheuristics 29
3.2.2 Metaheuristics Search Structure 30
3.2.3 Metaheuristics Search Strategy 31
3.2.4 Diversification and Intensification 32
3.3 Particle Swarm Optimization 35
3.3.1 Overview of Particle Swarm Optimization 35
3.3.2 Algorithm of Particle Swarm Optimization 35
3.3.3 Analysis of Particle Swarm Optimization Search Structure 38
3.4 Differential Evolution 40
3.4.1 Overview of Differential Evolution 40
3.4.2 Algorithm of Differential Evolution 40
3.4.3 Analysis of Differential Evolution Search Structure 44
3.5 Artificial Bee Colony Algorithm 47
3.5.1 Overview of Artificial Bee Colony Algorithm 47
3.5.2 Algorithm of Artificial Bee Colony Algorithm 47
3.5.3 Analysis of Artificial Bee Colony Algorithm Search Structure 51
3.6 Firefly Algorithm 53
3.6.1 Overview of Firefly Algorithm 53
3.6.2 Algorithm of Firefly Algorithm 54
3.6.3 Analysis of Firefly Algorithm Search Structure 57
3.7 Comparison of Each Method 59

3.7.1 Analyze the Characteristics of the Superior Solution Set Search Problem 59
3.7.2 Analysis of Each Method 60
3.8 Summary 60

4 Superior Solution Set Search Method Based on Clustering 63
4.1 Introduction 63
4.2 Research Approaches 64
4.3 Firefly Algorithm Based on Cluster Information 65
4.3.1 Analysis of Firefly Algorithm Parameter 66
4.3.2 Proposal of Parameter Adjustment Rule Based on Cluster Information 68
4.3.3 Numerical Experiment 72
4.4 Adaptive Firefly Algorithm Based on Evaluation and Control of Search State for Superior Solution Set Search Problem 78
4.4.1 Analysis of Firefly Algorithm Parameters and Diversification and Intensification 79
4.4.2 Evaluation Indicators of Diversification and Intensification for the Superior Solution Set Search 82
4.4.3 Proposal of Adaptive Firefly Algorithm 86
4.4.4 Numerical Experiment 92
4.5 Summary 95

5 Superior Relation Based Superior Solution Set Search Method 101
5.1 Introduction 101
5.2 Analysis and New Interpretation of Superior Solution Set Search Problem 102
5.2.1 Analysis Interpretation of Superior Solution Set Search Problem 102
5.2.2 New Interpretation of Superior Solution Set Search Problem 103
5.3 Proposal of Superior Relation in Superior Solution Set Search Problem 105
5.3.1 Definition of Superior Solution Fitness 107

5.4 Superior Relation Based Firefly Algorithm for Superior Solution Set Search Problem 108
5.4.1 Proposal of Superior Relation Based Firefly Algorithm 108
5.4.2 Numerical Experiment 112
5.5 Superior Relation Based Genetic Algorithm for Superior Solution Set Search Problem 124
5.5.1 Proposal of Superior Relation Based Genetic Algorithm 124
5.5.2 Numerical Experiment 125
5.6 Summary 135

6 Conclusion **137**
6.1 Summary 137
6.2 Future Issues 139

A Benchmark Function **145**

B *k*-means Clustering **151**
B.1 Overview of *k*-means Clustering 151
B.2 Algorithm of *k*-means Clustering 152

C Crossover and Mutation for Real Genetic Algorithm **155**
C.1 Simulated Binary Crossover 155
C.2 Polynomial Mutation 155
C.3 Real Genetic Algorithm 159

References **161**

1 Introduction

1.1 Background

Optimization refers to the act of selecting an option (solution) that produces the optimum result for the purpose from a large number of options under certain constraints. Optimization has been applied in a wide range of fields such as engineering, economics, and sociology, and its importance is widely recognized. Especially in engineering, it is possible to mathematically describe the optimization problem such as planning problems and product design problems that maximizes or minimizes the results (objective functions) for the objectives set for each problem. As specific examples, the size / shape / topology optimization design problem of industrial products [1][2], the production plan optimization problem of factories in the production field [3][4][5], equipment optimization operation problem in the energy field [6][7][8], neural network learning problem in the machine learning field [9], portfolio optimization problem in the financial engineering field [10][11] and so on. These optimization problems are replaced as problems that determine the input (solution) so that the output (objective function) of the system has the optimum value for the objective. A system is an aggregate of multiple elements that interact with each other. Historically, since the simplex method [12] was developed by Dantzig, the development and improvement of optimization methods and the expansion of the scope of application have been promoted. In modern times, optimization for general systems (system optimization) has been established as one of the basic technologies of modern engineering.

The form of optimization in engineering consists of the following elements.

(1) Target real system

(2) Mathematical model representing a real system

(3) Optimization techniques applicable to mathematical models

Originally, in the above optimization form, it is a normal road that (2) formulates the mathematical model so that an effective optimization method can be applied, and (3) selects the optimization method that is effective for the target problem. Therefore, (2) and (3) are closely related.

After the theory of optimization evolved from the field of mathematics called convex analysis [13], important concepts related to optimization such as solution optimization and optimization problem duality are developed. Even today, it plays an important role in the analysis of optimization problems and the design of optimization methods. In this way, the optimization method constructed and designed under the methodology for deriving the optimum solution for the optimization problem based on the above theory is called the mathematical programming method [14][15][16][17][18][19][20][21]. The mathematical programming method follows the theory of optimization and uses analytical information (gradients, Hessian, etc.) of mathematical models to find mathematically guaranteed solutions. As typical mathematical programming methods, the simplex method for linear programming problems and the Newton-Raphson method for nonlinear programming problems are historically well known. Since many mathematical findings and properties such as the convergence and optimality of solutions have been clarified for mathematical programming, in the above optimization form (3), the mathematical programming method has been used conventionally.

However, since the mathematical programming method requires analytical information on the optimization problem, the applicable optimization problem class is defined by the type of method. This is because when the mathematical programming method is adopted in (3) of the above optimization form, the dependencies of (2) and (3) are strong with each other, so the applicable classes of (2) and (3) are It shows that it is limited.

- Because various conditions such as the actual system can be expressed in a mathematical model by the specification of the method and differentiability and continuity of the mathematical model of various conditions are required, the scope of application of optimization is limited.
- As a result of expressing the mathematical model so as to match the method, there is a gap between the real system and the mathematical model, and the usefulness and feasibility of the solution become insufficient.

In this way, in optimization based on mathematical programming, there are many problems in putting it into practical use because the method imposes constraints on the form of optimization.

Recently, owing to large-scale and complex actual systems, the demand for obtaining a solution with sufficient optimality for practical use has increased. Meanwhile, with the tremendous development in computer technology and improvement of optimization algorithms and modeling / simulation technologies, the demand for not only practical but also new optimization methods is increasing in the field of optimization. Moreover, in actual optimization, there is a great need to find a solution (quasi-optimal solution) having sufficient optimization within a practical time, rather than finding only an exact optimal solution over a long period of time. In this way, there is an increasing need for practical and new optimization methods in response to changes in the environment (large-scale / complex actual systems, limitation of calculation time, development of peripheral technology) surrounding the optimization field.

In recent years, metaheuristics (heuristic-approximate optimization methods) [22][23][24][25][26] have been attracting attention as a framework of optimization methods that can respond to the above-mentioned problems of conventional optimization methods in practical use and changes in the environment of the optimization field. Metaheuristics are characterized by a direct solution search method, a practical approximation method, and a discovery method. The solution direct search method is a method of optimizing using only the solution information of the optimization problem and the objective function value information. When metaheuristics, which is a direct solution search method, are adopted in the form of optimization (3), the method does not impose constraints on the mathematical model, unlike the mathematical programming method. Therefore, the form of optimization becomes flexible, such as high-precision modeling and acquisition of objective function values directly from simulators and measuring instruments. In addition, a highly optimal approximate solution can be obtained according to the practical calculation time. n addition, many methods search for suboptimal solutions based on empirically superior mechanisms such as biological and physical phenomena. For example, Particle Swarm Optimization [27, 28, 29, 30] is a foraging behavior as a school of birds and fish, Genetic Algorithm is inspired by the process of natural selection that belongs to the larger class of Evolutionary Algorithms [33], Differential Evolution [34, 35] is a method based on the mechanism of evolution of living organisms that belongs to Evolutionary Algorithms, Artificial Bee Colony Algorithm [36, 37] is an optimization algorithm based on the intelligent foraging behaviour of honey bee swarm, and Firefly Algorithm [38, 39, 40] is inspired by the flashing behavior of fireflies. In addition, metaheuristics have adjustable parameters, and the search can be performed efficiently by utilizing the degree of freedom of the parameters and setting them appropriately according to the problem structure and search conditions. In this way, metaheuristics with excellent versatility and search performance are attracting attention as optimization methods with high engineering value. Based on these backgrounds, this paper also deals with metaheuristics as research subjects.

In recent years, as the requirements for the performance of industrial products of large-scale and complex actual systems, the design, operation and control of systems [23, 41, 42], the construction of a new optimization problem and a optimization method with a strong awareness of practical applications has become an important issue. For example, in the shortest path search problem as a practical application of single-objective optimization, it may be required to present not only the optimal solution that gives the shortest path but also multiple alternatives that take into account unexpected situations such as accidents and traffic jams [43], and production planning problems as a practical application of single-objective optimization may require a combination of multiple processes (presentation of alternatives) so that a change in the operating environment can be dealt with promptly [44]. However, it is difficult to present multiple alternatives because the conventional single-objective optimization aims to search for only one global optimal solution or semi-optimal solution [19]. Moreover, in the practical application of multi-objective optimization, the objectives that are difficult to formulate or objectively evaluate, such as the evaluation of the design that is the subjective evaluation of the designer in the optimum design [45], and the evaluation of the driver's preference in the shortest path search problem [46]. However, conventional multi-objective optimization mainly deals with objectives that can be evaluated objectively, and it is difficult to consider objectives that are difficult to formulate or objectively evaluate. For example, both aircraft optimization [47] and jet engine optimization [48] deal with physical properties that can be objectively evaluated, and do not consider objectives that are difficult to formulate or objectively evaluate. Due to the sophistication of requirements for practical applications of such optimization, there are requirements that are difficult to meet sufficiently with conventional optimization.

In order to consider these requirements, it is considered that the acquisition of various solution sets that satisfy the user's desire level is an effective means. The above solution set is, for example, an alternative in unforeseen circumstances. It is expected to meet requirements that are difficult to fully consider with conventional single-objective optimization. In addition, by giving a solution that considers the purpose that is difficult to formulate or objectively evaluate from various solution sets that satisfy the user's desire level. It is expected to meet requirements that are difficult to consider in conventional multi-objective optimization. Since the desire level in optimization can be evaluated based on the objective function value and the difference in properties between solutions can be evaluated based on the distance in the determinant space, various solution sets that are defined as a set composed of solutions with excellent objective function values and distances between solutions.

Until now, attempts have been made to obtain multiple solutions with excellent evaluation of the purpose based on the single-objective optimization problem [49, 50, 51, 52]. The document [49] seeks multiple solutions with excellent objective evaluation in order for the user to select the final solution from multiple options in the module optimal placement problem of the notebook PC. In the document [50], after pointing out that a solution that meets the specifications is not uniquely determined for design problems in the engineering field, in order to present multiple alternatives to the designer for LSI module placement problems, we propose an optimization method based on the Genetic Algorithm that searches for multiple optimal solutions. In the document [51] and the document [52], Particle Swarm Optimization has been proposed for efficiently searching for multiple optimal solutions. However, in any of the documents, the definition of the required solution set remains ambiguous, and a mathematically strict definition by mathematical formula is not made.

By the way, for the multi-modal optimisation problem, it is empirically known that there is a case where the objective function value is excellent and there are multiple local optimum solutions at different distances in the determinant space. It is highly similar to a set composed of solutions with excellent objective function values and distances between solutions. Such a multimodal problem appears in the modeling of a complicated real system, but the complexity of the real system cannot be fully considered on the premise of applying an optimization algorithm that constrains the modeling [23]. From the viewpoint of linking the modeling of the actual system and the optimization algorithm, it is important to assume the application of metaheuristics.

1.2 Purpose and Positioning of This Paper

The general metaheuristics for single-objective optimization problem, such as Particle Swarm Ptimization, Differential Evolution, and Artificial Bee Colony Algorithm etc., search for the only global solution or suboptimal solution, which are not suitable to search for multiple excellent solution sets. On the other hand, the Firefly Algorithm, which is a method of metaheuristics and simulates the courtship behavior of firefly, can search for multiple excellent solution sets because the search point group is divided into multiple search points. Therefore, it is considered that Firefly Algorithm has a basic property for the superior solution set search problem.

Then, with the superior solution set search problem as a new optimization problem, the approach to search for the only optimal solution by conventional single-objective optimization and the multi-objective optimization approach create requirements that are difficult to fully consider. In this paper, we propose an approach to search for the superior solution set for the following two items. Then, we develop the superior solution set search method based on these two approaches.

(a) In acquiring the superior solution set, we aim for an approach that indirectly achieves the purpose by acquiring all local solutions including the superior solution set. The approach dose not use parameters δ and ε, which define the superior solution set search problem, explicitly for evaluation of search points.

(b) We aim for an approach that directly achieves the purpose by acquiring the superior solution set. The approach uses parameters δ and ε, which define the superior solution set search problem, explicitly for evaluation of search points.

For approach (a), when applying FA to the superior solution set search, we propose an FA that utilizes cluster information by adding a mechanism (cluster) for dividing into multiple groups that is clearer. However, the adaptability of this algorithm is insufficient, and it is possible to consider the adaptability and further improve the adaptability by using a diversification and intensification strategy. In order to improve the search performance of metaheuristics in single-objective optimization problem, it is important to appropriately realize the search guideline of the diversification and intensification [19, 22, 23, 53]. In addition, by utilizing the diversification and intensification of single-objective optimization problems, it is possible to improve the adjustability of metaheuristics adjusting adaptive parameters based on the evaluation and control of search states. By applying the above-mentioned concept of diversification and intensification to the superior solution set search problem, efficient search and improvement of search performance can be realized at the same time. However, the superior solution set search problem searches for multiple local optimal solutions, in which the evaluation values of the objective functions are excellent and the solutions are separated from each other. Based on this, the search dynamics differ depending on the optimization problem, so the metaheuristics search strategy also differs in concreteness and construction. However, in order to search multiple promising regions in parallel in the superior solution set search problem, it is desirable that (1) while maintaining diversification among clusters, (2) the property of diversification and intensification within clusters like the conventional single-objective optimization. Based on the above, this paper develops an adaptive FA in which the parameter adjustment rule based on the evaluation and control of diversification and intensification is added to the FA by analyzing the parameters of the FA from the viewpoint of diversification and intensification.

For approach (b), it is the first proposal of a strict superior solution set search problem and the construction of a research base mainly for method development of the superior solution set search problem by defining a superior relation for the superior solution set search problem. As approach (a), in acquiring the superior solution set, we have taken an approach that indirectly achieves the purpose by acquiring all the local solutions including the superior solution set. However, approach (a) does not explicitly include a definition of the superior solution set in its algorithm, it cannot be said to be a strictly superior solution set search method because it does not reflect the user's desire level. This approach proposes a superior relation for the superior solution set, inspired by multi-objective optimization. This superior relation is applied to FA to develop the superior solution set search method.

1.3 Structure of This Paper

The structure of this paper is described below. Fig. 1.1 shows the chapter structure of this paper.

- Chapter 1 describes the background of this research, the purpose and outline of

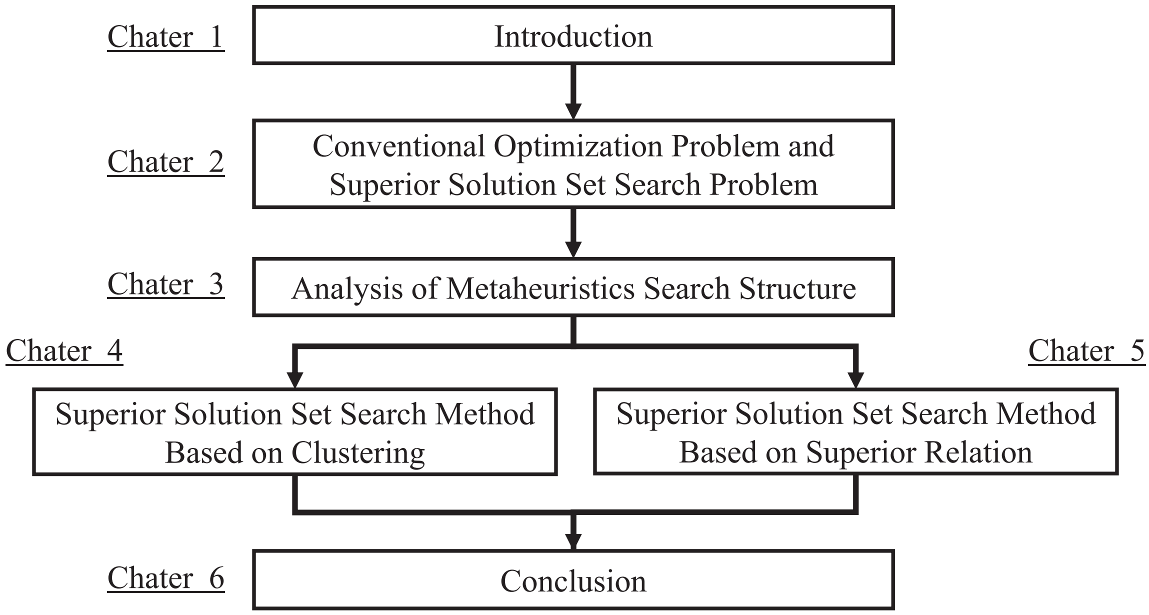


Fig. 1.1: Outline of Dissertation

this research, and the structure of this paper.

- Chapter 2 describes that there are requirements in the practical application of optimization that are difficult to fully consider in conventional optimization problems and optimization methods. Furthermore, we propose a " superior solution set search problem" that satisfies the above requirements and aims to search for a superior solution set with an excellent objective function value and a distance in the determinant space.
- In Chapter 3, we analyze the search structure of metaheuristics and extract search strategies common to metaheuristics. Furthermore, while comparing with typical metaheuristics such as Particle Swarm Optimization and Differential Evolution, we discuss Firefly Algorithm in which the search point group is divided into multiple parts when applied to a multimodal function in a single-objective optimization problem to clarify an affinity with a superior solution set search

problem.

- In Chapter 4, from the analysis of the search structure of Firefly Algorithm, it is stated that Firefly Algorithm has the property of searching multiple promising regions in parallel, and it is clear that it has a high affinity for the superior solution set search problem. I made it. By taking advantage of this property and incorporating a cluster structure, we propose a parameter adjustment rule for Firefly Algorithm. Then, assuming a basic case in the superior solution set search problem, a numerical experiment is performed for a benchmark function having multiple optimal solutions that are separated from each other. We compare the Firefly Algorithm with Firefly Algorithm based on cluster information and adaptive Firefly Algorithm, and examine the usefulness of the proposed method. In addition, from the viewpoint of diversification and intensification and practical optimization, the adaptability and search performance for the superior solution set search problem are improved by the approach based on the search strategy in metaheuristics for the superior solution set search problem. We propose an adaptive Firefly Algorithm. Then, assuming a basic case in a superior solution set search problem, a numerical experiment is performed for a benchmark function having multiple optimal solutions that are separated from each other. We compare the adaptive Firefly Algorithm with Firefly Algorithm and Firefly Algorithm based on cluster information to examine the usefulness of the proposed method.
- In Chapter 5, we propose the superior relation in the superior solution set search problem. It starts with the definition of the superior solution set search problem and the superior relation, and goes through the consideration of the superior solution set search problem in applying the superior relation to the application to the conventional single-objecive optimization method. Then, the numerical experiment is described. It is shown not only theoretically but also experimentally that the superior solution set can be searched directly by comparing the superior solution set search performance before and after the application of the superior relation. Then, the usefulness of the proposed method is shown.

- Chapter 6 describes the achievements obtained in this paper and future issues and prospects.

2 Usual Optimization Problem and Superior Solution Set Search Problem

2.1 Introduction

As a preparation for the analysis of metaheuristics, which is the subject of this paper, the basic structure of general optimization methods is mathematically organized and described in a unified manner [54][55][56][57][58]. First, we give a concrete formulation of the optimization problem dealt with in this paper. Next, the structure of the optimization method for solving the optimization problem is mathematically described. An overview of the optimization method forms the basis for the analysis of the metaheuristic search structure performed in the Chapter 3.

2.2 Optimization Problem

2.2.1 Formulation of Optimization Problem

The optimization problem is formulated as a problem to find the value of the decision variable that maximizes or minimizes the objective function under the given constraints.

Definition 2.1 (Optimization Problem) In N-dimensional real space $\boldsymbol{X} \in \mathbb{R}^N$, give a real space function whose domain is $f : \mathbb{R}^N \to \mathbb{R}^1$ and the non-empty closed set $\boldsymbol{S} \in \mathbb{R}^N$. The problem of minimizing the objective function $f(\boldsymbol{x})$ on the set $\boldsymbol{S}$ defines as the Eq.(2.1).

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \tag{2.1a}$$

$$\text{subj.to} \quad \boldsymbol{x} \in \boldsymbol{S} \tag{2.1b}$$

$$\boldsymbol{S} \subseteq \boldsymbol{X} \tag{2.1c}$$

Here, $\boldsymbol{S}$ is the feasible region. $\boldsymbol{x} \in \boldsymbol{S}$ is called an feasible solution, and $\boldsymbol{x} \notin \boldsymbol{S}$ is called an infeasible solution. □

For the solution $\boldsymbol{y}$ and the small positive number $\varepsilon > 0$ in the basic space, the open set is called the neighborhood $\boldsymbol{B}(\boldsymbol{x}, \varepsilon)$ of $\boldsymbol{x}$ in the Eq.(2.2). The schematic diagram of neighborhood $\boldsymbol{B}(\boldsymbol{x}, \varepsilon)$ is shown in Fig.2.1.

$$\boldsymbol{B}(\boldsymbol{x}, \varepsilon) = \{\boldsymbol{x} \in \boldsymbol{X} | \, \|\boldsymbol{x} - \boldsymbol{y}\| < \varepsilon\} \tag{2.2}$$

For any solution $\boldsymbol{x} \in \boldsymbol{S}$, the solution $\boldsymbol{x}^* \in \boldsymbol{S}$ that satisfies the condition of the Eq.(2.3) is called the global optimal solution.

$$\forall \boldsymbol{x} \in \boldsymbol{S}, \;\; f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \tag{2.3}$$

For a global optimal solution $\boldsymbol{x}^*$, exist a neighborhood $\boldsymbol{B}(\boldsymbol{x}', \varepsilon)$ of a viable solution $\boldsymbol{x}'$ and the local optimal solution $\boldsymbol{x}'$ satisfies the condition of the Eq.(2.4).

$$\forall \boldsymbol{x} \in \boldsymbol{B}(\boldsymbol{x}', \varepsilon) \cap \boldsymbol{S}, \;\; f(\boldsymbol{x}') \leq f(\boldsymbol{x}) \tag{2.4}$$

2.2.2 Classification of Optimization Problem

Optimization problem is classified into several types as follows, depending on the properties of the feasible region $\boldsymbol{S}$ by the Eq.(2.1) and the objective function $f(\cdot)$.

Linearity (or Non-linearity) of Objective Function (or Constraint Function):
Linear Programming Problem, Nonlinear Programming Problem (High-dimensional, Multimodal)

With (or Without) Constraint Function:
Unconstrained Optimization Problem, Constrained Optimization Problem

Objective Function Scalar (or Vector):
Single-objective Optimization Problem, Multi-objective Optimization Problem

Continuous (or Discrete) Decision Variables:
Continuous Optimization, Combinatorial Optimization Problem, Mixed Integer Programming Problem

For a specific type of problem, a powerful optimization method for obtaining a global optimal solution is known. In particular, when solving an optimization problem, if the class of the problem is known, it is possible to apply an appropriate optimization algorithm for that class.

2.3 Usual Optimization Problem

2.3.1 Single-Objective Optimization Problem and Multimodal Optimization Problem

In this paper, we will introduce various problems based on continuous and unrestricted optimization problem. First, we introduce the continuous optimization problem through the following Definition 2.2.

Definition 2.2（Continuous Optimization Problem） Objective function $f(\boldsymbol{x})$ with $m \geq 0$ inequality constraints $g_i(\boldsymbol{x})$ and $l \geq 0$ equality constraints $h_j(\boldsymbol{x})$ is formulated as follows:

$$\min_{x} \quad f(\boldsymbol{x}) \tag{2.5a}$$

$$\text{subj.to} \quad g_i(\boldsymbol{x}) \leq 0 \quad i = 1, \ \cdots, \ m \tag{2.5b}$$

$$h_j(\boldsymbol{x}) = 0 \quad j = 1, \ \cdots, \ l \tag{2.5c}$$

Where $f : \mathbb{R}^N \rightarrow \mathbb{R}^1$ is the objective function to be minimized over the N-variable vector $\boldsymbol{x} \in \boldsymbol{S} \subset \mathbb{R}^N$. $\boldsymbol{S}$ represents the feasible region. □

If $m = l = 0$, it is called an unconstrained optimization problem. Without loss of generality, we discuss a single-objective optimization problem and a multimodal optimization problem based on the unconstrained and continuous optimization problem in this paper. In the single-objective optimization problem, the main purpose is to find the global optimal solution $\boldsymbol{x}^*$ is given as the Eq.(2.6) in objective function space [19]. The schematic diagram of global optimal solution is shown in Fig.2.1.

$$\boldsymbol{x}^* = \{f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}) \ (\forall \boldsymbol{x} \in \boldsymbol{S})\} \tag{2.6}$$

In the multimodal optimization problem, the main purpose is to find multiple optimal solutions set $\boldsymbol{OS}$ (global and local) given as the Eq.(2.7), so that the user can have a better knowledge about different optimal solutions in the search space and as and when needed, the current solution may be switched to another suitable optimum solution. The schematic diagram of $\boldsymbol{OS}$ is shown in Fig.2.1.

$$\boldsymbol{OS} = \{\boldsymbol{x}' \mid f(\boldsymbol{x}') \leq f(\boldsymbol{x}) \ (\forall \boldsymbol{x} \in \boldsymbol{B}(\boldsymbol{x}', \varepsilon))\} \tag{2.7}$$

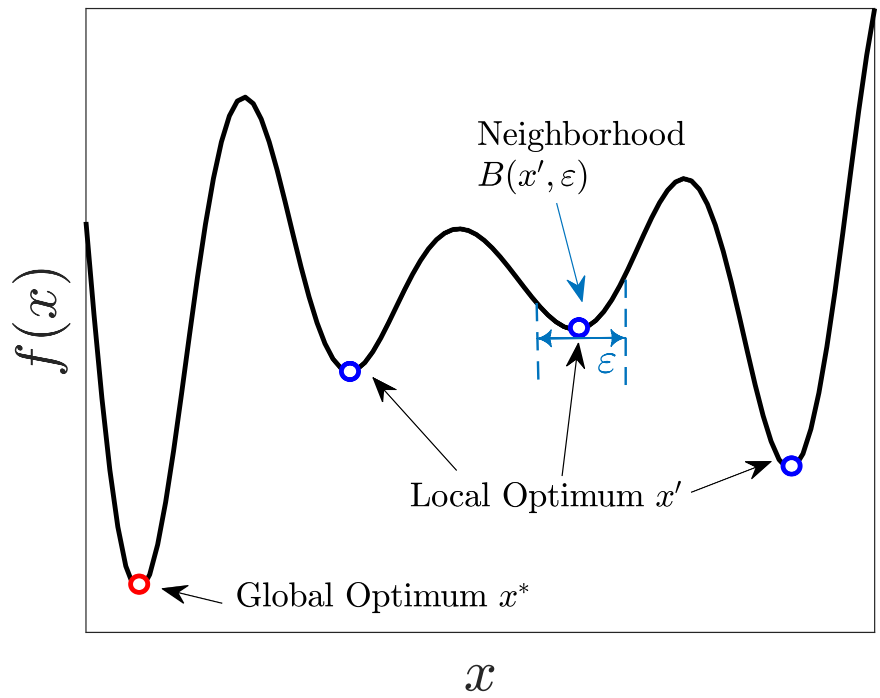


Fig. 2.1: Overview of Various Definitions of Optimization Problem

2.3.2 Multi-Objective Optimization Problem

(a) Overview of Multi-Objective Optimization Problem

When humans make decisions, many need to consider multiple indicators at the same time. For example, when choosing an apartment, we consider from many perspectives such as "rent", "distance from the station", and "floor plan". In product design, it is necessary to balance conflicting criteria such as "durability" and "lightness". Such a problem that makes the best choice (determination of a solution) by considering multiple objectives at the same time is formulated as a multi-objective optimization problem [59]. In the following, the definitions of the multi-objective optimization problem are described for the case of minimizing all objective functions.

Definition 2.3 (Multi-objective Optimization Problem) When the N-variable vector is $\boldsymbol{x} \in \mathbb{R}^N$ and the feasible region is $\boldsymbol{S} \subseteq \mathbb{R}^N$, the multi-objective optimization problem that minimizes the r objective function vector $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ \cdots, \ f_r(\boldsymbol{x})]^\top$ is defined as follows:

$$\min_{\boldsymbol{x}} \quad \boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ \cdots, \ f_r(\boldsymbol{x})]^\top \tag{2.8a}$$

$$\text{subj.to} \quad \boldsymbol{x} \in \boldsymbol{S} \tag{2.8b}$$

□

The superiority or inferiority of the solution in multi-objective optimization is judged by the dominance relation. When the Eq.(2.9) is satisfied, the solution $\boldsymbol{x}$ dominates the solution $\boldsymbol{y}$ ($\boldsymbol{x} \prec \boldsymbol{y}$). Hereafter, $k = 1, \ 2, \ \cdots, r$.

$$\boldsymbol{x} \prec \boldsymbol{y} \iff \forall k, \ f_k(\boldsymbol{x}) \leq f_k(\boldsymbol{y}) \ \wedge \ \exists k, \ f_k(\boldsymbol{x}) < f_k(\boldsymbol{y}) \tag{2.9}$$

The optimal solution in multi-objective optimization is defined as the Pareto solutions, which is not superior to all other solutions. Pareto solutions generally are a set where is no objective superiority or inferiority relationship between Pareto solutions. The Pareto solutions set $\boldsymbol{PS}$ is given as in the Eq.(2.10). The schematic diagrams of $\boldsymbol{PS}$ in determination variable space and objective function space are shown in Figs.2.2 and 2.3.

$$\boldsymbol{PS} = \{\boldsymbol{x} \in \boldsymbol{X} \mid \forall \boldsymbol{y} \in \boldsymbol{X}, \boldsymbol{y} \not\prec \boldsymbol{x}\} \tag{2.10}$$

(b) Request for Multi-Objective Optimization

In multi-objective optimization, the superiority or inferiority of Pareto solutions cannot be objectively determined, and Pareto solutions generally exist as a set. Due to this property, the following two approaches can be considered as solutions to multipurpose optimization problems. Each of these approaches has its advantages.

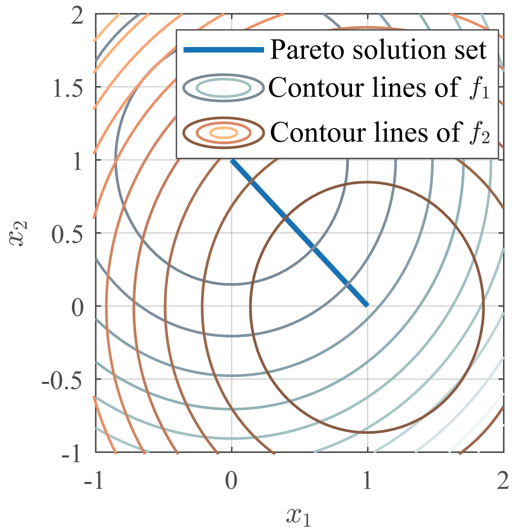


Fig. 2.2: Schematic Diagram of the Coefficient of Determination Variable Space in Multi-objective Optimization (Pareto Solution Set Generated by Two Functions)

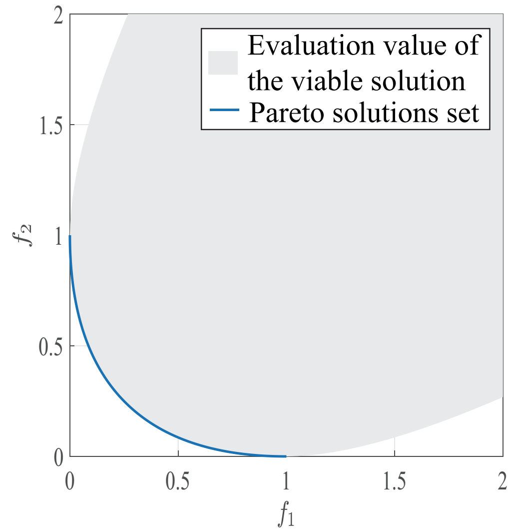


Fig. 2.3: Schematic Diagram of the Objective Function Space in Multi-objective Optimization (Corresponds to the Fig.2.2)

(1) By using the user's preference information, the only Pareto solution according to the preference is obtained. It is a solution to traditional multi-objective optimization problems and requires user preference. Typical methods include linear scalarization [59] and ϵ-constraint method [59].

(2) Find a large number of Pareto solution sets that approximate the Pareto frontier. This is a method that applies the multipoint search type discovery approximation method to multi-objective optimization, and does not require user preference information.

From the viewpoint of engineering application, in addition to finding the "optimal solution" itself, there is a demand for "information that supports decision-making" such as trade-off relationships between objectives and dependencies between objectives and decision-making variables [48]. Since the method (2) requires a large number of Pareto solutions by combining the method (2) with knowledge extraction techniques such as data mining. Research is conducting on "multi-objective optimization for decision-making support" that extracts useful knowledge for decision-making from the Pareto solution set and utilizes it for decision-making.

2.4 Superior Solution Set Search Problem

2.4.1 Requirements for Practical Application of Optimization

In the practical application of optimization, presentation of multiple alternatives assuming unforeseen circumstances that is difficult to fully consider in general single-objective optimization and consideration of objectives that are difficult to formulate or objectively evaluate that are difficult to fully consider in general multi-objective optimization are required.

As an example where it is necessary to present an alternative plan assuming an unforeseen situation, (1) consideration of accidents and traffic jams in the shortest path search problem [43], (2) response to changes in the operating environment in the production planning problem [44], (3) consideration of technical problems in the shape design optimization problem.

(1) In the shortest path search problem as a practical application of single-objective optimization, multiple routes (candidates) may be required to avoid the effects of accidents and traffic jams. However, it is difficult to present multiple alternatives because the usual single-objective optimization searches for the only combination of routes with the shortest distance among the innumerable routes from the starting point to the target point.

(2) In the production planning problem as a practical application of single-objective optimization, it may be necessary to combine multiple processes so that it can respond quickly when a change in the operating environment occurs. However, it is difficult to present multiple alternatives in the usual single-objective optimization because it searches for the only combination of processes that can reduce the production cost most.

(3) In the example of the shape design optimization problem as a practical application of single-objective optimization, if a technical problem occurs at the design stage, an alternative design plan is required. However, in usual single-objective optimization, it is difficult to present multiple alternatives in order to search for the only combination of dimensions that optimizes physical properties.

In this way, usual single-objective optimization aims to search for only one global optimal solution or semi-optimal solution, so it is difficult to present multiple solutions that can be alternatives required in practical applications.

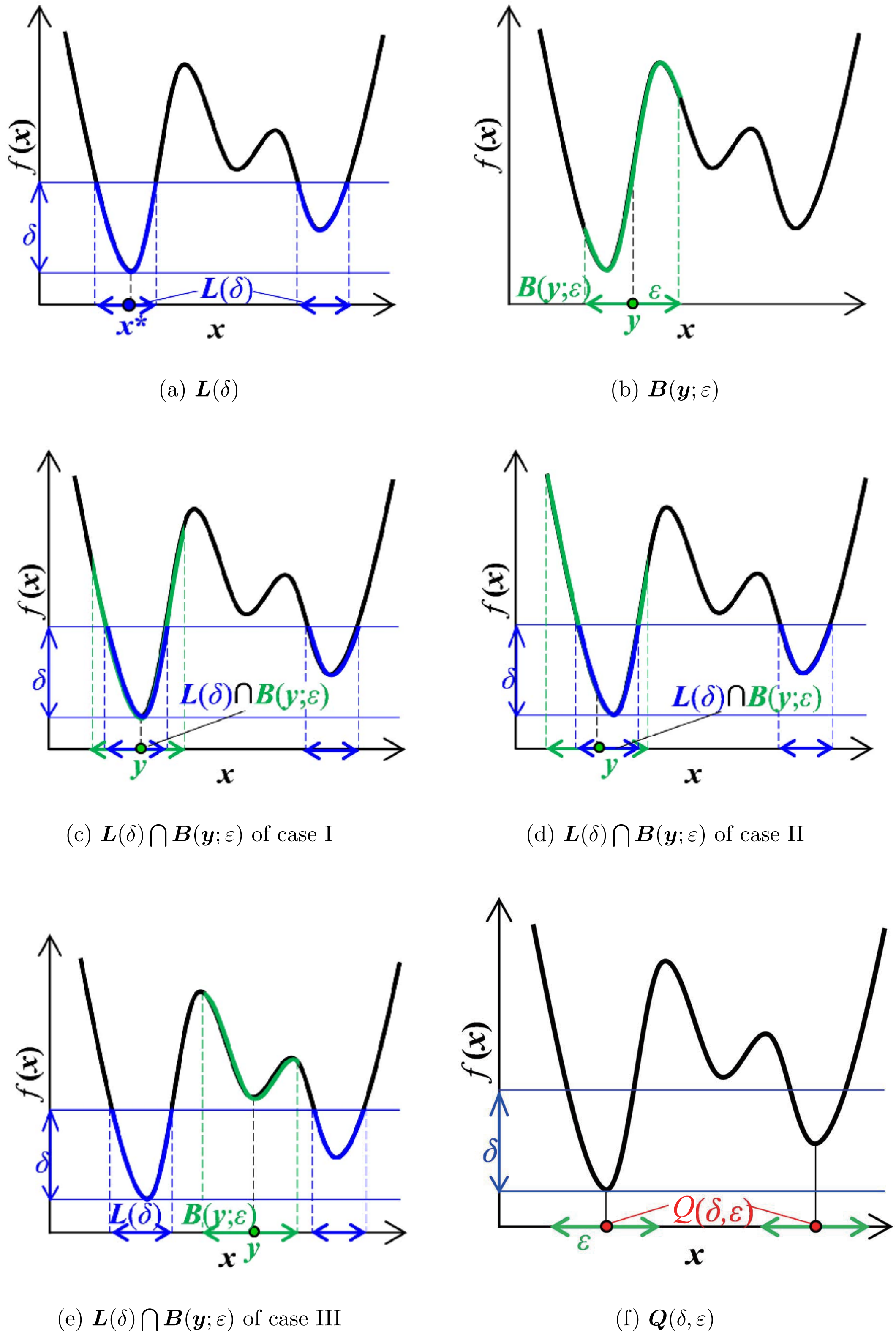
In addition, examples that require consideration of objectives are difficult to formulate or objectively evaluate. Design consideration in the shape design optimization problem [45] and taste consideration in the driver's in the shortest path search problem [46] can be mentioned. Considering the morphological optimization of the free-form surface shell structure as an example of the shape design optimization problem, it is desirable that the design and structural rationality can be considered at the same time in the morphological optimization of such a free-form surface shell structure. In the shortest path search problem, it is desirable to be able to consider the distance to the destination and the preference of the person traveling (a route with a wide road, a route with many straight lines, etc.) at the same time. However, in the usual multi-objective optimization, since the purpose that can be evaluated objectively is mainly dealt with, it is difficult to consider the design and structural rationality, and the distance to the destination and the preference of the moving person at the same time. Therefore, it can be said that it is difficult to consider the purpose for which formulation and objective evaluation are difficult with the usual multi-objective optimization. As described above, in the practical application of optimization, there are multiple requirements that are difficult to meet with the usual optimization approach.

2.4.2 Overview of Superior Solution Set Search Problem

In single objective optimization, if a variable decision (solution) is given, it is determined as the objective function value corresponding to the performance of the solution value. However, it is possible that solutions that are away from each other may have different characteristics in terms of performance and properties, even if their evaluated values (objective function values) are about the same. Therefore, various solution sets are expected having excellent objective function values and considerably large mutual distances in the solution space apart for meeting the demands of practical application.

We explain the superior solution set search problem [60][61] proposed to search for various solution sets with similar evaluated values and long solution distances. Obtaining the superior solution set makes it possible to present alternatives for accidents and technical problems, which are very important in engineering research. In actual optimization, for example, as seen in industrial design, there are objects that can be objectively evaluated (such as product performance) and those that can be subjectively evaluated (such as design). Moreover, there are optimization problems having multiple purposes and different properties.

Based on the above description, we propose a concept and define it based on a solution in which the objective function value is superior to the criterion determined by the user, and the superior solution set is constructed from solutions that are far apart from each other in the solution space. By choosing the final solution from the superior solution set, we expect to satisfy the requirements of actual application. Furthermore, there are parameters that can be arbitrarily determined by the user from the excellent solution set. Because it is possible to adjust the objective function value of the solutions included in the superior solution set and the distance between solutions, it is expected that users can respond to different demands.



(a) $\boldsymbol{L}(\delta)$ (b) $\boldsymbol{B}(\boldsymbol{y};\varepsilon)$

(c) $\boldsymbol{L}(\delta)\bigcap\boldsymbol{B}(\boldsymbol{y};\varepsilon)$ of case I (d) $\boldsymbol{L}(\delta)\bigcap\boldsymbol{B}(\boldsymbol{y};\varepsilon)$ of case II

(e) $\boldsymbol{L}(\delta)\bigcap\boldsymbol{B}(\boldsymbol{y};\varepsilon)$ of case III (f) $\boldsymbol{Q}(\delta,\varepsilon)$

Fig. 2.4: An Example of the Superior Solution Set in a Multimodal Function of One-dimensional ($N = 1$)

2.4.3 Proposal of Superior Solution Set Search Problem

Based on the Section 2.4.2, we define a superior solution set proposed in this paper. However, this paper deals with the minimization problem of the objective function $f(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^N$). Fig.2.4 shows an example of a superior solution set in a multimodal function of one-dimensional. The horizontal axis represents the decision variable and the vertical axis represents the objective function value.

Definition 2.4 (**Solution Set $\boldsymbol{L}(\delta)$ Considering the Objective Function Value**) Define a solution set $\boldsymbol{L}(\delta)$ that takes into account the objective function value. The Eq.(2.11) defines a set $\boldsymbol{L}(\delta) \subseteq \boldsymbol{X}$ of solution $\boldsymbol{x} \in \boldsymbol{X}$ levels that satisfy the constraints $\delta \geq 0$ of the objective function value based on the objective function value of the global optimal solution $f(\boldsymbol{x}^*)$. Where $\boldsymbol{X}$ represents the feasible region.

$$\boldsymbol{L}(\delta) = \{\boldsymbol{x} \in \boldsymbol{X} \mid f(\boldsymbol{x}) \leq f(\boldsymbol{x}^*) + \delta\} \tag{2.11}$$

From the Eq.(2.11) and Fig.2.4(a), $\boldsymbol{L}(\delta)$ is a solution set considering the objective function value, which is determined by the global optimum solution $\boldsymbol{x}^*$ and the parameter δ defined by the user. □

Definition 2.5 (**Solution Set $\boldsymbol{B}(\boldsymbol{y}, \varepsilon)$ Considering Distance**) Define a solution set $\boldsymbol{B}(\boldsymbol{y}; \varepsilon)$ considering the distance. Define ε-neighborhood $\boldsymbol{B}(\boldsymbol{y}, \varepsilon)$ (open sphere with radius $\varepsilon > 0$ centered on $\boldsymbol{y}$) for any solution $\boldsymbol{y} \in \mathbb{R}^N$ in the Eq.(2.12).

$$\boldsymbol{B}(\boldsymbol{y}, \varepsilon) = \{\boldsymbol{x} \in \mathbb{R}^N \mid \|\boldsymbol{x} - \boldsymbol{y}\| < \varepsilon\} \tag{2.12}$$

From Eq.(2.12) and Fig.2.4(b), $\boldsymbol{B}(\boldsymbol{y}, \varepsilon)$ is a solution set considering the distance in the solution space, which is determined by the any solution $\boldsymbol{y}$ and the parameter ε defined by the user. □

Definition 2.6 (Superior Solution Set $\boldsymbol{Q}(\delta, \varepsilon)$) The superior solution set $\boldsymbol{Q}(\delta, \varepsilon)$ is defined from $\boldsymbol{L}(\delta)$ and $\boldsymbol{B}(\boldsymbol{y}; \varepsilon)$. Eq.(2.13) defines the superior solution set $\boldsymbol{Q}(\delta, \varepsilon)$ as a superior solution $\boldsymbol{y} \in \boldsymbol{L}(\boldsymbol{x}^*, \delta)$ satisfies $f(\boldsymbol{y}) \leq f(\boldsymbol{x})$ $(\forall \boldsymbol{x} \in \boldsymbol{L}(\boldsymbol{x}^*, \delta) \cap \boldsymbol{B}(\boldsymbol{y}, \varepsilon))$.

$$\boldsymbol{Q}(\delta, \varepsilon) = \{\boldsymbol{y} \in \boldsymbol{L}(\delta) \mid f(\boldsymbol{y}) \leq f(\boldsymbol{x}) \ (\forall \boldsymbol{x} \in \boldsymbol{L}(\delta) \cap \boldsymbol{B}(\boldsymbol{y}, \varepsilon))\} \tag{2.13}$$

□

A superior solution belongs to $\boldsymbol{L}(\delta) \cap \boldsymbol{B}(\boldsymbol{y}; \varepsilon)$, but $\boldsymbol{L}(\delta) \cap \boldsymbol{B}(\boldsymbol{y}; \varepsilon)$ depends on the central $\boldsymbol{y}$ of the ε-neighborhood. As shown in Fig.2.4(c), if $\boldsymbol{y} \in \boldsymbol{L}(\boldsymbol{x}^*; \delta)$ satisfies the condition of $f(\boldsymbol{y}) \leq f(\boldsymbol{x})$ in the Eq.(2.13), $\boldsymbol{y}$ is a superior solution. On the other hand, if $\boldsymbol{y}$ is not the local optimal solution as shown in Fig.2.4(d), $f(\boldsymbol{y}) \leq f(\boldsymbol{x})$ in the Eq.(2.13) does not meet the condition and is not a superior solution. Also, as shown in Fig.2.4(e), if $\boldsymbol{y}$ is a local optimal solution but does not belong to $\boldsymbol{L}(\delta)$, the Eq.(2.13) does not satisfy $\boldsymbol{y} \in \boldsymbol{L}(\delta)$ and is not a superior solution. Therefore, as shown in Fig.2.4(f), the superior solution set $\boldsymbol{Q}(\delta, \varepsilon)$, which is a set of diverse local optimal solutions, has the difference between from the objective function value of the global optimal solution is within δ and has various local solutions with distances of ε or more. We propose an optimization problem that aims to search for the superior solution set as the superior solution set search problem. From the above, it is possible to define various solution sets that satisfy the demand level described at the beginning of this Section by the user setting the parameters δ and ε appropriately.

2.4.4 Comparison of Each Usual Optimization Problem and Superior Solution Set Search Problem

In usual optimization problems, both the multi-objective optimization and the multimodal optimization problem are searching for multiple solutions, which can present multiple options for user. For the multi-objective optimization problem, multiple solutions are objectively presented based on the superiority or inferiority of the evaluation

Table 2.1: Comparison of Each Usual Optimization Problem and Superior Solution Set Search Problem

Optimization Problem	Coefficient Space	Objective Function Space
Superior Solution Set Search Problem	Constrained by Distance	Constrained by Constant
$\{\boldsymbol{x} \mid \forall \boldsymbol{y} \in \boldsymbol{L}(\delta) \cap \boldsymbol{B}(\boldsymbol{x}, \varepsilon), f(\boldsymbol{x}) \leq f(\boldsymbol{y})\}$	$\forall \boldsymbol{y} \in \boldsymbol{B}(\boldsymbol{x}, \varepsilon), 0 < \varepsilon < \infty$	$\forall \boldsymbol{y} \in \boldsymbol{L}(\delta), f(\boldsymbol{x}) \leq f(\boldsymbol{y})$
Single-objective Optimization Problem	Unconstrained	Superiority by Fitness
$\{\boldsymbol{x} \mid \forall \boldsymbol{y} \in \boldsymbol{X}, f(\boldsymbol{x}) \leq f(\boldsymbol{y})\}$	$\forall \boldsymbol{y} \in \boldsymbol{X}$	$f(\boldsymbol{x}) \leq f(\boldsymbol{y})$
Multimodal Optimization Problem	Unconstrained	Superiority by Fitness
$\{\boldsymbol{x} \mid \forall \boldsymbol{y} \in \boldsymbol{B}(\boldsymbol{x}, \varepsilon), f(\boldsymbol{x}) \leq f(\boldsymbol{y})\}$	$\forall \boldsymbol{y} \in \boldsymbol{B}(\boldsymbol{x}, \varepsilon), \varepsilon \gg 0$	$f(\boldsymbol{x}) \leq f(\boldsymbol{y})$
Multi-objective Optimization	Unconstrained	Dominance Relationship by Fitness
$\{\boldsymbol{x} \mid \forall \boldsymbol{y} \in \boldsymbol{X}, \boldsymbol{y} \not\prec \boldsymbol{x}\}$	$\forall \boldsymbol{y} \in \boldsymbol{X}$	$\boldsymbol{y} \not\prec \boldsymbol{x}$

values. Similarly, for the multimodal optimization problem, all optimization solutions are objectively presented based on the evaluation values. Here, both the multi-objective optimization and the multimodal optimization problem are only setting in objective function space by superiority of the evaluation values, and are unconstrained in coefficient space (see Table 2.1).

The superior solution set search problem is also seaching for the multiple solutions, which not only have excellent objective function values but also are separated from each other. However, compared with the multi-objective optimization and the multimodal optimization problem, the superior solution set search problem is constrained by constant (δ) in objective function space and constrained by distance (ε) in coefficient space. For the superior solution set search problem, we propose that other optimization problems do not have distance constraints between the solutions in coefficient space (see Table 2.1). Therefore, when proposing multiple alternatives, make a subjective or objective assessment of the target.

From the Eq.(2.13), when changing parameters δ and ε which define superior solution set $\mathcal{Q}(\delta, \varepsilon)$ can adjuste the distribution transition of the superior solution set search problem. Especially, when $\delta = 0$ and $\varepsilon \gg 0$, the superior solution set search problem

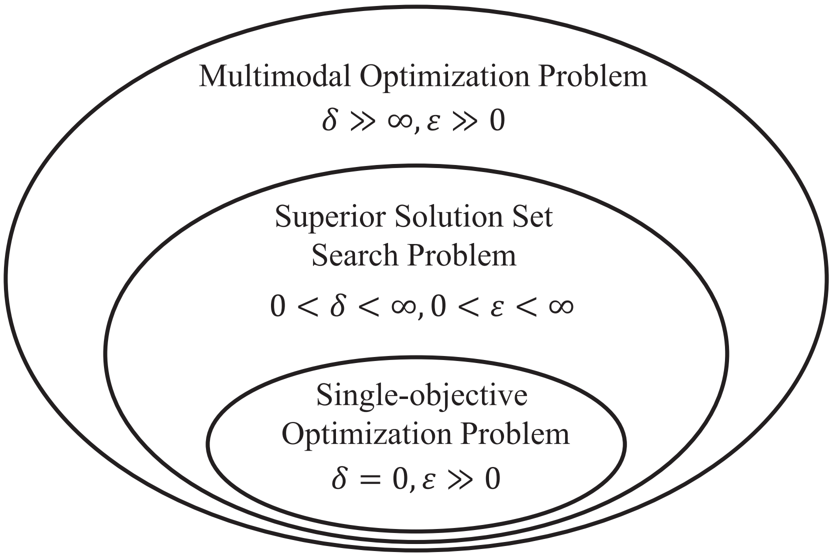


Fig. 2.5: Containment Diagrams of Various Optimization Problems

becames to the single-objective optimization problem. when $\delta \gg 0$ and $\varepsilon \gg 0$, the superior solution set search problem becames to the multimodal optimization problem. Containment diagrams of various optimization problems is shown in Fig.2.5.

2.4.5 Properties of Algorithm Suitable for Searching the Superior Solution Set Problem

As described above, the superior solution set is composed of a plurality of locally optimal solutions with excellent objective function values and distances from each other. In order to search for the superior solution set with such characteristics, an algorithm that can search in parallel near multiple local optimal solutions (promising regions) with excellent objective function values and distances between solutions is required. In order to search multiple promising areas in parallel, it is desirable to have properties that (a) maintain diversity among clusters, (b) diversify and intensify within the cluster as in the usual single-objective optimization method. In addition, since it is desirable to be able to maintain the complexity of the actual system during modeling, it is desirable to use metaheuristics, which are less restrictive to modeling because they are direct search types. On the other hand, many of the metaheuristics proposed so far aim to search for the only global optimal solution or quasi-optimal solution, so it is difficult to search multiple promising regions in parallel. In Chapter 3, we list several typical metaheuristics and consider each method of metaheuristics from the above viewpoint.

In Chapter 4, the optimization method for the superior solution set search problem is based on the multimodal optimization problem in single-objective optimization, and is a subset of all global and local optimal solutions. Inspired by the approach of indirectly solving the superior solution set search problem by acquiring the superior solution set.

In Chapter 5, the solution update using the "superior relation" in the research field of multi-objective optimization, we have taken the approach of directly acquiring the superior solution set by defining and using the superior relation by the desire level in the set search problem and explicitly including the definition of the superior solution set, and updating the solution using this superior relation.

2.5 Summary

In this chapter, based on the single-objective optimization problem, a superior solution set was defined by a mathematical formula as a set of local optimal solutions whose evaluation values are superior by a certain amount or more and the distance between solutions is a certain distance or more. We proposed the superior solution set search problem with the goal of finding this solution set. It is useful to meet difficult requirements that cannot be considered by conventional optimization.

3 Metahuristics

3.1 Introduction

In this Chapter, the search strategies and search structures common to metaheuristics are extracted from the search structure analysis of typical metaheuristic algorithm. Furthermore, while comparing with typical metaheuristics such as Particle Swarm Optimization and Differential Evolution etc., we clarify that the Firefly Algorithm has an affinity with the superior solution set search problem, in which the search point swarm is divided into multiple parts when applied to a multimodal function in a single-objective optimization problem.

3.2 Metaheuristics Search Structure and Search Strategy

3.2.1 Interpretation of Metaheuristics

Metaheuristics are generally based on analogies with various phenomena such as biological and physical. To adaptively adjust the parameters in accordance with the structure of the problem to be optimized and the search situation and develop a practical optimization method with excellent search performance, it is necessary to not only analyze the essential parts of many metaheuristics but also extract and use the search structures and search strategies of ordinary metaheuristics. In this Section, we discuss an important search structure and search strategy used in many metaheuristics.

3.2.2 Metaheuristics Search Structure

Metaheuristics are focused on biological phenomena, and it is important to consider the essential search structure of metaheuristics by abstracting out the viewpoint based on analogy and excellent mechanisms [19][22][23]. We analyze various metaheuristics and there are two common operations below.

Neighborhood generation: from the current search point, it is an operation to generate destination candidates using information such as interaction between search points and search history.

Updating Search Point: the search point operation is moved based on a certain rule of the generated mobile destination candidate.

Neighborhood generation is a mechanism / operation that creates the next search point, and updating search point is a mechanism / operation that selects search information. The neighborhood generation and the updating search point affect each other, and the search of metaheuristics is performed by repeating these two operations. In metaheuristics, each search point or the entire search point group holds and stores useful information (best solution information, search history, etc.) obtained so far, and aims to obtain a good solution. In other words, the search process in the direction and area where the solution is improved by using the useful information obtained by the search, is repeating to find a better solution. The metaheuristic search structure is shown in Fig. 3.1. In this paper, we consider that these operations are used to analyze the search structure of metaheuristics.

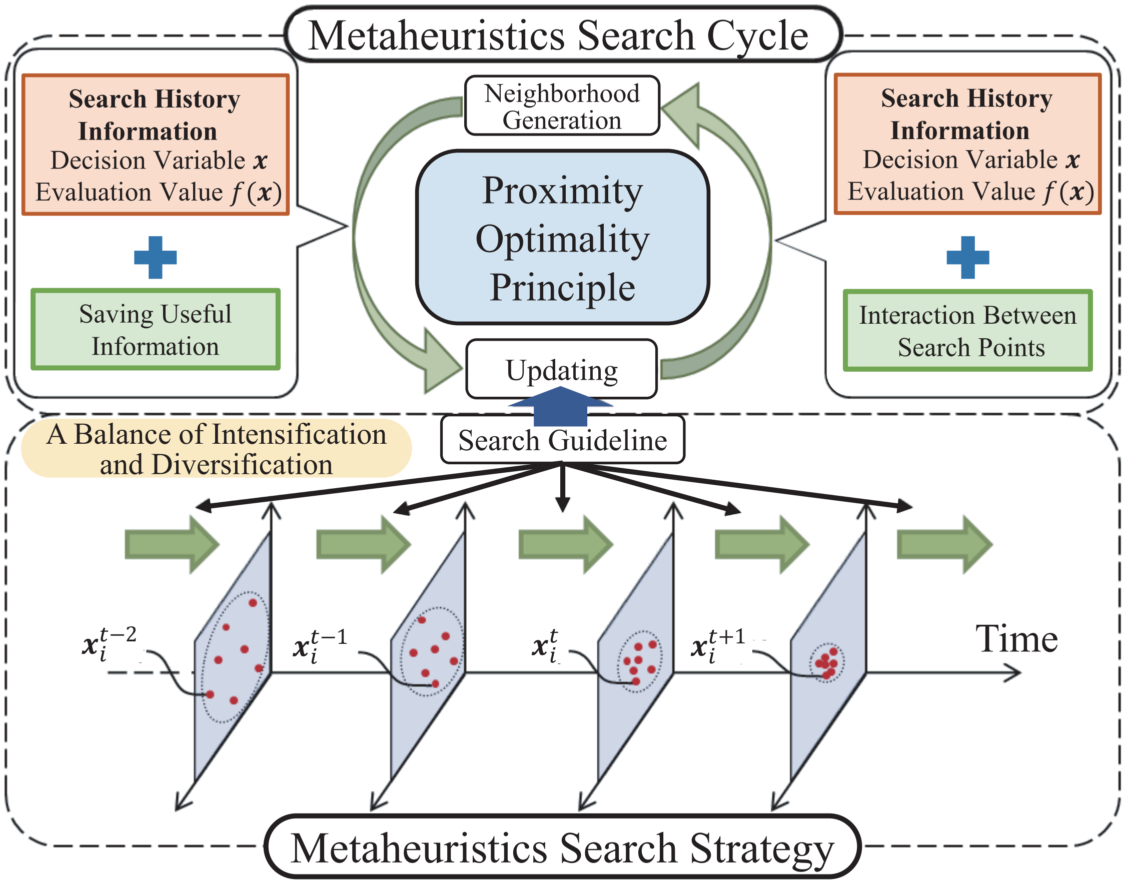


Fig. 3.1: Metaheuristics Search Cycle and Search Structure

3.2.3 Metaheuristics Search Strategy

Bias structure is known to exist in optimization problems in the field of engineering. Many metaheuristics use the bias solution structure in the algorithm. The proximity optimality principle (POP) [19][22] is a vague principle based on the experience that "good solutions have some similar structure." For many optimization problems, it is very likely that a better solution can be efficiently searched for using a good solution with a similar structure. Here, "good solution" and "similar structure" of POP are interpreted as follows.

Good solution: a solution with good evaluated values.

Similar structure: distance between solutions is short.

Furthermore, diversification and intensification are always applied as search guidelines for metaheuristics that utilize the effects of the proximity optimization principle.

3.2.4 Diversification and Intensification

In metaheuristics, there is diversification and intensification [19, 22, 23, 53] as a search guideline for effectively performing the cycle of neighborhood generation and selection. Diversification and intensification are an important guideline for realizing an ideal search process. Intensification has a strong meaning of convergence that collects search points, and diversification has a meaning of convergence suppression that does not collect search points more than necessary or divergence that separates search points. It is an important issue that maintains an appropriate balance between diversification and intensification to improve the search performance of metaheuristics.

In this paper, in order to reduce the influence between mechanisms and operations, we independently develop the mechanisms and operations for diversification and intensification. By independently providing the function of diversification and intensification, it is possible to easily formulate guidelines for subsequent diversification and intensification. Fig. **??** shows the metaheuristic cycle and the basic concept of diversification and intensification. The important functions in the optimization search are roughly divided into "wide area search" and "narrow area search". Interpret diversification and intensification from each function of wide area search and narrow area search.

- Intensification is a search guideline that aims to strongly advance the search in a good direction / region based on the proximity optimality principle to improve the short-term solution in a narrow range of the solution space, that is, to perform a local search.
- Diversification is a search guideline that aims to avoid the search from staying in

a local region / space and incorporat more information into the search in a wide space to improve the long-term solution, that is, to perform a global search.

In wide-area search, a mechanism for searching in a wide range is constructed, especially from the viewpoint of suppressing intensification. The main means of concentrating suppression in existing metaheuristics is to suppress the movement of the search point in a good direction / region by adjusting parameters. On the other hand, in narrow-area search, a mechanism for searching in a narrow range is constructed, especially from the viewpoint of suppressing diversification. Diverse suppression in existing metaheuristics is a means to contribute to the discovery of better solutions by mutual search using the good solution information found by the search by adjusting parameters. The distribution of search points is important for diversification and intensification. The process from diversification and intensification so far can be thought of as changing the search range of the search point from a wide range to a narrow range. The necessary conditions for an excellent search structure from the viewpoint of diversification and intensification extracted through the above analysis are shown below.

- Directivity for a specific area / direction. Furthermore, the adjustment of its directivity in the search process. In particular, by setting a good solution for a specific area, it becomes a search that effectively utilizes the proximity optimization principle.
- Region of perturbation / search point distribution. In addition, the adjustment of that area during the search process. In particular, by narrowing the area to a good solution, it becomes a search that effectively utilizes the proximity optimization principle.
- Use of random numbers. The role of alleviating excessive "directivity toward a specific area / direction" and "reduction of perturbation / search point distribution" in the search process.

From the above, it is considered that the search structure of metaheuristics is embedded

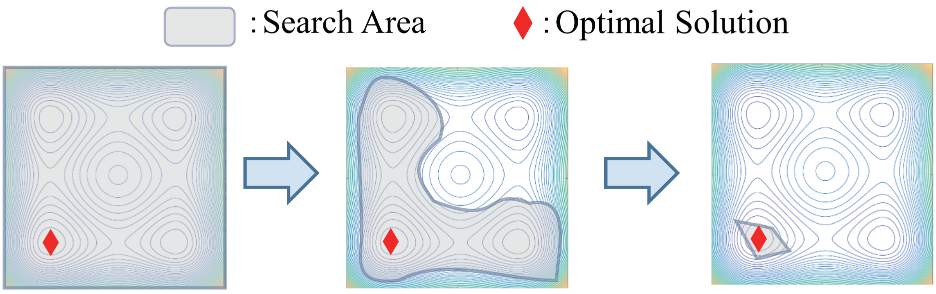


Fig. 3.2: Image of Metaheuristics Search Strategy of Diversification and Intensification

with operations for realizing diversification and intensification, especially in the neighborhood generation. And image of metaheuristics search strategy of diversification and intensification shows in Fig.3.2.

Furthermore, in excellent metaheuristics, it is considered that each operation embedded in the search structure has the ability to realize the search strategy by adjusting the realization state of diversification and centralization in the search process. On the other hand, it is considered that some methods do not have the ability to adjust diversification and intensification, although the operations for realizing diversification and intensification are embedded. Therefore, in order to improve the search performance of metaheuristics and realize high search performance, it is extremely important what kind of relationship does the search structure of metaheuristics, especially the generation of neighborhoods, have with the search strategy based on diversification and intensification.

3.3 Particle Swarm Optimization

3.3.1 Overview of Particle Swarm Optimization

Particle Swarm Optimization (PSO) [27, 28, 29, 30] is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. PSO is originally attributed to Kennedy, Eberhart and Shi [28] and was first intended for simulating social behaviour [29], as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization. The book by Kennedy and Eberhart describes many philosophical aspects of PSO and swarm intelligence. An extensive survey of PSO applications is made by Poli [30]. Recently, a comprehensive review on theoretical and experimental works on PSO has been published by Bonyadi and Michalewicz.

PSO is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position, but is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

3.3.2 Algorithm of Particle Swarm Optimization

In this Section, in order to outline each operation of PSO, we set the number of search points $m \in \mathbb{N}^{\mathbf{1}}$, i-th search point $\boldsymbol{x}_i \in \mathbb{R}^N$ $(i = 1, 2, \cdots, m)$, and the number of iterations $t \in \mathbb{N}^{\mathbf{1}}$.

First, in t-th iteration, the particle's best known position $\boldsymbol{pbest}_i^t$ and the swarm's best known position $\boldsymbol{gbest}^t$ is defined by the Eqs.(3.1) and (3.2).

$$\boldsymbol{pbest}_i^t = \begin{cases} \boldsymbol{pbest}_i^t & f(\boldsymbol{pbest}_i^t) < f(\boldsymbol{x}_i^{t-1}) \\ \boldsymbol{x}_i^{t-1} & \text{otherwise} \end{cases} \tag{3.1}$$

$$\boldsymbol{gbest}^t = \boldsymbol{pbest}_{i_g}^t,\ i_g = \arg\min_i f(\boldsymbol{pbest}_i{}^t) \tag{3.2}$$

From the current particle's position $\boldsymbol{x}_i^t$, each the particle's position refers to the particle's best known position $\boldsymbol{pbest}_i^t$ and the swarm's best known position $\boldsymbol{gbest}^t$ to generate the particle's velocity $\boldsymbol{v}_i^{t+1}$ by the Eq.(3.3).

$$\boldsymbol{v}_i^{t+1} = w \cdot \boldsymbol{v}_i^t + c_1 \cdot rand_1 \cdot (\boldsymbol{pbest}_i^t - \boldsymbol{x}_i^t) + c_2 \cdot rand_2 \cdot (\boldsymbol{gbest}^t - \boldsymbol{x}_i^t) \tag{3.3}$$

However, $rand_1$ and $rand_2$ are uniform random numbers of $[0, 1] \in \mathbb{R}^1$, and w, c_1, and c_2 are weight parameters for each argument.

Update the particle's position by the Eq.(3.4).

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1} \tag{3.4}$$

Particle Swarm Optimization for the minimization problem of the objective function $f(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^N$) is shown by **Algorithm 3.1**.

Algorithm 3.1 Particle Swarm Optimization（PSO）

procedure PSO$(m, w, c_1, c_2, T_{\max})$

Step 1: Preparation

Set the maximum number of iterations $T_{\max}$, the number of particles $2 \leq m \in \mathbb{N}^1$, particle parameters $0 < w \in \mathbb{R}^1,\ 0 < c_1 \in \mathbb{R}^1,\ 0 < c_2 \in \mathbb{R}^1$, and let $t = 1$.

Step 2: Initialization

Set initial position $\boldsymbol{x}_i^1 \in \mathbb{R}^N\ (i = 1, 2, \cdots, m)$ and initial velocity $\boldsymbol{v}_i^1 \in \mathbb{R}^N\ (i = 1, 2, \cdots, m)$ of each particle randomly in the feasible region.

for $i = 1$ to m **do**

$\boldsymbol{pbest_i}^1 = \boldsymbol{x_i}^1,\ i = 1, 2, \cdots, m$

$\boldsymbol{gbest}^1 = \boldsymbol{pbest}_{i_g}^1$

Where $i_g = \arg\min_i f(\boldsymbol{pbest}_i^1)$.

end for

Step 3: Updating $\boldsymbol{v}_i$ and $\boldsymbol{x}_i$

for $i = 1$ to m **do**

$\boldsymbol{v}_i^{t+1} = w \cdot \boldsymbol{v}_i^t + c_1 \cdot rand_1 \cdot (\boldsymbol{pbest}_i^t - \boldsymbol{x}_i^t) + c_2 \cdot rand_2 \cdot (\boldsymbol{gbest}^t - \boldsymbol{x}_i^t)$

$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1}$

end for

Step 4: Updating $\boldsymbol{pbest}_i$ and $\boldsymbol{gbest}_i$

Let $I = \{i \mid f(\boldsymbol{x}_i^{t+1}) < f(\boldsymbol{pbest}_i^t),\ i = 1, 2, \cdots, m\}$.

for $i = 1$ to m **do**

if $i \in I$ **then**

$\boldsymbol{pbest_i}^{t+1} = \boldsymbol{x_i}^{t+1}$

else

$\boldsymbol{pbest_i}^{t+1} = \boldsymbol{pbest_i}^t$

$\boldsymbol{gbest_i}^{t+1} = \boldsymbol{pbest_{i_g}}^t$

Where $i_g = \arg\min_i f(\boldsymbol{pbest_i}^{t+1})$.

end for

Step 7: Termination

if $t = T_{\max}$ **then**

The algorithm is terminated.

else

Return to **Step 3**, set $t := t + 1$.

end procedure

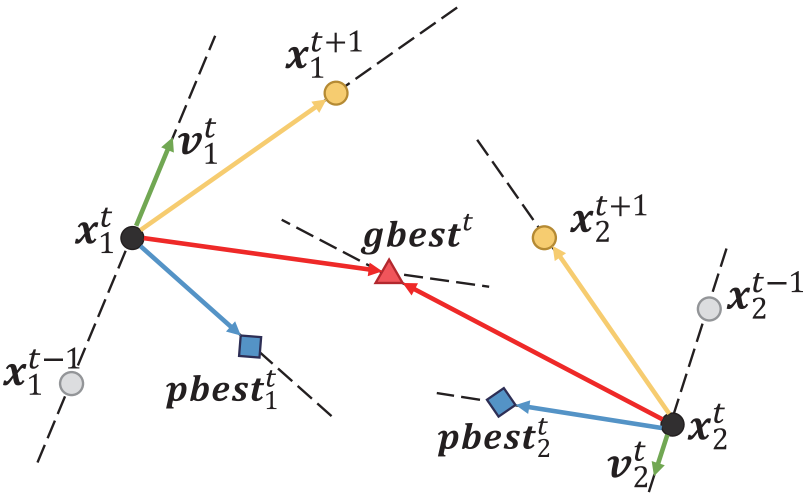


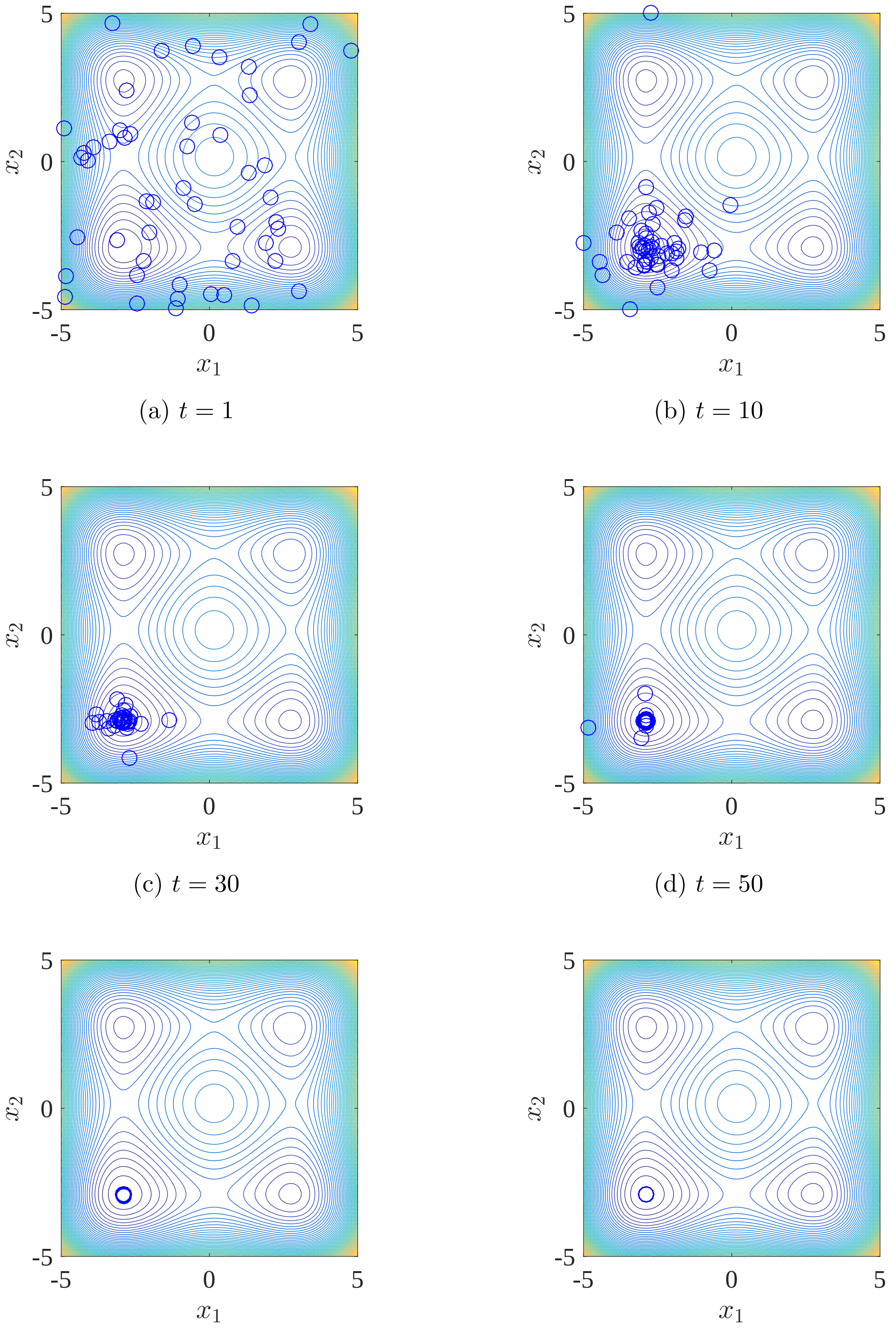
Fig. 3.3: Movement of the Search Point in Particle Swarm Optimization

3.3.3 Analysis of Particle Swarm Optimization Search Structure

In this paper, we consider the neighborhood generate of PSO that shows in Fig. 3.3 according to the update formula (see the Eq.(3.3). The moving vector $\boldsymbol{v}^{t+1}$ (the neighborhood generate) is represented as a linear combination of the moving vectors $\boldsymbol{v}^t$ and the difference vectors from $\boldsymbol{x}_i^t$ to $\boldsymbol{pbest}_i^t$ and $\boldsymbol{gbest}^t$.

Also, because PSO is an absolute movement, the search point must be updated to the neighborhood solution. The search point $\boldsymbol{x}_i^t$ does not always move in the direction of improvement, but the Eqs.(3.1) and (3.2) mean that $\boldsymbol{pbest}$ and $\boldsymbol{gbest}$ mean that always move in the direction of improvement. In other words, since the weak descent condition is satisfied for $\boldsymbol{pbest}$ and $\boldsymbol{gbest}$, the search point swarm moves in the descent direction.

Fig. 3.4 shows the each search state when PSO is applied to the 2^N minima function. In the Fig. 3.4, $\circ$ represents each search point. It can be confirmed that the PSO search point swarm is finally concentrated in one promising area. From the above analysis, it can be confirmed that PSO is suitable for the single-objective optimization problem.



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

(c) $t = 30$ (d) $t = 50$

(e) $t = 100$ (f) $t = 500$

Fig. 3.4: Transition of Search by Particle Swarm Optimization ($w = 0.729, c_1 = c_2 = 1.4955$)

3.4 Differential Evolution

3.4.1 Overview of Differential Evolution

Differential Evolution (DE) [34, 35] is one of the Evolutionary Algorithm [33] devised by K. Price and R. Storn in 1995. It is similar to the basic operation of the evolutionary optimization method, the search is performed by repeating differential mutation, crossover, and selection.

3.4.2 Algorithm of Differential Evolution

This Section outlines each operation of DE and describes the algorithm of DE. DE searches by three evolutionary operations: mutation, crossover, and selection [32]. Since there are multiple types of DE depending on the method of differential mutation and crossover as follows:

$$\mathrm{DE}/Base/Numpair/Crossover$$

$Base$ indicates how to select a basic individual at the time of differential mutation, $Numpair$ indicates the number of individual pairs selected at the time of difference, and $Crossover$ indicates the crossover method. In general, binomial crossover is written as Bin, and exponential crossover is written as Exp.

The number of individuals in the solution group is m, the number of generations is t, the number of the solution individual is i, and the individual vector is $\boldsymbol{x}_i^t$ $(i = 1, 2, \ldots m)$.

First, the mutation vector $\boldsymbol{v}_i^t$ is generated by each individual vector $\boldsymbol{x}_i^t$. Several generation methods of the mutation vector $\boldsymbol{v}_i^t$ have already been proposed are shown below. One of generation example is shown in the Fig. 3.5.

- DE/rand/1

$$\boldsymbol{v}_i^t = \boldsymbol{x}_{\mathrm{r}_1}^t + F \cdot (\boldsymbol{x}_{\mathrm{r}_2}^t - \boldsymbol{x}_{\mathrm{r}_3}^t) \tag{3.5}$$

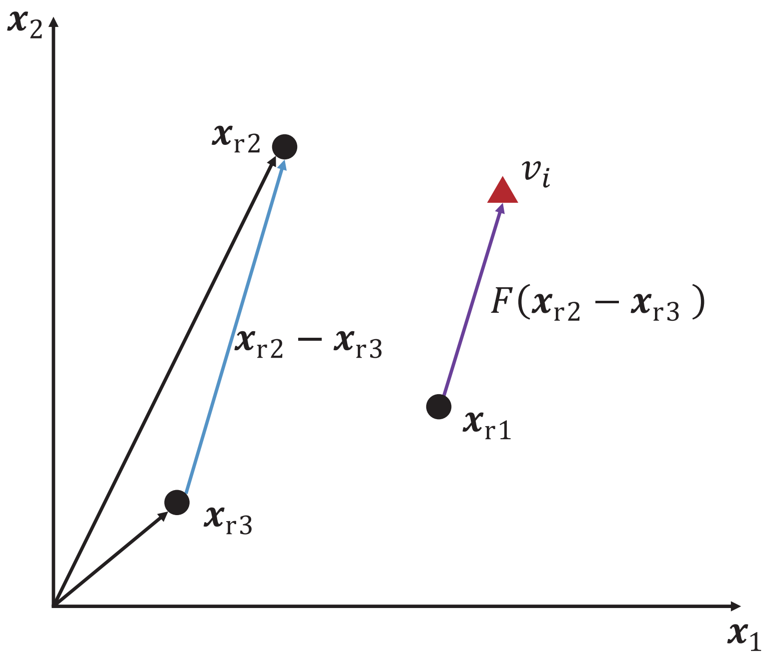


Fig. 3.5: Example of Differential Mutation Generation

- DE/best/1

$$\boldsymbol{v}_i^t = \boldsymbol{x}_{\text{best}}^t + F \cdot (\boldsymbol{x}_{\text{r}_1}^t - \boldsymbol{x}_{\text{r}_2}^t) \tag{3.6}$$

- DE/rand/2

$$\boldsymbol{v}_i^t = \boldsymbol{x}_{\text{r}_1}^t + F \cdot (\boldsymbol{x}_{\text{r}_2}^t - \boldsymbol{x}_{\text{r}_3}^t + \boldsymbol{x}_{\text{r}_4}^t - \boldsymbol{x}_{\text{r}_5}^t) \tag{3.7}$$

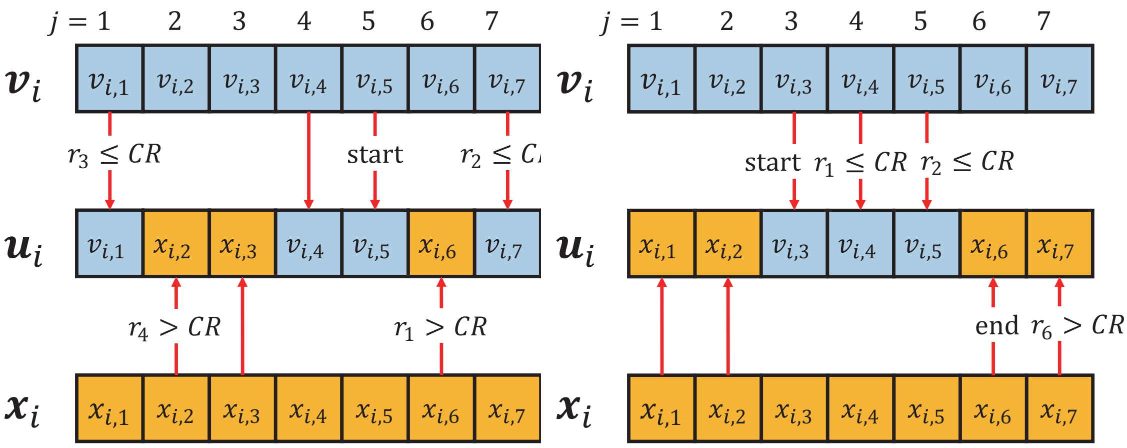
- DE/best/2

$$\boldsymbol{v}_i^t = \boldsymbol{x}_{\text{best}}^t + F \cdot (\boldsymbol{x}_{\text{r}_1}^t - \boldsymbol{x}_{\text{r}_2}^t + \boldsymbol{x}_{\text{r}_3}^t - \boldsymbol{x}_{\text{r}_4}^t) \tag{3.8}$$

- DE/rand-to-best/1

$$\boldsymbol{v}_i^t = \boldsymbol{x}_i^t + F \cdot (\boldsymbol{x}_i^t - \boldsymbol{x}_{\text{best}}^t + \boldsymbol{x}_{\text{r}_1}^t - \boldsymbol{x}_{\text{r}_2}^t) \tag{3.9}$$

Here, F is called a scale factor, which is a parameter that actually determines the region where the mutation vector is generated. The size of $F \in [0, 1]$ is often set to a real number [34]. The above $\boldsymbol{x}_{\text{best}}^t$ is an individual vector with the best evaluation



(a) Binomial Crossover (b) Exponential Crossover

Fig. 3.6: Example of Crossover Generation

value in the solution group of the t-th generation, and $\boldsymbol{x}_{r_k}^t$ ($k \in \{1, 2, 3, 4, 5\}$) is a different individual vector randomly selected from the solution population, except for the reference vectors $\boldsymbol{x}_i^t$ and $\boldsymbol{x}_{\text{best}}^t$.

Generate test vector $\boldsymbol{u}_i^t$ by replacing the elements of individual vector $\boldsymbol{x}_i^t$ and the mutation vector $\boldsymbol{v}_i^t$ at the crossover generation. An example of crossover generation is shown in the Fig. 3.6. Several crossover generations have already been proposed, but two typical crossover generations are shown below.

- Binomial Crossover

$$u_{i,j}^t = \begin{cases} v_{i,j}^t & \text{rand}_j \leq CR \quad \text{or} \quad j = n \\ x_{i,j}^t & \text{otherwise} \end{cases} \tag{3.10}$$

- Exponential Crossover

$$u_{i,j}^t = \begin{cases} v_{i,j}^t & j = \langle n \rangle_N, ..., \langle n + l \rangle_N \\ x_{i,j}^t & \text{otherwise} \end{cases} \tag{3.11}$$

Here, CR is a parameter of the crossover probability,which determines the number to replace the element of $\boldsymbol{x}_i^t$ with the element of $\boldsymbol{v}_i^t$. CR is set to a real number in the range

$[0, 1] \in \mathbb{R}^1$ like F. In binomial crossover, from the crossover start point n, depending on the magnitude of the crossover probability CR, determine the $x^t_{i,j}$ $(j = 1, 2, \cdots, N)$ element of $\boldsymbol{x}^t_i$ to replace with the element of $\boldsymbol{v}^t_i$. On the other hand, exponential crossover starts from the crossover start point n when CR satisfies a condition larger than a uniform random number to replace the consecutive l elements of the $\boldsymbol{x}^t_i$ elements with the $\boldsymbol{v}^t_i$ elements. However, the probability $Pr(k)$ that the elements up to the $n + k$ th are replaced is given by the Eq.(3.12).

$$Pr(k = l) = CR^k \tag{3.12}$$

In the final selection, the crossover generated test vector $\boldsymbol{u}^t_i$ is compared to the solution. The next-generation individual vector $\boldsymbol{x}^{t+1}_i$ is determined by the following equation, and the individual vector group is updated by the Eq.(3.13).

$$\boldsymbol{x}^{t+1}_i = \begin{cases} \boldsymbol{u}^t_i & f(\boldsymbol{u}^t_i) \leq f(\boldsymbol{x}^t_i) \\ \boldsymbol{x}^t_i & \text{otherwise} \end{cases} \tag{3.13}$$

The above is an overview of the three major evolutionary operations that make up DE.

DE/rand/1/Bin for the minimization problem of the objective function $f(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathbb{R}^N)$ is shown by **Algorithm 3.2**.

Algorithm 3.2 Differential Evolution（DE/rand/1/Bin）

procedure DE/RAND/1/BIN(m, F, CR, T_{max})

Step 1: Preparation

Set the number of dimensions of solution vector $N \in \mathbb{N}^1$, the number of solutions $2 \leq m \in \mathbb{N}^1$, scale factor $F \in \mathbb{R}^1$, crossover probability $CR \in \mathbb{R}^1$ and the maximum generation $T_{\max} \in \mathbb{N}^1$.

Step 2: Initialization

Set the number of iterations $t = 1$.

In the feasible area, randomly generate solution $\boldsymbol{x}_i^1$ $(i = 1, 2, \cdots, m)$

Step 3: Mutation

for $i = 1$ to m **do**

Randomly selecte solutions $\boldsymbol{x}_{r1}^t,\ \boldsymbol{x}_{r2}^t,\ \boldsymbol{x}_{r3}^t \in \mathbb{R}^N$ which are different from each other.

Use scale factor F to generate the mutation vector $\boldsymbol{v}_i^t \in \mathbb{R}^N$ in the following equation.

$\boldsymbol{v}_i^t := \boldsymbol{x}_{r1}^t + F \cdot (\boldsymbol{x}_{r2}^t - \boldsymbol{x}_{r3}^t)$

end for

Step 4: Crossover

for $i = 1$ to m **do**

Determine the crossover start point n, and the crossover determination is performed using the crossover probability CR for each of the operation vector elements x_{ij} $(j = 1, 2, \cdots, N)$.

The elements of x_{ij}^t and v_{ij}^t are replaced as in the following equation, and generate the test solution $\boldsymbol{u}_i^t \in \mathbb{R}^N$.

$$u_{ij}^t := \begin{cases} v_{ij}^t & rand_j \leq CR \text{ or } j = n \\ x_{ij}^t & \text{otherwise} \end{cases}$$

end for

Step 5: Selection

Each solution $\boldsymbol{x}_i^t$ is compared with test solution $\boldsymbol{u}_i^t$ to update solutions in the following equation.

for $i = 1$ to m **do**

$$\boldsymbol{x}_i^{t+1} := \begin{cases} \boldsymbol{u}_i^t & f(\boldsymbol{u}_i^t) \leq f(\boldsymbol{x}_i^t) \\ \boldsymbol{x}_i^t & \text{otherwise} \end{cases}$$

end for

Step 6: Termination

if $t = T_{\max}$ **then**

The algorithm is terminated.

else

Return to **Step 2**, set $t := t + 1$.

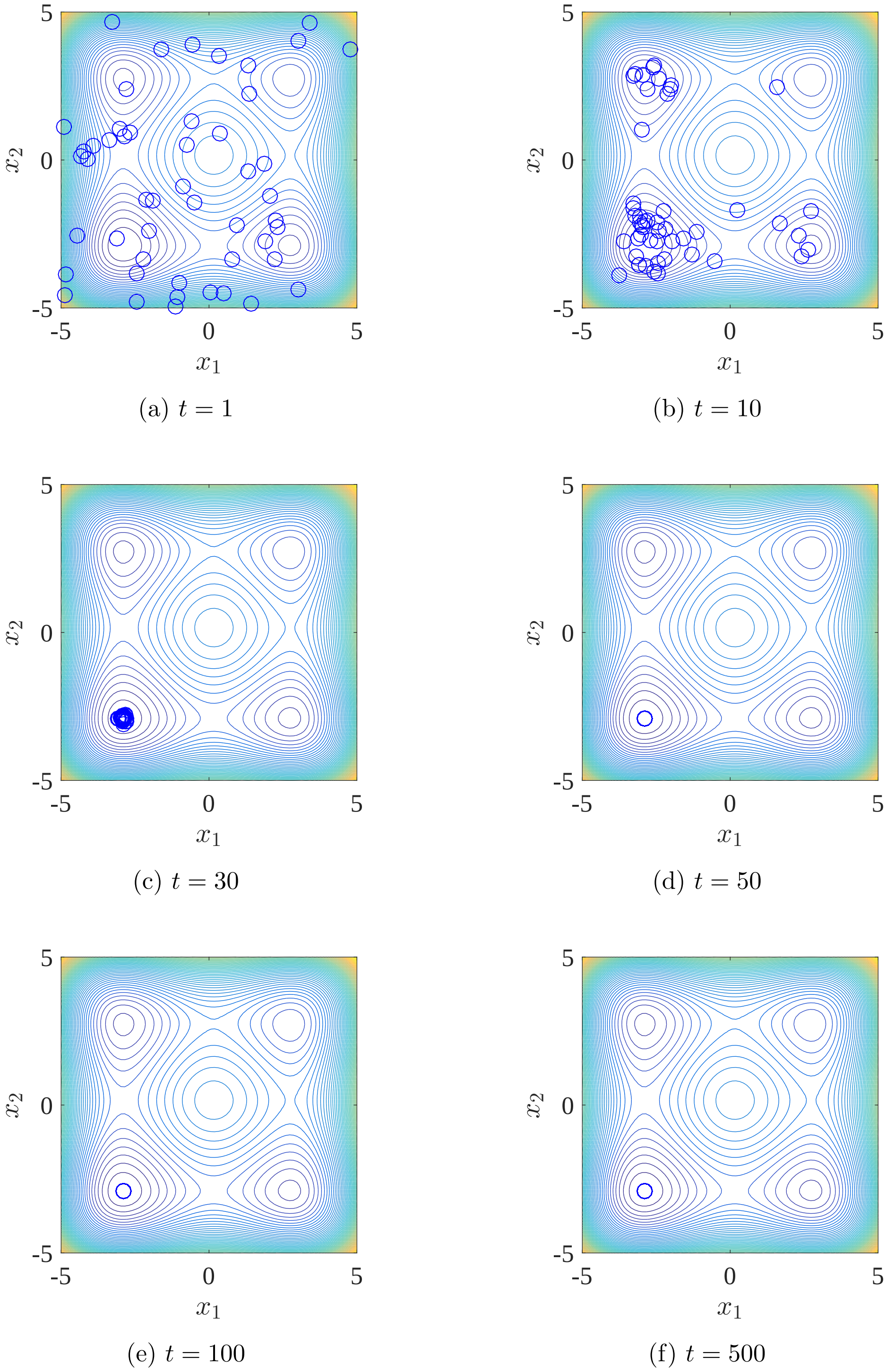
end procedure

3.4.3 Analysis of Differential Evolution Search Structure

Focus on the neighborhood generate consists of the mutation (see the Eq.(3.5)) and the crossover (see the Eq.(3.10)) according to the update formula. In the mutation, the solution $\boldsymbol{v}_i$ is generated by the difference vector between the search points $\boldsymbol{x}_{\mathrm{r}2}$ and $\boldsymbol{x}_{\mathrm{r}3}$ with the search point $\boldsymbol{x}_{\mathrm{r}1}$ as the reference. In the crossover, the neighborhood solution $\boldsymbol{u}_i$ is generated by combining the search point $\boldsymbol{x}_i$ and the solution $\boldsymbol{v}_i$ for each element.

Since the mutation generates a solution $\boldsymbol{v}_i$ from a excellent search point swarm, it is highly possible that the solution $\boldsymbol{v}_i$ is excellent. Furthermore, in crossover, by combining the excellent search point $\boldsymbol{x}_i$ and the solution $\boldsymbol{v}_i$, the neighborhood solution $\boldsymbol{u}_i$ is generated in an even better region. Also since DE is an improvement movement, the search point $\boldsymbol{x}_i$ updates when the objective function value improves. In other words, movement of DE is performed only when each search point $\boldsymbol{x}_i$ satisfies the descent condition, and the excellent search point saves to the next search point swarm. Therefore, DE moves the search point $\boldsymbol{x}_i$ that satisfies the descent condition and utilizes the difference vector between the search points to generate an excellent neighborhood, which gathers in a good area (one place).

Fig. 3.7 shows the each search state when DE is applied to the 2^N minima function. In the Fig. 3.7, $\circ$ represents each search point. It can be confirmed that the DE search point swarm is finally concentrated in one promising area. From the above analysis, it can be confirmed that DE is suitable for the single-objective optimization problem.



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

(c) $t = 30$ (d) $t = 50$

(e) $t = 100$ (f) $t = 500$

Fig. 3.7: Transition of Search by Differential Evolution ($F = 0.5, CR = 0.5$)

3.5 Artificial Bee Colony Algorithm

3.5.1 Overview of Artificial Bee Colony Algorithm

The Artificial Bee Colony (ABC) Algorithm [36, 37] proposed by Karaboga is a metaheuristic with an analogy to the foraging behavior of honey bees. In a natural bee swarm, there are three kinds of honey bees to search foods generally, which include the employed bees, the onlookers, and the scouts.

The employed bees: The employed bees search the food around the food source in their memory, meanwhile they pass their food information to the onlookers.

The onlookers : The onlookers tend to select good food sources from those founded by the employed bees, then further search the foods around the selected food source.

The scouts : The scouts are translated from a few employed bees, which abandon their food sources and search new ones.

In a word, the food search of bees is collectively performed by the employed bees, the onlookers, and the scouts.

3.5.2 Algorithm of Artificial Bee Colony Algorithm

In this Section, in order to outline each operation of ABC algorithm, we set the number of the employed bees $n_e \in \mathbb{N}^1$, the number of the onlookers $n_o \in \mathbb{N}^1$, the number of iteration $t \in \mathbb{N}^1$, and the number of dimensions $N \in \mathbb{N}^1$. In the ABC algorithm, the search is performed by repeating the three stages of "Employed bees stage", "Onlookers stage", and "Scouts stage". Therefore, the units of the basic ABC algorithm can be explained by each stage as follows:.

(a) Employed Bees Stage

Each employed bee is only related to one food source site. Therefore, the number of employed bees and food source sites are equal ($n_e = n_o$). The employed bee changes the location of the food source (solution) in her memory based on local information (visual information), finds a nearby food source and evaluates its quality. The movement rule of the i-th employed bee satisfies the Eq.(3.14).

$$v_{i,j} = \begin{cases} x_{i,j} + \phi(x_{i,j} - x_{k,j}) & j = h \\ x_{i,j} & \text{otherwise} \end{cases} \tag{3.14}$$

Here, $\phi \in [-1, 1] \in \mathbb{R}^1$ is a uniform random number, and k is a dimension number randomly selected from $\{1, 2, \ldots, N\}$. j is an individual number randomly selected from $\{1, 2, \ldots, n_e\}$ excluding i. The selection rule of the candidate solution $\boldsymbol{v}_i$ satisfies the Eq.(3.15).

$$\boldsymbol{x}_i := \begin{cases} \boldsymbol{v}_i, & f(\boldsymbol{v}_i) < f(\boldsymbol{x}_i) \\ \boldsymbol{x}_i, & \text{otherwise} \end{cases} \tag{3.15}$$

s_i is the number of times the position of the i-th individual was not updated by the search. If the evaluation value of the candidate solution deteriorates, do not update the position of the employed bee and add 1 to s_i as in the Eq.(3.16).

$$s_i := \begin{cases} s_i, & f(\boldsymbol{v}_i) < f(\boldsymbol{x}_i) \\ s_i + 1, & \text{otherwise} \end{cases} \tag{3.16}$$

(b) Onlookers Stage

After all the employed bees complete the search, they share information about the amount of nectar. The onlookers evaluate the nectar information obtained from all the employed bees and select a food source related to their nectar amount. This choice of probability depends on the fitness value of the solution in the population. In basic ABC algorithm, roulette wheel selection scheme in which each slice is proportional in size to the fitness value fit_i is employed by the Eq.(3.17).

$$fit_i := \begin{cases} \dfrac{1}{1+f(\boldsymbol{x}_i)}, & 0 \leq f(\boldsymbol{x}_i) \\ 1+|f(\boldsymbol{x}_i)|, & \text{otherwise} \end{cases} \tag{3.17}$$

Calculate the probability of being selected p_i that the employed bee is selected for roulette wheel selection by the Eq.(3.18).

$$p_i = \frac{fit_i}{\sum\limits_{n=1}^{m_e} fit_n} \tag{3.18}$$

The update of onlookers are used in the same way as the update of employed bees following to the Eqs.(3.14) and (3.15). If the onlooker does not update, the trial counter updates with the Eq.(3.16).

(c) Scouts Stage

In one cycle, the ABC algorithm checks if any source needs renew after all employed bees and onlookers have completed the search. In order to decide if a food source site needs renew, the ABC algorithm uses the counter s_i which is used during search process to count the stagnation number of food source site. If the value of the counter s_i is greater than the limit parameter $limit$, then initialize the counter $s_i = 0$ and renew food source site by the Eq.(3.19).

$$x_{i,j} = x_{\min} + \phi_{i,j}(x_{\max} - x_{\min}) \tag{3.19}$$

However, $x_{\max}$ is the maximum value of the domain of the search space, $x_{\min}$ is the minimum value of the domain, and $\phi \in [0,1] \in \mathbb{R}^1$ is a random uniform number uniform.

Artificial Bee Colony Algorithm for the minimization problem of the objective function $f(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^N$) is shown by **Algorithm 3.3**.

Algorithm 3.3 Artificial Bee Colony Algorithm (ABC Algorithm)

procedure ABC ALGORITHM($n_e, n_o, Limit, T_{max}$)

Step 1: Preparation

Set the number of employed bees $n_e > 0$, the number of onlooker bees $n_o > 0$, the limit count $limit > 0$, and the maximum number of iterations $T_{\max}$.

Step 2: Initialization

In the feasible area, randomly generate solution $\boldsymbol{x}_i \in \mathbb{R}^N$ $(i = 1, 2, \cdots, n_e)$

Let limit count $s_i = 0$ and iteration count $t = 1$.

Step 3: Searching of Employed Bee

For every individual $\boldsymbol{x}_i (i = 1, 2, \cdots, n_e)$, generate the neighborhood $\boldsymbol{v}_i (i = 1, 2, \cdots, n_e)$.

for $i = 1$ to n_e **do**

$$v_{i,j} = \begin{cases} x_{i,j} + \phi(x_{i,j} - x_{k,j}) & j = h \\ x_{i,j} & \text{otherwise} \end{cases}$$

Here, k is randomly selected from $\{1, 2, \cdots, n_e\}$ and h is randomly selected from $\{1, 2, \cdots, N\}$.

ϕ is a uniform random number of $[-1\ 1] \in \mathbb{R}^1$.

Evaluate the generated neighborhood and perform the following formulas.

$$\boldsymbol{x}_i := \begin{cases} \boldsymbol{v}_i & f(\boldsymbol{v}_i) < f(\boldsymbol{x}_i) \\ \boldsymbol{x}_i & \text{otherwise} \end{cases}$$

$$s_i := \begin{cases} 0 & f(\boldsymbol{v}_i) < f(\boldsymbol{x}_i) \\ s_i + 1 & \text{otherwise} \end{cases}$$

end for

Step 4: Searching of Onlooker Bee

Calculate the fitness fit_i of all individuals by the following formula.

$$fit_i := \begin{cases} \frac{1}{1+f(\boldsymbol{x}_i)} & 0 \leq f(\boldsymbol{x}_i) \\ 1 + |f(\boldsymbol{x}_i)| & \text{otherwise} \end{cases}$$

Use roulette selection based on fit_i to selecte individual for searching by the following formula.

for $i = 1$ to n_o **do**

$$v_{i,j} = \begin{cases} x_{i,j} + \phi(x_{i,j} - x_{k,j}) & j = h \\ x_{i,j} & \text{otherwise} \end{cases}$$

$$\boldsymbol{x}_i := \begin{cases} \boldsymbol{v}_i & f(\boldsymbol{v}_i) < f(\boldsymbol{x}_i) \\ \boldsymbol{x}_i & \text{otherwise} \end{cases}$$

$$s_i := \begin{cases} 0 & f(\boldsymbol{v}_i) < f(\boldsymbol{x}_i) \\ s_i + 1 & \text{otherwise} \end{cases}$$

end for

Step 5: Searching of Scout Bee

if $s_i \geq limit (i = 1, 2, \cdots, n_e)$ **then**

Move the individual randomly in the feasible area and set $s_i := 0$.

Step 6: Termination

if $t = T_{\max}$ **then**

The algorithm is terminated.

else

Return to **Step 3**, set $t := t + 1$.

end procedure

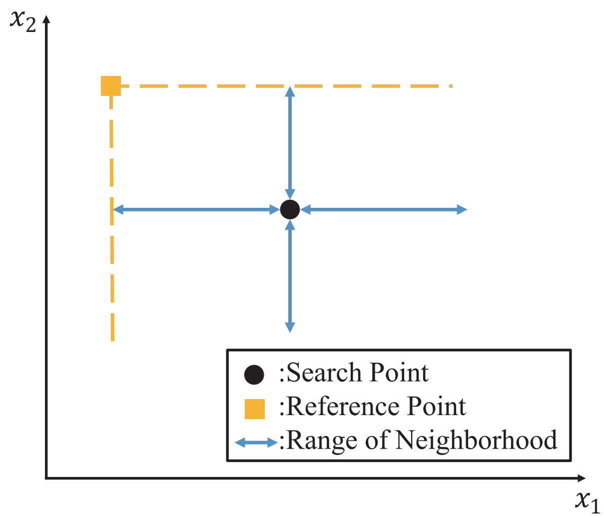
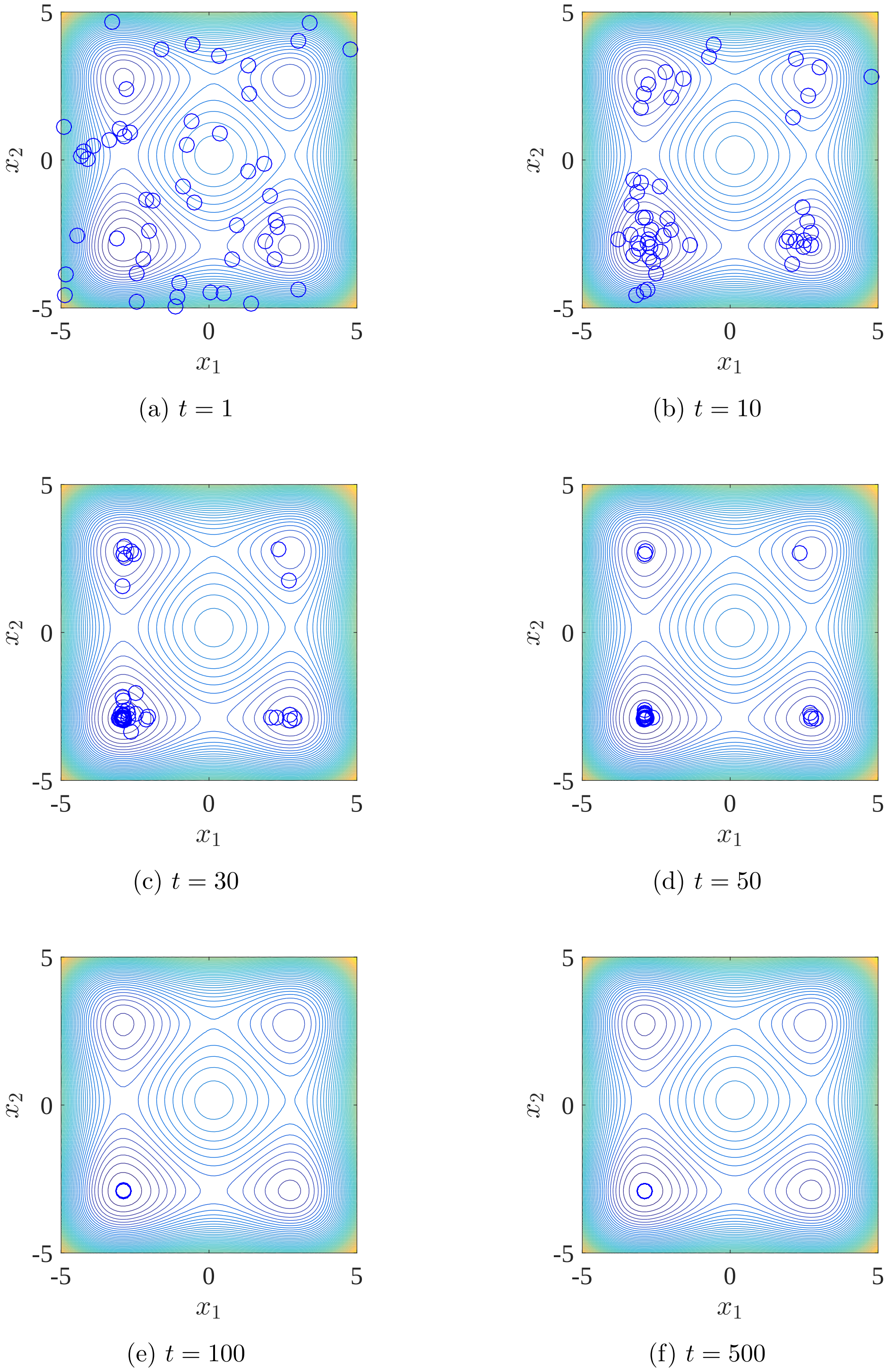


Fig. 3.8: Movement of the Search Point in Artificial Bee Colony Algorithm

3.5.3 Analysis of Artificial Bee Colony Algorithm Search Structure

Neighborhood generation of the ABC algorithm is done by changing only one element of the solution. After deciding the elements to change, select one reference vector from the other solutions and select one. Move the element closer to or further away from the reference vector (move to a random point on the arrow in the Fig. 3.8).

Since the ABC algorithm is an improvement movement, the search point $\boldsymbol{x}_i$ moves to the neighborhood solution $\boldsymbol{v}_i$ when the objective function value improves. In other words, the ABC algorithm moves only when each search point satisfies the descent condition, and saves the excellent search point to the next search point swarm. The neighborhood solution $\boldsymbol{v}_i$ can be expected to be generated in an excellent region because a perturbation along a single axis is applied by the excellent search point swarm. Therefore, the ABC algorithm moves the search point $\boldsymbol{x}_i$ that satisfies the descent condition and utilizes the difference vector between the search points to generate an excellent neighborhood, which gathers in a good area (one place).



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

(c) $t = 30$ (d) $t = 50$

(e) $t = 100$ (f) $t = 500$

Fig. 3.9: Transition of Search by Artificial Bee Colony Algorithm ($limit = 10$)

Fig. 3.9 shows the each state of the search when ABC algorithm is applied to the multimodal 2^N minima function. In the Fig. 3.9, $\circ$ represents each search point. It can be confirmed that the search point swarm of ABC algorithm is finally concentrated in one promising area. From the above analysis, it can be confirmed that ABC algorithm is suitable for the single-objective optimization problem.

3.6 Firefly Algorithm

3.6.1 Overview of Firefly Algorithm

Firefly Algorithm (FA) [38][39][40] is an optimization method based on the analogy of firefly activity. In actual firefly activity, each individual communicates with other fireflies by emitting light, feeding, and applying courtship behavior. Generally, bright firefly is attractive and each firefly is attracted to fireflies that emit bright light around themselves. Xin-She Yang proposed FA in 2007, which is a multi-point search type metaheuristic, and is abstracted as above phenomenon. Xin-She Yang idealized the courtship of fireflies in developing FA by the following three rules [40].

- All Fireflies are unisex so that any individual firefly will be attracted to all other fireflies.
- The attractiveness is proportional to their brightness, and for any two fireflies, the less bright one will be attracted to the brighter one. However, the intensity decreases as their mutual distance increases.
- If there are no fireflies brighter than a given firefly, it will move randomly.

In FA, each search point moves so as to approach another search point. At this time, FA determines the search point to be referred to and the movement amount to be referred to based on the light intensity (evaluation value of the search point) and attractiveness (coefficient for determining the movement amount).

First, in FA, each search point refers to a search point having a light intensity superior to itself and moves. The best search points move randomly. Next, each search point determines the amount of movement based on the appeal of the search point to be referenced. In actual firefly communication, the intensity of light is inversely proportional to the square of the distance, so far away fireflies can not be visually observed. Likewise, the attractiveness of search points in FA decreases with increasing distance between search points. And when moving, the stronger the attractiveness of the reference point, the greater the degree of movement.

3.6.2 Algorithm of Firefly Algorithm

In this Section, in order to outline each operation of FA, set the number of search points m, i-th search point $\boldsymbol{x}_i$ $(i = 1, 2, \cdots, m)$, the point group $\boldsymbol{P}^t = \{\boldsymbol{x}_i^t \mid i = 1, 2, \cdots, m\}$, and the number of iterations t. First, $\boldsymbol{x}_{cbest}^t$ is the search point having the best evaluation value in the search point group according to the Eq.(3.20).

$$\boldsymbol{x}_{cbest}^t = \arg \min_{\boldsymbol{x}_i^t \in \boldsymbol{P}^t} \{f(\boldsymbol{x}_i^t) \mid i = 1, 2, \cdots, m\} \tag{3.20}$$

Then, the reference point $\boldsymbol{z}_i$ and the reference point group $\boldsymbol{V}$ are saved according to the Eqs.(3.21) and (3.22).

$$\boldsymbol{z}_i = \boldsymbol{x}_i^t \ (i = 1, 2, \cdots, m) \tag{3.21}$$

$$\boldsymbol{V} = \{\boldsymbol{z}_i \mid i = 1, 2, \cdots, m\} \tag{3.22}$$

Each operation of FA will be described as follows. To grasp the light intensity and attractiveness quantitatively, we define the light intensity I_i of FA and attractiveness $\beta_{i,j}$ by the Eqs.(3.23) and (3.24).

$$I_i = \left(\left| f(\boldsymbol{x}_{cbest}^t) - f(\boldsymbol{x}_i^t) \right| + 1 \right)^{-1} \tag{3.23}$$

$$\beta_{i,j} = \beta_0 e^{-\gamma \left\| \boldsymbol{z}_j - \boldsymbol{x}_i^t \right\|^2} \tag{3.24}$$

Here β_0 is the maximum value of attraction, and γ is a parameter related to attenuation of light. Each firefly is attracted by all fireflies within stronger light intensity to move. The movement equation is defined by the Eq.(3.25), when firefly i is sucked into firefly j.

$$\boldsymbol{x}_i^t := \boldsymbol{x}_i^t + \beta_{i,j}(\boldsymbol{z}_j - \boldsymbol{x}_i^t) + \alpha \boldsymbol{R} \tag{3.25}$$

Here α is a parameter corresponding to the scale of the random number term and $\boldsymbol{R}$ represents a uniform random number vector to be changed in the range of $[-0.5, 0.5]^N$. The brightest firefly $\boldsymbol{x}_{cbest}^t$ moves randomly according to the following Eq.(3.26).

$$\boldsymbol{x}_{cbest}^t := \boldsymbol{x}_{cbest}^t + \alpha \boldsymbol{R} \tag{3.26}$$

Firefly Algorithm for the minimization problem of the objective function $f(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^N$) is shown by **Algorithm 3.4**.

Algorithm 3.4 Firefly Algorithm（FA）

procedure FA($\boldsymbol{x}, m, \alpha, \beta_0, \gamma, T_{\max}$)

Step 1: Preparation

Set the maximum number of iterations $T_{\max}$, the number of search points m, and the parameters $\alpha > 0$, $\beta_0 > 0$, and $\gamma > 0$.

Step 2: Initialization

Set the number of iterations $t = 1$.

In the feasible area $\boldsymbol{X} \subseteq \mathbb{R}^N$, randomly generate search point $\boldsymbol{x}_i^1$ $(i = 1, 2, \cdots, m)$, and the swarm set $\boldsymbol{P}^1 = \{\boldsymbol{x}_i^1 \mid i = 1, 2, \cdots, m\}$.

Step 3: Calculation of light intensity

Calculate the light intensity I_i of each search point $\boldsymbol{x}_i^t \in \boldsymbol{P}^t$.

$\boldsymbol{x}_{cbest}^t = \underset{\boldsymbol{x}_i^t \in \boldsymbol{P}^t}{\arg\min}\{f(\boldsymbol{x}_i^t) \mid i = 1, 2, \cdots, m\}$

$I_i = \left(|f(\boldsymbol{x}_{cbest}^t) - f(\boldsymbol{x}_i^t)| + 1\right)^{-1}$

Each search point $\boldsymbol{x}_i^t \in \boldsymbol{P}^t$ is sorted in descending order of I_i^t.

Save the reference solution $\boldsymbol{z}_i$ and solution set $\boldsymbol{V}$.

$\boldsymbol{z}_i = \boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, m)$, $\boldsymbol{V} = \{\boldsymbol{z}_i \mid i = 1, 2, \cdots, m\}$

Set $i = 1$ and $j = 1$.

Step 4: Movement of the search point

for $i = 1$ to m **do**

    if $I_i < I_j$ **then**

        $\boldsymbol{x}_i^t := \boldsymbol{x}_i^t + \beta_0 e^{-\gamma \|\boldsymbol{z}_j - \boldsymbol{x}_i^t\|^2}(\boldsymbol{z}_j - \boldsymbol{x}_i^t) + \alpha \boldsymbol{R}$

        Here, $\boldsymbol{R} \in [-0.5, 0.5]^N$ is a uniform random vector.

        Let $j := j + 1$.

        if $j = m$ **then**

            $\boldsymbol{x}_{cbest}^t := \boldsymbol{x}_{cbest}^t + \alpha \boldsymbol{R}$

            Set $j := 1$.

end for

Step 5: Updating search points

Update the search point $\boldsymbol{x}_i^t \in \boldsymbol{P}^t$ and the swarm set $\boldsymbol{P}^t$.

$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, m)$, $\boldsymbol{P}^{t+1} = \{\boldsymbol{x}_i^{t+1} \mid i = 1, 2, \cdots, m\}$

Step 6: Termination

if $t = T_{\max}$ **then**

    The algorithm is terminated.

else

    Return to **Step 3**, set $t := t + 1$.

end procedure

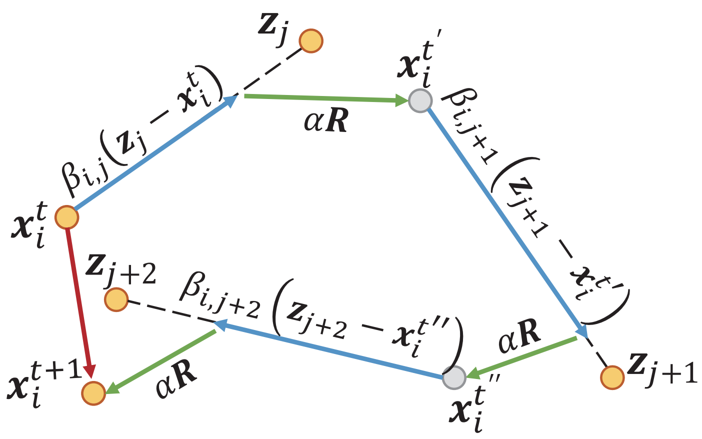


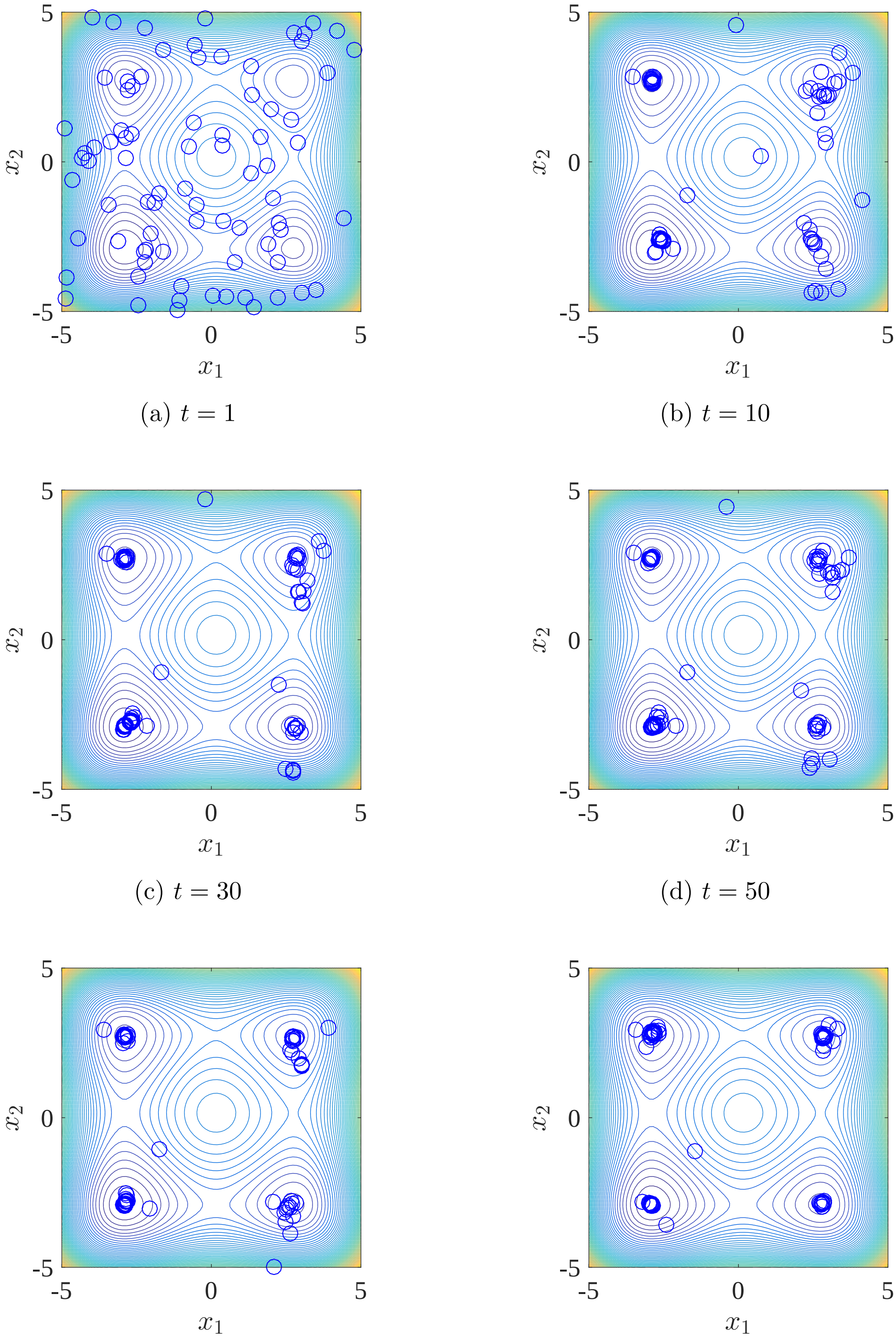
Fig. 3.10: Movement of the Search Point in Firefly Algorithm

3.6.3 Analysis of Firefly Algorithm Search Structure

Focus on the neighborhood generate according to the update formula shows in Fig. 3.10. As represented by the update formula of the Eq.(3.27), the neighborhood solution $\boldsymbol{x}_i^{t'}$ generates as a linear combination of the difference vector $\beta_{i,j}(\boldsymbol{z}_j - \boldsymbol{x}_i^t)$ and the uniform random number vector $\alpha\boldsymbol{R}$. Since FA is an absolute move, the search point $\boldsymbol{x}_i^t$ always moves to the neighborhood solution $\boldsymbol{x}_i^{t'}$.

$$\boldsymbol{x}_i^{t'} := \boldsymbol{x}_i^t + \beta_{i,j}(\boldsymbol{z}_j - \boldsymbol{x}_i^t) + \alpha\boldsymbol{R} \tag{3.27}$$

Furthermore, from the Eq.(3.24), if the distance between $\boldsymbol{x}_i^t$ and $\boldsymbol{z}_j$ is small, $\beta_{i,j}$ is large, and if the distance between $\boldsymbol{x}_i^t$ and $\boldsymbol{z}_j$ is large, $\beta_{i,j}$ is small. In other words, if the distance between $\boldsymbol{x}_i^t$ and $\boldsymbol{z}_j$ is extremely small, $\beta_{i,j} \simeq \beta_0$, when $\beta_0 = 1$ movement closer to $\boldsymbol{z}_j - \boldsymbol{x}_i^t$. On the other handm if the distance between $\boldsymbol{x}_i^t$ and $\boldsymbol{z}_j$ is extremely large, $\beta_{i,j} \simeq 0$, it has almost no effect on the movement of the first argument $\boldsymbol{x}_i^t$ and $\boldsymbol{z}_j$. In other words, the amount of movement is small when the distance between the search point and the reference point is long, and the amount of movement is large when the distance is short. Therefore, FA is a dynamic that gathers search points that are



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

(c) $t = 30$ (d) $t = 50$

(e) $t = 100$ (f) $t = 500$

Fig. 3.11: Transition of Search by Firefly Algorithm ($\beta_0 = 1, \gamma = 0.5, \alpha = 0.05$)

close to each other, and the search point swarm has the property of being divided into multiple swarms.

Fig. 3.11 shows the each state of the search when FA is applied to the multimodal 2^N minima function. In the Fig. 3.11, $\circ$ represents each search point. From the above analysis, it can be confirmed that FA is suitable for the superior solution set search problem.

3.7 Comparison of Each Method

3.7.1 Analyze the Characteristics of the Superior Solution Set Search Problem

In Chapter 2, We proposed the superior solution set search problem. We analyzed the structure and characteristics of the superior solution set search problem, the superior solution set is a diverse set of solutions with different properties between solutions. So we know the requirements of the superior solution set search method as follows:

(1) the method has a mechanism for searching multiple promising regions in parallel.

(2) the method is a multi-point search type and can apply multimodal structure optimization.

(3) the method has the search mechanism of the distance.

(4) the method can handle versatility for Black-Box optimization for practical optimization which changes in the surrounding environment.

Based on the above requirements, we analyze several representative algorithms and find out which methods are suitable for superior solution search.

3.7.2 Analysis of Each Method

All the methods introduced in this chapter meet the requirements of (2) and (4) above, but only FA is the only method that meets all requirements. Therefore, I think FA is most suitable for the development of superior solution set search problem.

In response to requirements (1) and (3), we conduct a specific analysis of the FA. From the updated formula of FA (see Eq.(3.24)), we can see that when the search point of FA is moving, the distance between the search point and the reference point must be considered. So FA meets the requirement (3), the method has the search mechanism of the distance.

At the same time, FA adjusts the magnitude of the difference vector according to the distance. When the distance is long, the magnitude of the difference vector becomes small and the search point does not approach the reference point. When the distance is short, the magnitude of the difference vector does not decrease and the search point approaches the reference point. So that, the search point group is divided into multiple groups, and each group searches for a local optimum solution. Therefore, FA also meets requirement (1), the method has a mechanism for searching multiple promising regions in parallel.

Since there is a high possibility that the superior solution can be obtained by finding multiple local optimum solutions, FA has excellent properties to search for the superior solution set. In the following Chapters 4 and 5, we have proposed two methods based on FA for the superior solution set search problem.

3.8 Summary

In this chapter, we compared typical metaheuristics such as Particle Swarm Optimization, Differential Evolution, Firefly Algorithm etc. and pointed out that Firefly Algorithm has a mechanism for adjusting the amount of movement based on the distance from the reference point. From this feature, it was clarified that Firefly Algorithm can obtain multiple local optimum solutions and has a basic method for searching for the superior solution set, which is a subset of the local optimum solution set.

4 Superior Solution Set Search Method Based on Clustering

4.1 Introduction

In this Chapter, from the analysis of the search structure of the Firefly Algorithm (FA), FA has the property of searching multiple promising regions in parallel. It was clarified that FA has a high affinity for the superoir solution set search problem. By taking advantage of this property and incorporating a cluster structure, we propose a parameter adjustment rule for FA. Then, assuming a basic case in the superior solution set search problem, a numerical experiment is performed for a benchmark function having multiple optimal solutions that are separated from each other. Compare the proposed method with the original FA and examine the usefulness of the proposed method.

I add an adaptive mechanism (diversification and intensification) to the above proposed method to further improve search performance. In addition, by analyzing the FA search dynamics from the perspective of diversification and intensification, we clarify the diversification and intensification of FA and the ability to adjust parameters. Based on the evaluation and control of the FA search state in the superior solution set search problem, an FA with high adaptability is constructed. Then, assuming a basic case in the superior solution set search problem, a numerical experiment is performed for a benchmark function having multiple optimal solutions that are separated from

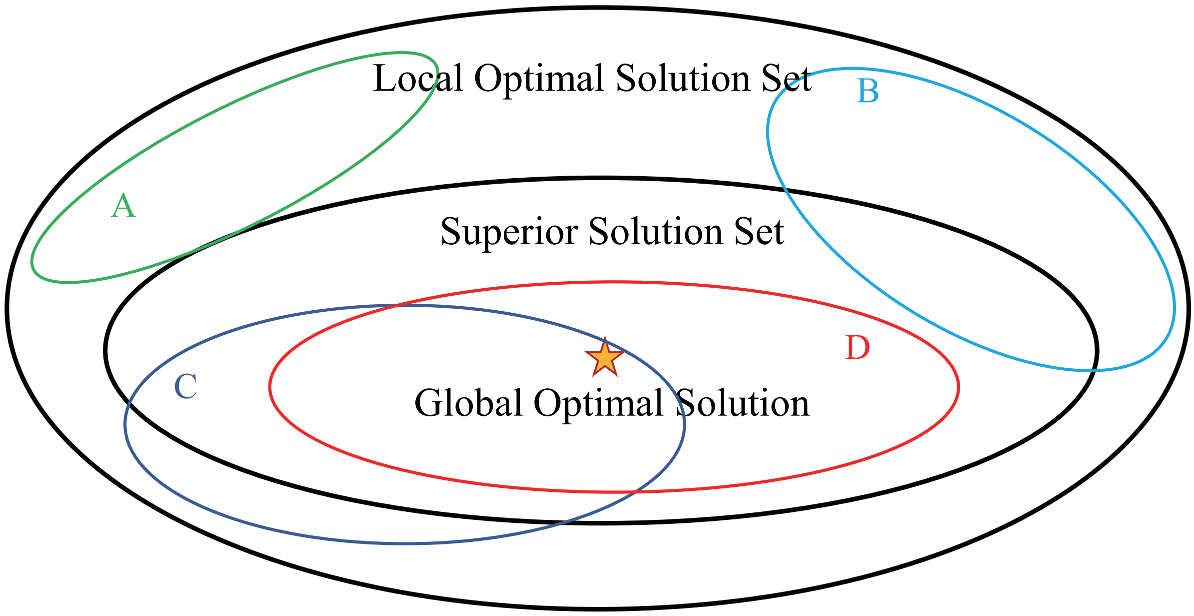


Fig. 4.1: Inclusion Relation of Global Optimal Solution, Superior Solution Set, and Local Optimal Solution Set

each other. We compare the original FA and FA based on cluster information with the adaptive FA, and examine the usefulness of the adaptive FA.

4.2 Research Approaches

The superior solution set is part of the local optimal solution set that contains the global optimal solution (see Fig. 4.1). Therefore, if we get all the local optimal solutions, we can get all superior solutions. However, it is difficult to obtain all local optimal solution sets in numerical calculation, so I obtain the superior solution set approximately by acquiring a part of the local optimal solution set.

Then, there are various cases when obtaining some solutions of the local optimal solution set in numerical calculation. For example, there is a case such as “A” that does not include the superior solution at all. There is a case such as “B” that includes some superior solutions, but does no include the global optimal solution. There is a case such as “C” that includes the global optimal solution and some superior solutions. There is a case such as “D” that includes only the global optimal solution and some superior solutions. Fig. 4.1 shows a schematic diagram of each case.

I developed two approaches, one directly acquires the superior solution, and the other indirectly acquires the superior solution. In order to search for the superior solution set in indirectly approach, it is necessary to find more local optimal solutions with higher optimization. In this way, there may be more superior solutions in obtaining the local optimal solutions. On the other hand, in directly approach, it is necessary to use parameters δ and ε, which are defined the superior solution set search problem, to clearly evaluate the search points, so that the obtained solution meets the constraints of the superior solution.

In this Chapter, the purpose is to efficiently obtain “C”, which is a part of the superior solution set, to indirectly acquire the superior solution. In next Chapter, the purpose is to efficiently obtain “D”, which is a part of the superior solution set, to directly acquire the superior solution.

4.3 Firefly Algorithm Based on Cluster Information

Many of the metaheuristics proposed so far aim to search for the only global optimal solution or quasi-optimal solution, and since each search point performs a global search. It is difficult to search multiple promising regions in parallel. On the other hand, a superior solution set is composed of a plurality of locally optimal solution sets in which the evaluation value of the objective function is superior by a certain value or more and the distances between the solutions are separated by a certain value or more. In order to search for a superior solution set, it is necessary to have the ability to search for promising regions in parallel.

The characteristics of FA are the reference of the solution based on the light intensity of the firefly and the movement by the difference vector toward the referenced solution. This amount of movement decreases as the distance between the search point and the referenced search point increases. In other words, the amount of movement is small when the distance between the moving search point and the reference point is long. On the other hand, when the distance is short, the amount of movement is large. FA has the properties of an algorithm for basic superior solution set search, in which the search point group is divided into multiple by these properties [88, 92, 94].

Also, at low dimensions, FA can simultaneously find multiple local optimal solutions in the search process. However, as dimensions increases, the ability of FA to search for superior solution set in parallel is weakening. Therefore, it is an important issue that the development of an algorithm to further improve low-dimensional search performance for the superior solution set search problem, and add the ability to search for high-dimensional superior solution set.

From the analysis in Chapter 3, compared to other metaheuristics, FA has the property that it is divided into multiple groups in the search process of the superior solution set in parallel. Therefore, unlike many single-purpose optimization methods, it is expected that the superior solution set can be searched efficiently by utilizing the properties of FA. On the other hand, there is still much room for improvement in applying FA to the search for the superior solution set. Further improvement in FA performance can be expected by adding a mechanism (cluster) for dividing into multiple clearer groups.

4.3.1 Analysis of Firefly Algorithm Parameter

In this Section, we focus on the following operations in the FA algorithm and analyze the search dynamics of FA.

- The neighborhood solution is generated from the difference vector by the Euclidean distance between the search point $\boldsymbol{x}_i$ and the reference point $\boldsymbol{z}_j$ and the perturbation by random number.

- The search point $\boldsymbol{x}_i$ is always updated to the generated neighborhood.

First, since the update of the search point of FA allows the deterioration of the search point, it is difficult to improve the search point only by perturbation of the random number term. On the other hand, the random number term in FA is considered to contribute to the local search after the search points have converged.

It is considered that the major contribution to the improvement of the FA search point is the movement of the update point $\boldsymbol{x}_i$ toward the better reference point $\boldsymbol{z}_j$. However, when this movement is small, efficient improvement cannot be made. On the other hand, when the movement is large, all the search points converge around the best solution without dividing into multiple groups. In order to solve the superior solution set search problem efficiently, it is necessary to properly adjust the movement toward the excellent reference point. In addition, in a search divided into multiple clusters, it is considered difficult to control all clusters uniformly because the search status for each cluster is different. FA performance can be expected to improve by making appropriate adjustments according to the status of each cluster.

FA converges to multiple local optimal solutions (subsets of superior solutions) because of the mechanism for adjusting the amount of movement based on the distance between the search point and the reference point. The parameter γ affects the mechanism of movement adjustment. If you set γ large, the amount of movement will decrease. If you set γ small, the amount of movement will increase.

I did experiments that change the parameter γ of FA to confirm the number of local optimal solutions obtained. So I changed γ from 0.05 to 1.0 in 20 runs of 0.05 for benchmark functions Function 1 (F_1), Function 2 (F_2), Function 3 (F_3), and Function 4 (F_4) are given in Table 4.1, as given in Table 4.2 $\sim$ Table 4.5. From the experimental results, I know that the adjustment of the parameter γ has a huge impact on the results. In FA, it is extremely important to set the parameter γ in order to realize a search divided into many clusters.

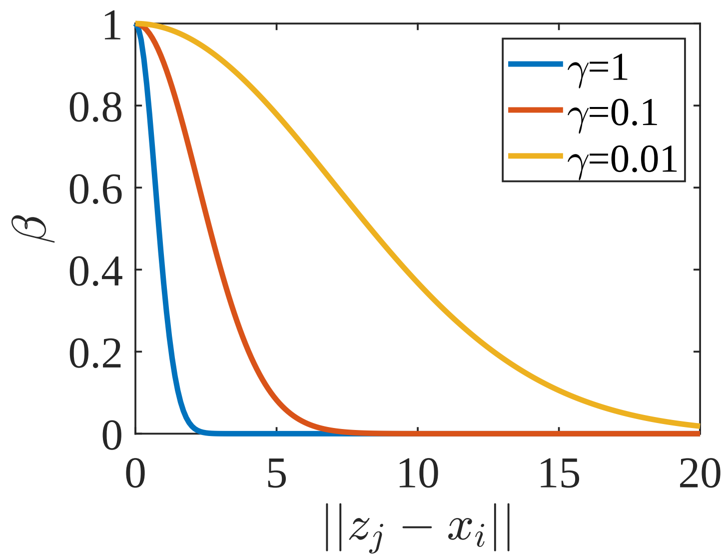


Fig. 4.2: Variations of β

4.3.2 Proposal of Parameter Adjustment Rule Based on Cluster Information

From the analysis so far, it can be seen that γ is a parameter that expresses the firefly analogy that is the basis of FA, and has a great influence on the performance of searching for the superior solution set. Furthermore, FA automatically divides into multiple clusters in the process of searching for the superior solution set. Therefore, we propose an adjustment rule for the parameter γ, which takes into account the property that FA divides into multiple clusters in the process of searching for the superior solution set. The proposed γ adjustment rule is shown in **Algorithm 4.1** below.

FA brings β closer to β_0 when it refers to a search point that is closer. On the other hand, FA divides the search point group into multiple points by bringing β closer to zero when referring to a search point that is far away. In this paper, the search point $\boldsymbol{x}_i^t (i = 1, 2, \cdots, m)$ is used as a clustering method to divide K clusters $\boldsymbol{U} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_K\}$. Here, various methods can be used to divide each cluster, but this time, the k-means clustering [62, 63, 64] (see **Algorithm B.1** in Appendix B), which is

Algorithm 4.1 Gamma Adjustment Rule

procedure GAMMA ADJUSTMENT RULE(m, K, C, P, β_0)
  Step 1: Preparation
  Set the number of clusters K and the parameters C and β_0.
  Give a set of search points $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m)$ and reference points $(\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_m)$.
  $\boldsymbol{z}_i = \boldsymbol{x}_i (i = 1, 2, \cdots, m)$
  Step 2: Cluster assignment
  Partition the search points $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m)$ into $K(\leq m)$ clusters $\boldsymbol{U} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_K\}$.
  $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_K\}$=$k$-MEANS ALGORITHM($\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m, K$) ▷ See Algorithm B.1
  Find the best point $\boldsymbol{Gbest}_k (k = 1, \cdots, K)$ of each cluster $\boldsymbol{U}_k (k = 1, 2, \cdots, K)$.
  $\boldsymbol{Gbest}_k = \underset{\boldsymbol{x}_l^t \in \boldsymbol{U}_k}{\arg\min}\{f(\boldsymbol{x}_i) | i = 1, 2, \cdots, |\boldsymbol{U}_k|\}$
  Step 3: Calculation of γ
  for $i = 1$ to m **do**
    Calculation parameter γ_i of search point $\boldsymbol{x}_i \in U_k$ by reference point $\boldsymbol{z}_j$.
    $\gamma_i = \begin{cases} -\ln(C^t/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2 & \boldsymbol{z}_j \in \boldsymbol{U}_k \\ -\ln(P^t/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2 & \boldsymbol{z}_j \notin \boldsymbol{U}_k \end{cases}$
    Here, k refers to the cluster number to which the search point $\boldsymbol{x}_i$ is assigned.
  end for
end procedure

easy to implement, is used. The Eq.(4.1) can be derived by transforming the Eq.(3.24). Variations of β is shown by Fig. 4.2.

$$\gamma = -\ln(\beta/\beta_0) / \left\|\boldsymbol{z}_j - \boldsymbol{x}_i^t\right\|^2 \tag{4.1}$$

Then, when the search point $\boldsymbol{x}_i^t \in \boldsymbol{U}_k$ refers to the reference point $\boldsymbol{z}_j \in \boldsymbol{U}_k$, the parameter C is used to adjust γ by replacing the reference point $\boldsymbol{z}_j \in \boldsymbol{U}_k$ with the best solution $\boldsymbol{Gbest}_k = \underset{\boldsymbol{x}_l^t \in \boldsymbol{U}_k}{\arg\min}\{f(\boldsymbol{x}_l^t) | l = 1, 2, \cdots, |\boldsymbol{U}_k|\}$ and replacing β with C according to the Eq.(4.2). On the other hand, when the search point $\boldsymbol{x}_i^t \notin \boldsymbol{U}_k$ refers to the reference point $\boldsymbol{z}_j \notin \boldsymbol{U}_k$, the parameter P is used to adjust γ by replacing the reference point $\boldsymbol{z}_j \notin \boldsymbol{U}_k$ with the best solution $\boldsymbol{Gbest}_k$ and replacing β with P according to the Eq.(4.2). For the proposed adjustment rule of γ, refer to the following **Algorithm 4.1**, and the image of adjustment of γ is shown in Fig. 4.3.

$$\gamma_i = \begin{cases} -\ln(C/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2 & \boldsymbol{z}_j \in \boldsymbol{U}_k \\ -\ln(P/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2 & \boldsymbol{z}_j \notin \boldsymbol{U}_k \end{cases} \tag{4.2}$$

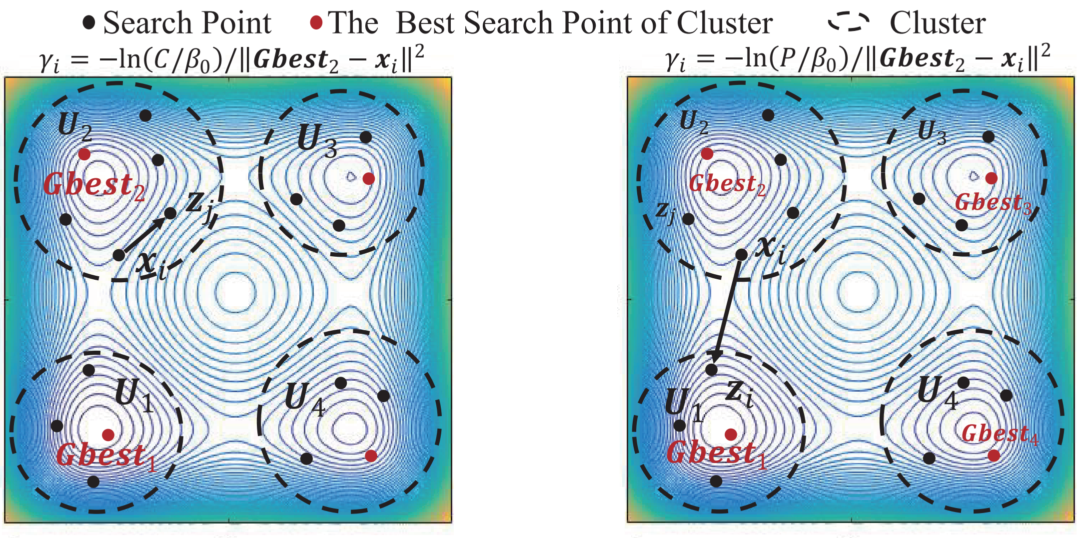


Fig. 4.3: Image of Adjustment Rule of γ_i

At the same time, the parameter α is the step width of the random perturbation to control the range of the search area in FA. Considering the stability of each group, it is difficult to converge when the perturbation is strong, and it is easy to converge when the perturbation is weak in the superior solution set search problem. Therefore, a wide area is searched at the beginning of the search, and a narrow area is searched at the end of the search. In FA, the distribution of perturbations gradually shrinks as the search points move even if they deteriorate. Therefore, the scheduling adjustment rule is used, and the concrete adjustment rule of α is shown in the Eq.(4.3).

$$\alpha^t = \alpha_{\max} - \frac{t}{T_{\max}} \cdot (\alpha_{\max} - \alpha_{\min}) \tag{4.3}$$

Firefly Algorithm based on cluster information for the minimization problem of the objective function $f(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathbb{R}^N)$ is shown by **Algorithm 4.2**.

Algorithm 4.2 Firefly Algorithm based on Cluster Information（FA-CI）

procedure FA-CI($m, \alpha_{\max}, \alpha_{\min}, \beta_0, K, C, P, T_{\max}$)

Step 1: Preparation

Set the maximum number of iterations $T_{\max}$, the number of search points m, $\beta_0 > 0$, K, C, P, $\alpha_{\max}$, and $\alpha_{\min}$. Set the number of iterations $t = 1$, $\alpha^1 = \alpha_{\max}$.

Step 2: Initialization

In the feasible area $\boldsymbol{X} \subseteq \mathbb{R}^N$, randomly generate search point $\boldsymbol{x}_i^1$ $(i = 1, 2, \cdots, m)$, and the swarm set $\boldsymbol{D}^1 = \{\boldsymbol{x}_i^1 \mid i = 1, 2, \cdots, m\}$.

Step 3: Calculation of light intensity

Calculate the light intensity I_i of each search point $\boldsymbol{x}_i^t \in \boldsymbol{D}^t$.

$\boldsymbol{x}_{cbest}^t = \underset{\boldsymbol{x}_i^t \in \boldsymbol{V}^t}{\arg\min}\{f(\boldsymbol{x}_i^t) \mid i = 1, 2, \cdots, m\}$

$I_i = \left(|f(\boldsymbol{x}_{cbest}^t) - f(\boldsymbol{x}_i^t)| + 1\right)^{-1}$

Each search point $\boldsymbol{x}_i^t \in \boldsymbol{P}^t$ is sorted in descending order of I_i^t.

Save the reference solution $\boldsymbol{z}_i$ and solution set $\boldsymbol{V}$.

$\boldsymbol{z}_i = \boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, m)$, $\boldsymbol{V} = \{\boldsymbol{z}_i \mid i = 1, 2, \cdots, m\}$

Assign each search point $\boldsymbol{x}_i^t$ to K clusters $\boldsymbol{U} = \{U_1, U_2, \cdots, U_K\}$.

$\{\boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_K\}$=$k$-MEANS ALGORITHM$(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m, K)$ ▷ See Algorithm B.1

Find the best point $\boldsymbol{Gbest}_k (k = 1, \cdots, K)$ of each cluster $\boldsymbol{U}_k (k = 1, 2, \cdots, K)$.

for $k = 1$ to K **do**

  $\boldsymbol{Gbest}_k = \underset{\boldsymbol{x}_l^t \in \boldsymbol{U}_k}{\arg\min}\{f(\boldsymbol{x}_l^t) | l = 1, 2, \cdots, |\boldsymbol{U}_k|\}$

end for

Set $i = 1$ and $j = 1$.

Step 4: Movement of the search point

for $i = 1$ to m **do**

  if $I_i < I_j$ **then**

    γ_i=GAMMA ADJUSTMENT RULE$(\boldsymbol{x}_i^t \in U_k, \boldsymbol{Gbest}_k^t, C, P)$ ▷ See Algorithm 4.1

    Move the search point $\boldsymbol{x}_i^t$ referring solution $\boldsymbol{z}_i \in \boldsymbol{V}$.

    $\boldsymbol{x}_i^t := \boldsymbol{x}_i^t + \beta_0 e^{-\gamma_i \|\boldsymbol{z}_j - \boldsymbol{x}_i^t\|^2}(\boldsymbol{z}_j - \boldsymbol{x}_i^t) + \alpha \boldsymbol{R}$

    Here, $\boldsymbol{R} \in [-0.5, 0.5]^N$ is a uniform random vector.

    Let $j := j + 1$.

    if $j = m$ **then**

      Move the best search point $\boldsymbol{x}_{cbest}^t \in \boldsymbol{D}^t$.

      $\boldsymbol{x}_{cbest}^t := \boldsymbol{x}_{cbest}^t + \alpha \boldsymbol{R}$

      Set $j := 1$.

end for

Step 5: Updating search points and each parameter

Update the search point $\boldsymbol{x}_i^t \in \boldsymbol{D}^t$ and the swarm set $\boldsymbol{D}^t$.

$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, m)$, $\boldsymbol{D}^{t+1} = \{\boldsymbol{x}_i^{t+1} \mid i = 1, 2, \cdots, m\}$

$\alpha^{t+1} = \alpha_{\max} - \frac{t}{T_{\max}} \cdot (\alpha_{\max} - \alpha_{\min})$

Step 6: Termination

if $t = T_{\max}$ **then**

  The algorithm is terminated.

else

  Return to **Step 3**, set $t := t + 1$.

end procedure

Table 4.1: Benchmark Function (See Appendix A)

Functions	Definitions	Search Space
Function 1	$\min\Big(f(\boldsymbol{x}+\boldsymbol{Y}),\ f(\boldsymbol{x}-\boldsymbol{Y}),\ f(\boldsymbol{x}+\boldsymbol{Z}),\ f(\boldsymbol{x}-\boldsymbol{Z})\Big)$	$[-5, 5]$
Function 2	$\min\Big(f(\boldsymbol{x}+\boldsymbol{Y}),\ g(\boldsymbol{x}+\boldsymbol{Z}),\ f(\boldsymbol{x}-\boldsymbol{Y}),\ g(\boldsymbol{x}-\boldsymbol{Z})\Big)$	
Function 3	$\min\Big(f(\boldsymbol{x}+\boldsymbol{Y})+\mathrm{E},\ g(\boldsymbol{x}+\boldsymbol{Z})-\mathrm{E},\ f(\boldsymbol{x}-\boldsymbol{Y})+\mathrm{F},\ g(\boldsymbol{x}-\boldsymbol{Z})-\mathrm{F}\Big)$	
Function 4	$\min\Big(g(\boldsymbol{x}+\boldsymbol{Y}),\ f(\boldsymbol{x}+\boldsymbol{Z}),\ g(\boldsymbol{x}-\boldsymbol{Y}),\ f(\boldsymbol{x}-\boldsymbol{Z})\Big)$	
$f(\boldsymbol{x})$	Sphere Function : $\sum_{i=1}^{n}(x_i)^2$	
$g(\boldsymbol{x})$	Schwefel′s Function : $\sum_{i=1}^{n}\Big(\sum_{j=1}^{i}x_j\Big)^2$	

Vectors	Definitions
$\boldsymbol{Y}$	$[2.5, 2.5, \cdots, 2.5, 2.5]^n$
$\boldsymbol{Z}$	$[2.5, -2.5, \cdots, 2.5, -2.5]^n$

Constants	Definitions
E	2.5
F	5

4.3.3 Numerical Experiment

(a) Numerical Experiment Conditions

In the numerical experiment, we compared the fixed parameter γ of FA with the proposed γ adjustment rule of FA which obtains superior solutions $\boldsymbol{x}^* \in \mathcal{Q}(\delta, \varepsilon)$ to evaluate the performance. However, it is impossible to obtain a strict superior solutions $\boldsymbol{x}^*$, which $\boldsymbol{x}_i^{T_{\max}}$ satisfies $\|\boldsymbol{x}_i^{T_{\max}} - \boldsymbol{x}^*\| \leq \eta$, and we determine that the $\boldsymbol{x}^*$ has been found. In this paper, we set up the problem so that the superior solution set $\mathcal{Q}(\delta, \varepsilon)$ is determined regardless of δ and ε for basic study. The benchmark functions Function 1 (F_1), Function 2 (F_2), Function 3 (F_3), and Function 4 (F_4) are given in Table 4.1, and they have multiple global optimal solutions $\boldsymbol{x}^\star$ far apart from each other for 5, 10, and 20 dimensions. In this case, we have $\boldsymbol{x}^* = \boldsymbol{x}^\star(\forall\delta \geq 0 \wedge \forall\varepsilon > 0)$.

We compare the proposed method with the original Firefly Algorithm and evaluate

Table 4.2: Results of Numerical Experiments (F_1)

Dim.		FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
			0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 5$	Mean	3.92 (16th)	1	2	2.02	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3.98	3.96	3.90
	Best	4 (1st)	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	Worst	3 (14th)	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3
	S.D.	0.27	0	0	0.14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.14	0.19	0.30
$n = 10$	Mean	3.20 (1st)	1	2.26	3.36	3.02	1.86	1.4	0.96	0.5	0.36	0.22	0.16	0.08	0.08	0.04	0.04	0.02	0.04	0.02	0	0
	Best	4 (1st)	1	4	4	4	3	3	2	3	2	1	2	1	1	1	1	1	1	1	0	0
	Worst	3 (1st)	1	1	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.54	0	0.69	0.60	0.85	0.76	0.88	0.70	0.74	0.60	0.42	0.42	0.27	0.27	0.19	0.20	0.14	0.20	0.14	0	0
$n = 20$	Mean	2.82 (1st)	1.82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	2 (1st)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.37	0.52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.3: Results of Numerical Experiments (F_2)

Dim.		FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
			0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 5$	Mean	3.84 (1st)	1	1.20	1.32	1.66	2.04	1.92	1.98	1.72	1.58	1.56	1.40	1.16	1.12	1.06	1	1	1	1	1	1.02
	Best	4 (1st)	1	2	2	3	4	3	3	3	2	2	2	2	2	2	1	1	1	1	1	2
	Worst	3 (1st)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S.D.	0.37	0	0.40	0.47	0.59	0.53	0.53	0.53	0.43	0.50	0.50	0.50	0.49	0.37	0.33	0.24	0	0	0	0	0.14
$n = 10$	Mean	3.40 (1st)	0.26	0.3	0.76	0.7	0.34	0.28	0.12	0	0.02	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	Worst	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.61	0.44	0.46	0.43	0.46	0.48	0.45	0.33	0	0.14	0	0	0	0	0	0	0	0	0	0	0
$n = 20$	Mean	1.32 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	0 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.4: Results of Numerical Experiments (F_3)

Dim.		FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
			0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 5$	Mean	3.80 (1st)	1.00	1.28	1.86	1.96	2.02	1.94	1.92	1.92	1.84	1.74	1.56	1.40	1.42	1.38	1.24	1.18	1.14	1.10	1.16	1.10
	Best	4 (1st)	1	2	3	3	4	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
	Worst	2 (1st)	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S.D.	0.40	0	0.45	0.50	0.57	0.47	0.55	0.49	0.49	0.37	0.44	0.50	0.49	0.50	0.49	0.43	0.39	0.35	0.30	0.37	0.30
$n = 10$	Mean	3.44 (1st)	0.38	0.28	0.50	0.62	0.42	0.22	0.10	0.04	0.02	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	Worst	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.54	0.49	0.45	0.51	0.49	0.50	0.42	0.30	0.20	0.02	0	0	0	0	0	0	0	0	0	0	0
$n = 20$	Mean	1.78 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	0 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.5: Results of Numerical Experiments (F_4)

Dim.		FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
			0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 5$	Mean	3.76 (1st)	1.00	1.32	1.92	1.96	2.12	2.02	1.90	1.86	1.82	1.70	1.56	1.42	1.40	1.38	1.24	1.16	1.14	1.10	1.16	1.10
	Best	4 (1st)	1	2	3	3	4	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
	Worst	2 (1st)	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S.D.	0.42	0	0.50	0.52	0.60	0.45	0.53	0.49	0.49	0.35	0.44	0.48	0.45	0.49	0.43	0.43	0.39	0.35	0.30	0.37	0.30
$n = 10$	Mean	3.32 (1st)	0.40	0.28	0.50	0.63	0.40	0.21	0.11	0.04	0	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.60	0.49	0.45	0.51	0.49	0.50	0.42	0.30	0.20	0	0	0	0	0	0	0	0	0	0	0	0
$n = 20$	Mean	1.62 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	0 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

the performance of the proposed method (FA-CI) for the basic case of the superior solution set search problem. As a shared experiment condition, the number of trials was set to 50. The maximum number of iterations was 1000 and η was 0.5. In addition, the recommended values of other parameters β_0 was 1.0, α was using the Eq.(4.8) where $\alpha_{\max}$ was 1.0 and $\alpha_{\min}$ was 0.01. The parameters of original FA, we changed γ from 0.05 to 1.0 in 20 runs of 0.05, as given in Table 4.2 $\sim$ Table 4.5. The parameters of FA-CI were $C = 10^{-7}$ referring to points in same cluster, $P = 10^{-9}$ referring to points in different clusters, cluster number $K = 4$, $\alpha_{\max} = 0.1$, and $\alpha_{\min} = 0.01$.

We determine the evaluation indicates by the number of superior solutions acquired in one trial. "Best" denotes the best evaluation value of 50 runs, "Worst" refers to the worst value of 50 runs, "Mean" denotes the average value of 50 runs, and "S.D." denotes the standard deviation of 50 runs. Moreover, the ranking of the FA-CI in the results of the original FA (OFA) is expressed as "Rank." The value of "Rank" indicates the order of the FA-CI in the results.

(b) Results of the Numerical Experiment

Table 4.2 $\sim$ Table 4.5 show the results of numerical experiment. In the case of the given dimensions in F_1, FA-CI has almost the same performance as other comparison methods, but it ranked 16 out of the 20 types in the original FA. In the case of 10 and 20 dimensions in F_1, the performance of FA-CI is superior to the results of the 1st compared to 20 types in the original FA. In the case of 5, 10, and 20 dimensions in F_2, F_3, and F_4, the performance of AFA is superior to that achieved with 20 types in the original FA. From the above, it is verified that FA-CI has excellent performance.

4.4 Adaptive Firefly Algorithm Based on Evaluation and Control of Search State for Superior Solution Set Search Problem

Metaheuristics can efficiently search through adjustable parameters of each method. However, it is necessary to consider the problem structure and the search condition for setting appropriate parameters, and doing so requires considerable expertise and experience. Meanwhile, it is looking forward that can also realize a high-performance search for various problems and search conditions with the optimization field of the external environment changing. Therefore, considering the application of metaheuristics to real systems, development of the adaptive parameter adjustment rule is important.

Furthermore, actual optimization problems are generally either single objective or multi-objective optimization problems, and to extend the degree of freedom of the choice of solutions for decision makers, the development of a method that yields multiple solutions at the same time for multimodal functions is important. From the above viewpoints, we proposed a superior solution set search problem [60, 61] in a previous study, which searches for various solution sets whose evaluated values are similar and the solution distance is far away. It is expected to present alternative proposals to solve problems when accidents or technical problems occur. Compared with other metaheuristics for single objective optimization problems, FA can be divided into multiple groups in the search process of the superior solution set search problem, and the search for multiple promising regions can be performed in parallel. Therefore, FA expected to be able to efficiently search for the superior solution set, which has different properties compared to many single objective optimization methods.

To improve the search performance of metaheuristics for a single objective optimization problem, it is important to appropriately realize search guidelines for diversification and intensification [22, 23, 24, 53]. Based on the universal search structure and strategy of many metaheuristics in single objective optimization problems, we introduce the approach "cluster diversification and intensification" for the superior solution set search problem. In addition, from the viewpoint of diversification and intensification, we analyze the parameters of FA and cluster diversification and intensification to develop a parameter adjustment rule for FA. We show that it is possible to improve the performance of finding the superior solution set by adaptive control of cluster diversification and intensification.

4.4.1 Analysis of Firefly Algorithm Parameters and Diversification and Intensification

In this Section, we analyze FA parameters and diversification / intensification, and clarify the relationship between the action of each FA operation and diversification / intensification. If the movement β of each search point increases, then the search points gather into a single cluster earlier. This is not conducive to search points that are divided into separate clusters and is inefficient for finding the superior solution set. However, if the movement β of each search point decreases, the search point group becomes more similar to random exploration. This is not conducive to exploring the evolution of points and is also inefficient for finding the superior solution set. Therefore, from the viewpoint of an optimization method that can search efficiently, determining the movement amount is extremely important. Determination of the amount of movement is directly related to the parameter γ. When the parameter γ, which adjusts the amount of movement increases, the movement of each search point decreases. Conversely, when γ decreases, the movement of the search points increases. This parameter can efficiently adjust the movement of the search points, including movement between the clusters, and can improve the performance of searching for the superior solution set. At the same time, in Section 4.3, we proposed a new adjustment rule for the parameter γ. In the adjustment rule, we use the parameters C and P to adjust the parameter γ to achieve the effect of grouping and adjusting the search state.

At the same time, parameter α is the step size quantization factor of the random movement term in FA, and it controls the range of exploration areas in FA. Considering the stability of each cluster in the superior solution set search, convergence is difficult when perturbation is strong, and it is easy when perturbation is weak. Therefore, it is expected to be adjusted appropriately according to the state of each cluster.

In the superior solution set search problem, it is desirable that diversification and intensification cerate clusters similar to those in conventional single-objective optimization methods while maintaining diversity between the clusters. We define diversification and intensification for the superior solution set search problem as follows. Diversification is defined as "searching a large area within a cluster", "suppressing attraction of solutions between clusters" and "expansion of perturbation." Intensification is defined as "searching a narrow area within a cluster", "promotion of attraction between solutions in different clusters" and "reduction of perturbation." The attraction in the solution to FA is controlled by the parameter γ. From the γ adjustment rule, we set parameters C and P to adjust the parameter γ. So, parameter C controls diversification and intensification within the cluster, and parameter P controls diversification and intensification between clusters.

From the above parameter analysis, i) the movement for approaching an excellent reference point weakens when γ is small (diversification), and the movement for approaching an excellent reference point becomes stronger when γ is large (intensification). As we know, when parameters C and P are small, γ is large. So, the movement for approaching an excellent reference point weakens when parameters C and P are large (diversification), and the movement for approaching an excellent reference point becomes stronger when parameters C and P are small (intensification). Moreover, ii) the perturbation becomes wider when α is large (diversification), and the perturbation becomes narrow when α is small (intensification). The relationship between FA parameters and diversification / intensification in this paper is shown in the Table 4.6.

Moreover, to efficiently search in a finite time search problem, "each cluster is diversified at the initial stage of searching", and "each cluster is intensified in the final searching stage." "All search clusters are close at the initial of search", and "each cluster is separated from the others at the end of search" with the aim of realizing a search strategy. Fig.4.4 shows an image of the search strategy used in the Function 1 in two-

Table 4.6: Relationship Between Firefly Algorithm Parameters and Diversification / Intensification

Parameters	Definition		
γ	Small	$\Longleftrightarrow$	Large
C	Large	$\Longleftrightarrow$	Small
P	Large	$\Longleftrightarrow$	Small
Search Strategy	Diversification	$\Longleftrightarrow$	Intensification

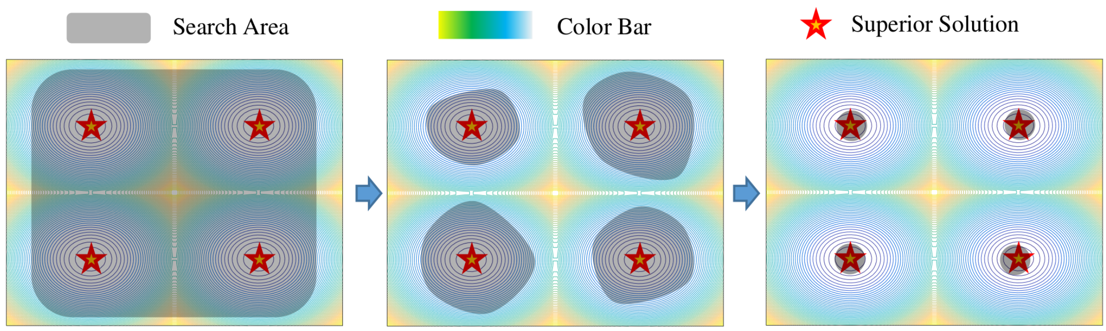


Fig. 4.4: Effective Search Strategy of Metaheuristics for Superior Solution Set Search Problem

dimensional space. Thus, a search strategy that appropriately realizes diversification and intensification in the search process is important in metaheuristics.

4.4.2 Evaluation Indicators of Diversification and Intensification for the Superior Solution Set Search

Generally, it is important to properly realize diversification and intensification for improving the performance of metaheuristics [19, 22, 23, 53, 65, 66, 67, 68], and in analysis of Section 4.4.1 for the superior solution set search problem, it is pointed out that FA does not have the adjustment ability of diversification and intensification. In this Section, we define evaluation indicators for the diversification and intensification for the superior solution set search problem newly. Furthermore, we show that the evaluation indicators can evaluate the search state of diversification and intensification for the superior solution set search problem by conducting numerical experiments.

(a) Evaluation Indicators

To realize diversification and intensification, 1) the strength of the interaction between the clusters, and 2) the magnitude of perturbation are adjusted. In this Section, we evaluate the realization state of diversification and intensification by "promotion / suppression of actions in the cluster and between clusters" and "expansion / reduction of perturbation" based on the analysis of parameters in Section 4.4.1.

Based on the above, we consider the evaluation indicators of actions by the cluster information ($\boldsymbol{U} = \{\boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_K\}$). In this paper, we evaluate the intensification state of search points within clusters and the degree of separation between clusters based on cluster information. It is desirable that diversification and intensification create cluster similar to those in conventional single-objective optimization methods while maintaining diversity between clusters. Therefore, the evaluation indicator $\mathcal{A}$ of a cluster is defined as the mean value of the distances between the search points in each cluster based on the Eq.(4.4), which can be used to evaluate the search area. The evaluation indicator $\mathcal{B}$ between clusters can be defined as the mean value of the centroid $\mathcal{G}_k = \frac{1}{|\boldsymbol{U}_k|} \sum_{\boldsymbol{x} \in \boldsymbol{U}_k} \boldsymbol{x}$ $(k = 1, \cdots, K)$ distances between each cluster based on the Eq.(4.5), which can evaluate the degree of separation between clusters. Fig.4.5 presents example

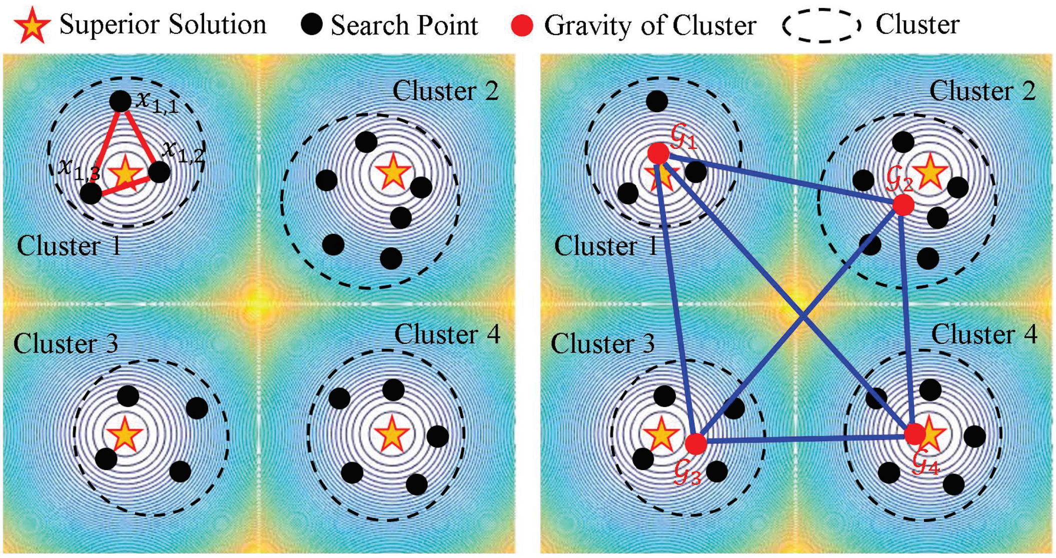


Fig. 4.5: Image of Evaluation indicators $\mathcal{A}$ and $\mathcal{B}$

images of the evaluation indicators $\mathcal{A}$ and $\mathcal{B}$.

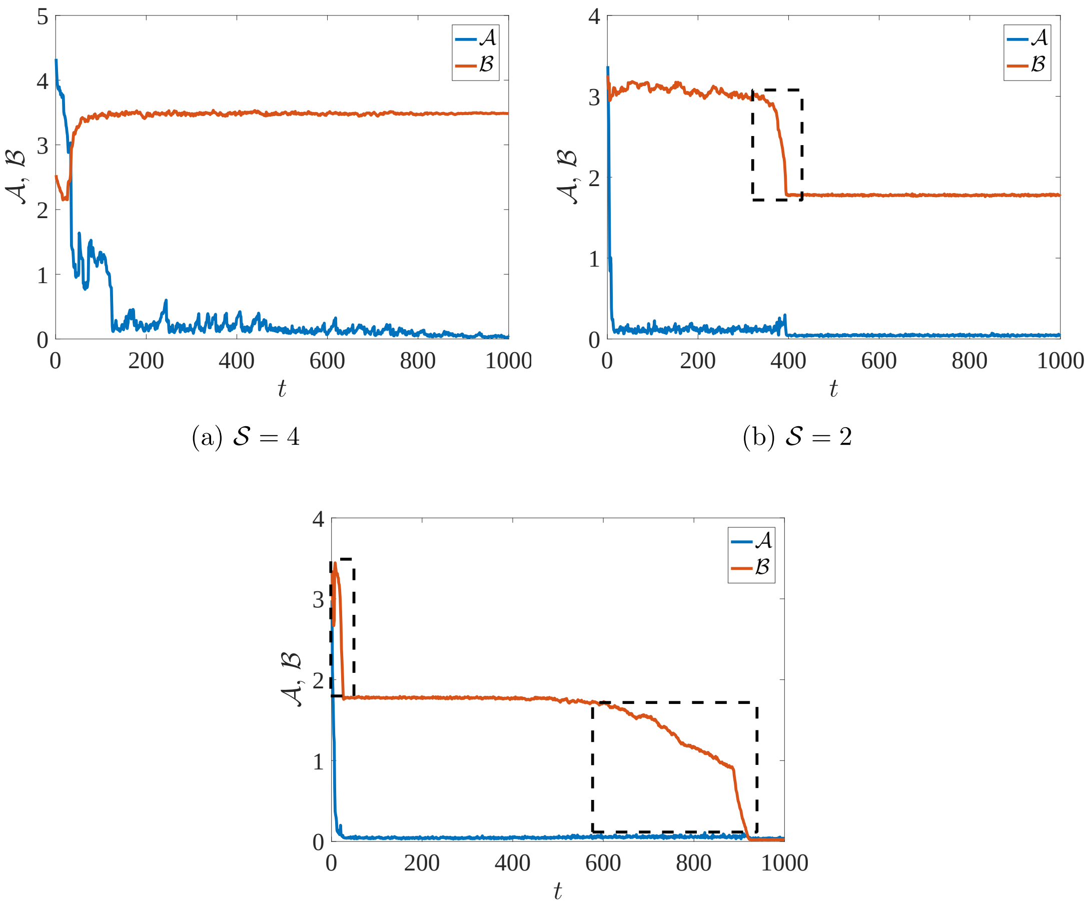
$$\mathcal{A} = \frac{1}{\sqrt{n}l} \sum_{k=1}^{K} \sum_{i=1}^{m_k-1} \sum_{j=i+1}^{m_k} \|\boldsymbol{x}_{k,i} - \boldsymbol{x}_{k,j}\| \tag{4.4}$$

$$\mathcal{B} = \frac{1}{\sqrt{n}L} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \|\mathcal{G}_i - \mathcal{G}_j\| \tag{4.5}$$

$$l = \sum_{k=1}^{K} {}_{m_k}\mathrm{C}_2 \tag{4.6}$$

$$L = {}_{K}\mathrm{C}_2 \tag{4.7}$$

Here, $\boldsymbol{x}_{k,i}$ is the search point $\boldsymbol{x}_i$ in the k-th cluster, $\mathcal{G}_k$ is the center of gravity of the k-th cluster, m_k is the number of search points in the k-th cluster, K is the number of clusters, $\mathcal{A}$ and $\mathcal{B}$ divided by $\sqrt{n}$ to remove the influence of dimension n.



(a) $\mathcal{S} = 4$

(b) $\mathcal{S} = 2$

(c) $\mathcal{S} = 1$

Fig. 4.6: Transitions of $\mathcal{A}$ and $\mathcal{B}$ in Original Firefly Algorithm

If $\mathcal{A}$ is large, the average distance between the search points in the cluster is large (the search point distribution is wide). If $\mathcal{A}$ is small, the average distance between the search points in the cluster is small (the search point distribution is narrow). If the fluctuation in $\mathcal{B}$ is large, the degree of separation between clusters varies greatly. An image of the search strategy for the superior solution set search problem is presented in Fig.4.4. While maintaining a constant distance between clusters, the search is performed by progressing from a wide area to a narrow area within the clusters. To facilitate this search strategy, the evaluation indicator $\mathcal{A}$ should be gradually reduced while the evaluation indicator $\mathcal{B}$ remains relatively constant.

(b) Numerical Experiment on Evaluation Indicators

In this Section, we show that the evaluation indicators $\mathcal{A}$ and $\mathcal{B}$ for diversification and intensification of clusters, respectively, can be evaluated through numerical experiments. First, we show the values and transitions of $\mathcal{A}$ and $\mathcal{B}$ when the number of superior solutions obtained $\mathcal{S}$ is different. Applying FA to benchmark function Function 1 (refer to Table 4.1), changes in the evaluation indicators for each number of superior solutions obtained are shown. As common experimental conditions, the number of search points m was set to 50, number of dimensions n was set to 5, α was set to 0.05, attenuation parameter γ was set to 0.25, and the maximum number of iterations $T_{\max}$ was set to 1000.

Fig.4.6 illustrates the transitions in the evaluation indicators $\mathcal{A}$ and $\mathcal{B}$ as the number of obtained superior solutions $\mathcal{S}$ changes. The evaluation indes $\mathcal{A}$ of a cluster decreases earlier and converges for each cluster during searching. The number of superior solutions obtained $\mathcal{S}$ is four in Fig.4.6(a). The evaluation indicator between clusters $\mathcal{B}$ tends to maintain a certain level and each cluster performs a parallel search while maintaining a certain distance between clusters. The numbers of superior solutions obtained $\mathcal{S}$ are two and one in Fig.4.6(b) and (c) respectively. The evaluation indicator $\mathcal{B}$ between clusters shows large fluctuations within the black box and it was confirmed that the clusters merged. Therefore, through numerical experiments, we have confirmed that the evaluation indicators $\mathcal{A}$ and $\mathcal{B}$ defined in this paper can accurately evaluate the clustering in a given search state.

4.4.3 Proposal of Adaptive Firefly Algorithm

(a) Parameter Adjustment Strategy

Metaheuristics have tunable parameters, and it is possible to perform a search by exploiting the degrees of freedom of the parameters and appropriately setting them according to the search conditions and problem structure. Considering the problem structure and search condition for appropriate parameter setting, which requires expert knowledge and considerable experience, is necessary. Meanwhile, metaheuristics in the field of optimization of the external environment is being developed to realize high-performance search for various problems and search conditions. Therefore, for applying metaheuristics to real systems, extracting effective knowledge on parameter setting / adjustment and systematic classification / organization and development of adaptive parameter adjustment are required.

Thus far, several studies on the adaptation of metaheuristics and parameter adjustment methods have been reported [27, 34, 36, 38, 69, 70, 71, 72, 73, 74, 75]. The authors believe that classification can be made from the viewpoint of adaptability of optimization algorithms. Based on this, parameter adjustment methods for metaheuristics can be classified as follows.

- Parameter fixed rule: Reference [27, 34, 36, 38]

  A mechanism for presetting parameters and searching without changing the search process.
- Scheduling Adjustment Rule: Reference [69, 70, 71]

  A mechanism for preliminarily adjusting parameters according to a schedule by a user.
- Reactive adjustment rule: Reference [72]

  A mechanism for adjusting parameters when some predetermined condition is satisfied in the search process.
- Adaptive adjustment rule: Reference [73]

  A mechanism for adjusting parameters that follows the search guidelines from the information obtained in the search process after giving some type of information, which is the guideline for the search from outside in advance.

- Autonomous adaptive adjustment rule [74, 75]
  A mechanism that creates and improves guidelines based only on internal information obtained in the search process and adjusts parameters.

(b) Rule for the Adjustment of Adaptive Firefly Algorithm Parameters

In this Section, we propose an Adaptive FA based on the evaluation and control of the cluster search state. We proposed diversification and intensification evaluation indicators for the superior solution set search based on FA, and quantitatively confirmed the evaluation of the diversification and intensification of clusters through numerical experiments in Section 4.4.2. Moreover, the relationship between parameter performance analysis and diversification and intensification was clarified in Section 4.4.1. Therefore, it is possible to control diversification and intensification so that the evaluation indicators follow a preset target value while evaluating the realization state of FA diversification and intensification in the search process.

In this paper, we adapt FA using clustering for application to the superior solution set search problem. Using the clustering algorithm, we divide the search point group into the k-th clusters U_k $(k = 1, \cdots, K)$. In addition, although arbitrary clustering methods can be used, we obtain cluster information using k-means clustering (see **Algorithm B.1** in Appendix B), which is one of the representative clustering methods. We then exploit the cluster information to improve the search capability for a superior solution set. Additionally, it is expected that the adjustment ability of diversification and intensification can be improved by adding adjustment rules for α and γ to FA.

First, we describe the adjustment rule of α. The search point moves even if it is worse at the time of "neighborhood generation" in FA, and it is desirable that distribution of the perturbation gradually shrinks. Therefore, we show the scheduling adjustment rule, and a concrete rule for adjusting α in the Eq.(4.8).

$$\alpha^t = \alpha_{\max} - \frac{t}{T_{\max}} \cdot (\alpha_{\max} - \alpha_{\min}) \tag{4.8}$$

Next, we describe the adjustment rule of γ. The adjustment rule for γ follows that proposed by the authors [88] in Section 4.3.2. The search point $\boldsymbol{x}_i$ belonging to the k-th cluster U_k holds its own γ_i. By the way, the Eq.(3.24) of γ in FA can be transformed into the Eq.(4.9). The distance $||\boldsymbol{z}_j - \boldsymbol{x}_i^t||^2$ from the search point $\boldsymbol{x}_i^t$ to the reference point $\boldsymbol{z}_j$ is replaced with $\|\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t\|$, and β is replaced with C and P. Here, $\boldsymbol{Gbest}_k$ $(= \arg\min_{\boldsymbol{x}_l^t \in \boldsymbol{U}_k}\{f(\boldsymbol{x}_l^t)|l = 1, 2, \cdots, |\boldsymbol{U}_k|\})$ is the best search point belonging to the cluster $\boldsymbol{U}_k$.

$$\gamma_i = -\ln(\beta/\beta_0)/||\boldsymbol{z}_j^t - \boldsymbol{x}_i^t||^2 \tag{4.9}$$

We use the Eq.(4.10) as the adjustment rule of γ_i with reference to the Eq.(4.9).

$$\gamma_i = \begin{cases} -\ln(C^t/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2, & \boldsymbol{z}_j \in U_k \\ -\ln(P^t/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2, & \boldsymbol{z}_j \notin U_k \end{cases} \tag{4.10}$$

Additionally, the new parameters C and P are adjusted according to adjustment rules based on diversification and intensification between clusters and within a cluster. First, we describe the adjustment rule for C. The Eq.(4.11) shows the adjustment rule of concrete C when the search point $\boldsymbol{x}_i^t$ refers to the search point in the same cluster. Given an advance target schedule $I_{\mathcal{A}}^t$ $(t = 1, 2, \cdots, T_{\max})$ for comparison with the value of the evaluation indicator $\mathcal{A}$, $\mathcal{A}$ can be adjusted by C based on the adjustment width ΔC following $I_{\mathcal{A}}^t$. If the evaluation indicator $\mathcal{A}$ is smaller than the target schedule $I_{\mathcal{A}}^t$, the search points of the cluster become diversification and adjustment makes C decrease. If the evaluation indicator $\mathcal{A}$ is larger than the target schedule $I_{\mathcal{A}}^t$, the search points of the cluster become intensification and adjustment makes C increase.

$$C^t = \begin{cases} \min\{C^{t-1} + \Delta C,\ C_{\max}\}, & \mathcal{A}^t \geq I_{\mathcal{A}}^t \\ \max\{C^{t-1} - \Delta C,\ C_{\min}\}, & \mathcal{A}^t < I_{\mathcal{A}}^t \end{cases} \tag{4.11}$$

Next, we describe the adjustment rule of P. By contrast, the Eq.(4.12) shows the specific adjustment rule of P when the search point $\boldsymbol{x}_i^t$ refers to a search point in different

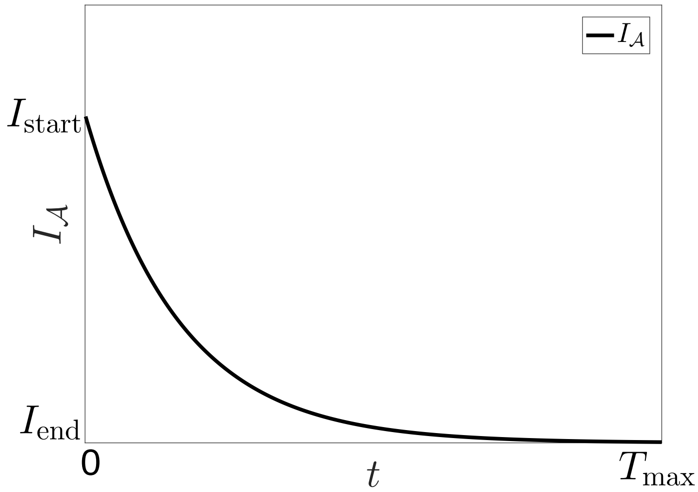


Fig. 4.7: Target Schedule $I_{\mathcal{A}}$

clusters. We adjust the parameter P such that the average value $\mathcal{B}$ of the center of gravity of each cluster does not fluctuate significantly. If the variation amount θ is greater than the average of fluctuations $\Delta\mathcal{B}$ in T times (see the Eq.(4.13)), the search points between clusters are intensified and adjustment causes P to increase. If the variation amount θ is smaller than the average of fluctuations $\Delta\mathcal{B}$ in T times (see the Eq.(4.13)), the search points within the clusters are diversified and adjustment causes P to decrease. We adjust P by the defined adjustment width ΔP.

$$P^t = \begin{cases} \min\{P^{t-1} + \Delta P,\ P_{\max}\},\ \Delta\mathcal{B}^t > \theta \\ P^{t-1},\ \ (-\theta \leq \Delta\mathcal{B}^t \leq \theta) \\ \max\{P^{t-1} - \Delta P,\ P_{\min}\},\ \Delta\mathcal{B}^t < -\theta \end{cases} \tag{4.12}$$

$$\Delta\mathcal{B}^t = \begin{cases} \left(\frac{1}{T}\sum_{i=t-(T+1)}^{t-1} \mathcal{B}^i\right) - \mathcal{B}^t,\ t > T+1 \\ 0,\ 0 < t \leq T+1 \end{cases} \tag{4.13}$$

It is possible to control the search conditions at each search stage using this adjustment rule when target value schedules $I_{\mathcal{A}}$ is appropriately set. Moreover, to efficiently solve within a reasonable period, we aim to realize a search strategy; that is the search points in each cluster are in various states at the beginning of the search, and the search points in each cluster are concentrated at the end of the search. In this paper, we use exponential schedule of the target value I^t, as shown in Fig.4.7, and the exponential schedule expression is expressed by the Eq.(4.14).

$$I^t = I_{\text{start}} \left(\frac{I_{\text{end}}}{I_{\text{start}}} \right)^{t/T_{\max}} \tag{4.14}$$

There are several ways to choose I_{start}. In this paper, based on our numerical experiments, we adopt a value of approximately $\frac{2}{K}$ on one side of the search area, which is represented by the hypercube in the benchmark problem. We simply set I_{end} to 0 or a sufficiently small positive value so that intensification of searching within clusters near the end of the search process is realized. The recommended values of I_{start} and I_{end} are 5 and 0.001, respectively, for the evaluation indicator $\mathcal{A}$.

Adaptive Firefly Algorithm for the minimization problem of the objective function $f(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^N$) is shown by **Algorithm 4.3**.

Algorithm 4.3 Adaptive Firefly Algorithm（AFA）

procedure AFA($m, \beta_0, T_{\max}, T, K, \theta, \alpha_{\max}, \alpha_{\min}, I_{\text{start}}, I_{\text{end}}, C_{\max}, C_{\min}, \Delta C, P_{\max}, P_{\min}, \Delta P$)

Step 1: Preparation

Set the maximum number of iterations $T_{\max}$, number of search points m, the number of clusters K, each parameter β_0, T, and θ, the target schedule parameters $\alpha_{\max}$, $\alpha_{\min}$, I_{start}, and I_{end}.

Set parameters $C_{\max}$, $C_{\min}$, ΔC, $P_{\max}$, $P_{\min}$, and ΔP.

Set $t = 1$, $\alpha^1 = \alpha_{\max}$, $C^1 = (C_{\max} + C_{\min})/2$, and $P^1 = (P_{\max} + P_{\min})/2$.

Step 2: Initialization

In the feasible area, randomly generate search point $\boldsymbol{x}_i^1$ $(i = 1, 2, \cdots, m)$.

Save the solution as $\boldsymbol{z}_i = \boldsymbol{x}_i^1$ $(i = 1, 2, \cdots, m)$, and set $i = 1$.

Step 3: Calculation of light intensity

Calculate the light intensity I_i of each search point $\boldsymbol{x}_i^t$ using the following equations.

$f_{\min}^t = \min\{f(\boldsymbol{x}_i^t) | i = 1, 2, \cdots, m\},\ I_i = \left(|f_{\min}^t - f(\boldsymbol{x}_i^t)| + 1\right)^{-1}$

Each search point $\boldsymbol{x}_i^t$ is sorted in descending order of I_i^t and set $i = 1$ and $j = 1$.

Step 4: Calculation of evaluation indicators

Assign each search point $\boldsymbol{x}_i^t$ to each cluster U_k $(k = 1, 2, \cdots, K)$.

Find the best solution $\boldsymbol{Gbest}_k^t = \arg\min\{f(\boldsymbol{y}^t) \mid \boldsymbol{y}^t \in U_k\}$ among the respective clusters U_k.

Calculate the evaluation indicators $\mathcal{A}$ and $\mathcal{B}$ by the following equations.

$\mathcal{A} = \frac{1}{\sqrt{n l}} \sum_{k=1}^{K} \sum_{i=1}^{m_k - 1} \sum_{j=i+1}^{m_k} \|\boldsymbol{x}_{k,i} - \boldsymbol{x}_{k,j}\|,\ l = \sum_{k=1}^{K} {}_{m_k}\mathrm{C}_2$

$\mathcal{B} = \frac{1}{\sqrt{n L}} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \|\mathcal{G}_i - \mathcal{G}_j\|,\ L = {}_K\mathrm{C}_2$

m_k is the number of search points in cluster U_k, and $\mathcal{G}_k$ is the center of gravity of cluster U_k.

Step 5: Movement of search points

for $i = 1$ to m **do**

  if $I_i < I_j$ **then**

$$\gamma_i = \begin{cases} -\ln(C^t/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2, & \boldsymbol{z}_j \in U_k \\ -\ln(P^t/\beta_0)/||\boldsymbol{Gbest}_k^t - \boldsymbol{x}_i^t||^2, & \boldsymbol{z}_j \notin U_k \end{cases}$$

$$\boldsymbol{x}_i^t := \boldsymbol{x}_i^t + \beta_0 e^{-\gamma_i \|\boldsymbol{z}_j - \boldsymbol{x}_i^t\|^2} (\boldsymbol{z}_j - \boldsymbol{x}_i^t) + \alpha \boldsymbol{R}$$

  Let $j := j + 1$, until $j = m$ and move the search point $\boldsymbol{x}_i^t$ by the following equation.

$$\boldsymbol{x}_i^t := \boldsymbol{x}_i^t + \alpha \boldsymbol{R}$$

  Set $j := 1$.

end for

Step 6: Updating search points and each parameter

Update the search point $\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t$, and save the solution $\boldsymbol{z}_j = \boldsymbol{x}_i^{t+1}$.

$$\alpha^{t+1} = \alpha_{\max} - \frac{t}{T_{\max}} \cdot (\alpha_{\max} - \alpha_{\min})$$

$$C^{t+1} = \begin{cases} \min\{C^t + \Delta C,\ C_{\max}\},\ \mathcal{A}^{t+1} \geq I_{\mathcal{A}}^{t+1} \\ \max\{C^t - \Delta C,\ C_{\min}\},\ \mathcal{A}^{t+1} < I_{\mathcal{A}}^{t+1} \end{cases}$$

$$P^{t+1} = \begin{cases} \min\{P^t + \Delta P,\ P_{\max}\},\ \Delta\mathcal{B}^{t+1} > \theta \\ P^t,\ \ (-\theta \leq \Delta\mathcal{B}^{t+1} \leq \theta) \\ \max\{P^t - \Delta P,\ P_{\min}\},\ \Delta\mathcal{B}^{t+1} < -\theta \end{cases}$$

$$\Delta\mathcal{B}^{t+1} = \begin{cases} \left(\frac{1}{T} \sum_{i=t-T}^{t} \mathcal{B}^i\right) - \mathcal{B}^{t+1},\ t > T + 1 \\ 0,\ 0 < t \leq T + 1 \end{cases}$$

Step 7: Termination

if $t = T_{\max}$ **then**

  The algorithm is terminated.

else

  Return to **Step 3**, set $t := t + 1$.

end procedure

4.4.4 Numerical Experiment

(a) Numerical Experiment Conditions

In the numerical experiment, we compared the fixed parameter γ of FA with the proposed γ adjustment rule of FA which obtains superior solutions $\boldsymbol{x}^* \in \mathcal{Q}(\delta, \varepsilon)$ to evaluate the performance. However, it is impossible to obtain a strict superior solutions $\boldsymbol{x}^*$, which $\boldsymbol{x}_i^{T_{\max}}$ satisfies $\|\boldsymbol{x}_i^{T_{\max}} - \boldsymbol{x}^*\| \leq \eta$, and we determine that the $\boldsymbol{x}^*$ has been found. In this paper, we set up the problem so that the superior solution set $\mathcal{Q}(\delta, \varepsilon)$ is determined regardless of δ and ε for basic study. The benchmark functions Function 1 (F_1), Function 2 (F_2), Function 3 (F_3), and Function 4 (F_4) are given in Table 4.1, and they have multiple global optimal solutions $\boldsymbol{x}^\star$ far apart from each other for 5, 10, and 20 dimensions. In this case, we have $\boldsymbol{x}^* = \boldsymbol{x}^\star (\forall \delta \geq 0 \wedge \forall \varepsilon > 0)$.

In this paper, we examined the features of FA as they relate to the superior solution set search problem described in Section 4.4.1. FA has the property of searching in parallel while separating clusters, which means that the search points between clusters and within clusters can not be appropriately adjusted. Based on the concept of diversification and intensification, we proposed evaluation indicators $\mathcal{A}$ and $\mathcal{B}$. We proposed an adaptive FA (AFA) that adjusts the search points in clusters and between clusters based on the evaluation indicators. To illustrate the effects of adjusting the search points in clusters and between clusters, we compared the original FA, FA that adjusts by cluster information (FA-CI), an FA that only adjusts $\mathcal{A}$ (FA-test), and the proposed adaptive FA. We compared the original FA, FA-CI, and the FA that only adjusts $\mathcal{A}$ to confirm the effects of adjusting the search points within the clusters. We compare the FA that only adjusts $\mathcal{A}$ to the proposed adaptive FA to confirm the effects of adjusting the search points between clusters.

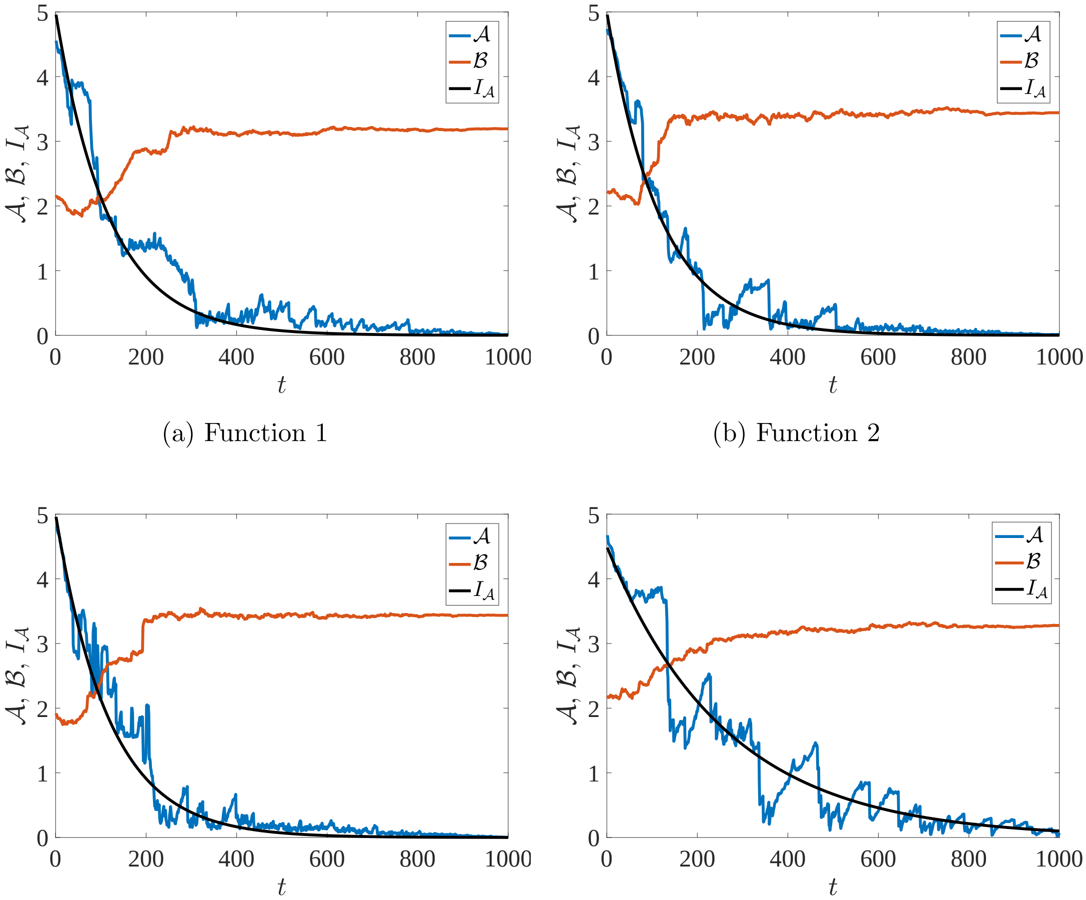
As a shared experiment condition, the number of trials was set to 50. The maximum number of iterations was 1000 and η was 0.5. In addition, the recommended values of other parameters β_0 was 1.0, α was using the Eq.(4.8) where $\alpha_{\max}$ was 1.0 and $\alpha_{\min}$ was 0.01. The other conditions for the original FA, FA-CI, FA-test, and AFA were shown below.

- The parameters of original FA, we changed γ from 0.05 to 1.0 in 20 runs of 0.05, as given in Table 4.7 ∼ Table 4.10.
- The parameters of FA-CI were $C = 10^{-7}$ referring to points in same cluster, $P = 10^{-9}$ referring to points in different clusters, cluster number $K = 4$, $\alpha_{\max} = 0.1$, and $\alpha_{\min} = 0.01$.
- The parameters of FA-test were $C_{\max} = 10^{-6}$, $C_{\min} = 10^{-8}$, $\Delta C = 10^{-8}$ referring to points in same cluster, $P = 10^{-6}$ referring to points in different clusters, cluster number $K = 4$, $\alpha_{\max} = 0.1$, and $\alpha_{\min} = 0.01$.
- The parameters of AFA were $C_{\max} = 10^{-6}$, $C_{\min} = 10^{-8}$, $\Delta C = 10^{-8}$ referring to points in same cluster, $P_{\max} = 10^{-8}$, $P_{\min} = 10^{-10}$, $\Delta P = 10^{-10}$ referring to points in different clusters, cluster number $K = 4$, $\theta = 0.3$, and $T = 5$.

We determine the evaluation indicates by the number of superior solutions acquired in one trial. "Best" denotes the best evaluation value of 50 runs, "Worst" refers to the worst value of 50 runs, "Mean" denotes the average value of 50 runs, and "S.D." denotes the standard deviation of 50 runs. Moreover, the ranking of the AFA in the results of the all comparison methods is expressed as "Rank." The value of "Rank" indicates the order of the AFA in the results of the all comparison methods.

(b) Results of the Numerical Experiment

Table 4.7 ∼ Table 4.10 show the results of the numerical experiment. The subscript in the results of the proposed method (AFA) indicates the order of the proposed method, including the case of fixed γ (OFA), FA that only adjusts $\mathcal{A}$ (FA-test), and FA that adjusts by cluster information (FA-CI). In addition, Fig.4.8 shows the transition of the diversification and intensification evaluation indicators $\mathcal{A}$ and $\mathcal{B}$ for the benchmark functions F_1, F_2, F_3, and F_4 $(n = 10)$. According to Fig.4.8, the evaluation indicator $\mathcal{A}$ follows target values $I^t_{\mathcal{A}}$ and the evaluation indicator $\mathcal{B}$ keeps a certain size (keeping



(a) Function 1 (b) Function 2

(c) Function 3 (d) Function 4

Fig. 4.8: Transitions of $\mathcal{A}$ and $\mathcal{B}$ in Adptive Firefly Algorithm

diversification) in the Adaptive FA. In the case of the given dimensions in F_1, AFA has almost the same performance as other comparison methods, but it ranked 15 out of the 23 types in all comparison methods. In the case of 10 and 20 dimensions in F_1, the performance of AFA is superior to the results of the 1st compared to 23 types in the all comparison methods. In the case of 5, 10, and 20 dimensions in F_2, F_3, and F_4, the performance of AFA is superior to that achieved with 23 types in the all comparison methods.

From Table 4.7 $\sim$ Table 4.10, the performance of AFA can be confirmed to be slightly better than that of FA that only adjusts $\mathcal{A}$ (FA-test). It can also be confirmed that the performance of FA-test is slightly better than OFA. It can be said that AFA has excellent performance, and it reduces the effort involved in parameter setting. In particular, it can be concluded that as the number of dimensions increases, the performance of AFA improves. Therefore, by introducing the parameter adjustment rule based on evaluation and control of the search state of the cluster, the ability to adjust diversification and intensification of the cluster was improved, and the adaptability and search performance of AFA were improved. Furthermore, we illustrated the effects of adjusting search points within clusters and between clusters.

4.5 Summary

In this chapter, we proposed the diversification and intensification for the superior solution set search, and based on this search strategy , we proposed an adaptive FA based on the evaluation and control of the search state for the superior solution set search. The superior search performance of the proposed adaptive FA was verified by numerical experiments.

Table 4.7: Results of Numerical Experiments (F_1)

Dim.		AFA	FA-test	FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
					0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 5$	Mean	3.96 (15th)	3.96 (15th)	3.92 (16th)	1	2	2.02	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3.98	3.96	3.90
	Best	4 (1st)	4 (1st)	4 (1st)	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
	Worst	3 (14th)	3 (14th)	3 (14th)	1	2	3	4	4	4	4	4	4	4	4	4	4	4	4	4	4	3	3	3
	S.D.	0.19	0.20	0.27	0	0	0.14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.14	0.19	0.30
$n = 10$	Mean	3.58 (1st)	3.30 (1st)	3.20 (2nd)	1	2.26	3.36	3.02	1.86	1.4	0.96	0.5	0.36	0.22	0.16	0.08	0.08	0.04	0.04	0.02	0.04	0.02	0	0
	Best	4 (1st)	4 (1st)	4 (1st)	1	4	4	4	3	3	2	3	2	1	2	1	1	1	1	1	1	1	0	0
	Worst	3 (1st)	2 (2nd)	2 (2nd)	1	1	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.42	0.70	0.54	0	0.69	0.60	0.85	0.76	0.88	0.70	0.74	0.60	0.42	0.42	0.27	0.27	0.19	0.20	0.14	0.20	0.14	0	0
$n = 20$	Mean	3.20 (1st)	3 (2nd)	2.82 (3th)	1.82	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	4 (1st)	4 (1st)	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	2 (1st)	2 (1st)	2 (1st)	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.67	0.72	0.62	0.52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.8: Results of Numerical Experiments (F_2)

Dim.		AFA	FA-test	FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
					0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n=5$	Mean	4 (1st)	3.92 (2nd)	3.84 (3th)	1	1.20	1.32	1.66	2.04	1.92	1.98	1.72	1.58	1.56	1.40	1.16	1.12	1.06	1	1	1	1	1	1.02
	Best	4 (1st)	4 (1st)	4 (1st)	1	2	2	3	4	3	3	3	2	2	2	2	2	2	1	1	1	1	1	2
	Worst	4 (1st)	3 (2nd)	3 (2nd)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S.D.	0	0.24	0.37	0	0.40	0.47	0.59	0.53	0.53	0.53	0.43	0.50	0.50	0.50	0.49	0.37	0.33	0.24	0	0	0	0	0.14
$n=10$	Mean	3.86 (1st)	3.42 (2nd)	3.40 (3th)	0.26	0.3	0.76	0.7	0.34	0.28	0.12	0	0.02	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	4 (1st)	4 (1st)	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	Worst	3 (1st)	2 (2nd)	2 (2nd)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.48	0.70	0.61	0.44	0.46	0.43	0.46	0.48	0.45	0.33	0	0.14	0	0	0	0	0	0	0	0	0	0	0
$n=20$	Mean	2.36 (1st)	1.98 (3th)	1.32 (3th)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	3 (2nd)	2 (3th)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	1 (1st)	1 (1st)	0 (2nd)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.59	0.63	0.84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.9: Results of Numerical Experiments (F_3)

Dim.		AFA	FA-test	FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
					0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 5$	Mean	4 (1st)	3.94 (2nd)	3.80 (3th)	1.00	1.28	1.86	1.96	2.02	1.94	1.92	1.92	1.84	1.74	1.56	1.40	1.42	1.38	1.24	1.18	1.14	1.10	1.16	1.10
	Best	4 (1st)	4 (1st)	4 (1st)	1	2	3	3	4	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
	Worst	4 (1st)	3 (2nd)	2 (3th)	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S.D.	0	0.24	0.40	0	0.45	0.50	0.57	0.47	0.55	0.49	0.49	0.37	0.44	0.50	0.49	0.50	0.49	0.43	0.39	0.35	0.30	0.37	0.30
$n = 10$	Mean	3.58 (1st)	3.48 (2nd)	3.44 (2nd)	0.38	0.28	0.50	0.62	0.42	0.22	0.10	0.04	0.02	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	4 (1st)	4 (1st)	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
	Worst	3 (1st)	3 (1st)	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.57	0.62	0.54	0.49	0.45	0.51	0.49	0.50	0.42	0.30	0.20	0.02	0	0	0	0	0	0	0	0	0	0	0
$n = 20$	Mean	2.26 (1st)	1.86 (2nd)	1.78 (3th)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	3 (2nd)	2 (3th)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	0 (2nd)	1 (1st)	0 (2nd)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.85	0.63	0.46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.10: Results of Numerical Experiments (F_4)

Dim.		AFA	FA-test	FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
					0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 5$	Mean	4 (1st)	3.82 (2nd)	3.76 (3th)	1.00	1.32	1.92	1.96	2.12	2.02	1.90	1.86	1.82	1.70	1.56	1.42	1.40	1.38	1.24	1.16	1.14	1.10	1.16	1.10
	Best	4 (1st)	4 (1st)	4 (1st)	1	2	3	3	4	3	3	3	2	2	2	2	2	2	2	2	2	2	2	2
	Worst	4 (1st)	3 (2nd)	2 (3th)	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S.D.	0	0.62	0.42	0	0.50	0.52	0.60	0.45	0.53	0.49	0.49	0.35	0.44	0.48	0.45	0.49	0.43	0.43	0.39	0.35	0.30	0.37	0.30
$n = 10$	Mean	3.78 (1st)	3.54 (2nd)	3.32 (3th)	0.40	0.28	0.50	0.63	0.40	0.21	0.11	0.04	0	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	4 (1st)	4 (1st)	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	2 (1st)	2 (1st)	2 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.63	0.64	0.60	0.49	0.45	0.51	0.49	0.50	0.42	0.30	0.20	0	0	0	0	0	0	0	0	0	0	0	0
$n = 20$	Mean	2.02 (1st)	1.98 (2nd)	1.62 (3th)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Best	3 (1st)	3 (1st)	2 (2nd)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Worst	0 (1st)	0 (1st)	0 (1st)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.79	0.68	0.53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 4.11: Results of Numerical Experiments (Shekel's Function)

Dim.		AFA	FA-test	FA-CI	Original Firefly Algorithm (γ is fixed and varies between 0.05 and 1.0.)																			
					0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
$n = 2$	Mean	4 (1st)	3.92 (2nd)	3.84 (3th)	1	1.20	1.32	1.66	2.04	1.92	1.98	1.72	1.58	1.56	1.40	1.16	1.12	1.06	1	1	1	1	1	1.02
	Best	4 (1st)	4 (1st)	4 (1st)	1	2	2	3	4	3	3	3	2	2	2	2	2	2	1	1	1	1	1	2
	Worst	4 (1st)	3 (2nd)	3 (2nd)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	S.D.	0	0.24	0.37	0	0.40	0.47	0.59	0.53	0.53	0.53	0.43	0.50	0.50	0.50	0.49	0.37	0.33	0.24	0	0	0	0	0.14
$n = 5$	Mean	3.86 (1st)	3.42 (2nd)	3.40 (3th)	0.26	0.3	0.76	0.7	0.34	0.28	0.12	0	0.02	0	0	0	0	0	0	0	0	0	0	0
	Best	4 (1st)	4 (1st)	4 (1st)	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0
	Worst	3 (1st)	2 (2nd)	2 (2nd)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	S.D.	0.48	0.70	0.61	0.44	0.46	0.43	0.46	0.48	0.45	0.33	0	0.14	0	0	0	0	0	0	0	0	0	0	0

5 Superior Relation Based Superior Solution Set Search Method

5.1 Introduction

In this Chapter, we analyze the properties of the superior solution set search problem, and we point out the structural similarities between the superior solution set search problem and multi-objective optimization problem. Based on the analyzed properties, we propose an superior solution fitness, which is an index that is inspired by a method based on superior relation in multi-objective optimization problems and includes parameters to judge the goodness of the search point. We propose an Firefly Algorithm (FA) with this index and an Genetic Algorithm (GA) [32] with this index for the superior solution set problem. Numerical experiments are then conducted using the superior solution set search problem, and the usefulness of the proposed methods are demonstrated while comparing the performance of the proposed methods with the conventional FA and the conventional GA.

5.2 Analysis and New Interpretation of Superior Solution Set Search Problem

5.2.1 Analysis Interpretation of Superior Solution Set Search Problem

It seems that the foundation of heuristic approximate optimization methods is a structure that finds even better solution by using information regarding a desirable solution. This is a major premise that is used to compare the solutions; single objective optimization solution is compared using some evaluation values, and multi-objective optimization solutions are compared using superior relations. In multi-objective optimization, a strategy [76, 77] has been proposed to converge the entire search point to the Pareto frontier by comparing search points using superior relations and passing superior search points to the next generation of solutions. The superiority of the solution in multi-objective optimization is determined using superior relation in K objective functions $\boldsymbol{f} : \boldsymbol{p} \in \mathbb{R}^N \mapsto [f_1, f_2, \cdots, f_K]^\top \in \mathbb{R}^K$. When satisfying the following Eq.(5.1), the solution $\boldsymbol{p}$ dominates the solution $\boldsymbol{q}$ ($\boldsymbol{p} \prec \boldsymbol{q}$).

$$\boldsymbol{p} \prec \boldsymbol{q} \Leftrightarrow \forall k, f_k(\boldsymbol{p}) \leq f_k(\boldsymbol{q}) \wedge \exists k, f_k(\boldsymbol{p}) < f_k(\boldsymbol{q}) \tag{5.1}$$

The optimal solution in multi-objective optimization is defined as a Pareto solution, which is a solution that is not superior to all other solutions.

In order to meet the needs of practical applications, the superior solution set is defined as "the difference between the evaluation values and the global optimal solution falls within a certain range" and "the distance from other local optimal solutions that is greater than a certain distance" (please refer to the Eqs.(2.11), (2.12), and (2.13)). Here, a constraint on the evaluation value provided by the user is $\delta \geq 0$. A constrain on the distance between the solutions is $\varepsilon > 0$, which consists of local optimal solutions that satisfy the condition "the difference between the evaluation value and the global optimal solution falls within δ" and "the distance to other local optimal solution is more than ε". This method is based on the aforementioned properties. Since relationships for comparing solutions are also defined in the definition of a superior solution set, there is a possibility that a method (superior relationship based on the evaluation value and distance) similar to that used in multi-objective optimization may be proposed, even when searching for the superior solution set.

The superior solution set search problem is similar to the multi-objective optimization problem from the definition of the solution set. Specifically, the superior solution and the Pareto solution have the following properties in common.

(1) It is defined by the relationship (allowing incomparable cases) for comparing solutions.
(2) It is defined as a solution for which the above relationship does not hold with all feasible solutions.
(3) One or more optimal solutions can always be defined in any non-empty subset $\boldsymbol{Y} \subseteq \boldsymbol{X}$ of the feasible region $\boldsymbol{X}$.

In multi-objective optimization, a strategy has been proposed in which the search points are compared according to the superiority relationship, and the excellent search points are positively left for the next generation, so that the entire search points are converged to the Pareto frontier. Based method). This method is based on the above properties. On the other hand, since the relationship for comparing solutions is defined in the definition of the superior solution set, a method equivalent to the method based on the superior relation in multi-objective optimization can be proposed even in the search of the superior solution set.

5.2.2 New Interpretation of Superior Solution Set Search Problem

The superior solution set was defined for the first time in section **??**. However, two new superior relations for the superior solution set are defined in order to analyze and exploit the superior solution set search problem described below, which are not mentioned in the definition of the ranking operation based on superior relation (see the Eq.(5.1)) in multi-objective optimization.

Definition 5.1 (**Relationship Definition Using Constraints on Evaluation Value**) The following relation is defined in order to capture constraints on the evaluation value as the following Eq.(5.2). For two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2 \in \boldsymbol{X}$, when $\boldsymbol{x}_1$ excels under the constraints of evaluation value than $\boldsymbol{x}_2$ ($\boldsymbol{x}_1 \prec_\delta \boldsymbol{x}_2$).

$$\boldsymbol{x}_1 \prec_\delta \boldsymbol{x}_2 \Leftrightarrow f(\boldsymbol{x}_1) + \delta < f(\boldsymbol{x}_2) \tag{5.2}$$

□

$\boldsymbol{x} \in \boldsymbol{X}$ is inferior to all feasible solutions using the above relationship. In other words, when satisfying $\boldsymbol{y} \not\prec_\delta \boldsymbol{x}$ ($\forall \boldsymbol{y} \in \boldsymbol{X}$), $\boldsymbol{x}$ is a solution when the difference between the evaluation value and the global optimal solution is within δ. This corresponds to the requirement that "a solution's evaluation value is within a certain range of the global optimum solution"

Definition 5.2 (**Relationship Definition Using Constraints on Distance**) To capture the constraints on the distance between the solutions, we define the distance function $dist(\cdot,\cdot)$ as the following Eq.(5.3).

$$\mathcal{N}(\boldsymbol{x};\varepsilon) = \{\boldsymbol{y} \in \mathbb{R}^N \mid dist(\boldsymbol{x},\boldsymbol{y}) < \varepsilon\} \tag{5.3}$$

For the two solutions $\boldsymbol{x}_1$, $\boldsymbol{x}_2 \in \boldsymbol{X}$, when $\boldsymbol{x}_1$ excels under the constraints of distance than $\boldsymbol{x}_2$ ($\boldsymbol{x}_1 \prec_\varepsilon \boldsymbol{x}_2$).

$$\boldsymbol{x}_1 \prec_\varepsilon \boldsymbol{x}_2 \Leftrightarrow f(\boldsymbol{x}_1) < f(\boldsymbol{x}_2) \wedge \boldsymbol{x}_1 \in \mathcal{N}(\boldsymbol{x}_2;\varepsilon) \tag{5.4}$$

□

$\boldsymbol{x} \in \boldsymbol{X}$ is inferior to all feasible solutions using the above relationship, i.e., when satisfying $\forall \boldsymbol{y} \in \boldsymbol{X}$, $\boldsymbol{y} \not\prec_\varepsilon \boldsymbol{x}$, $\boldsymbol{x}$ is a local optimal solution in which there is no superior solution closer than the distance ε. This corresponds is "a local optimal solution more than a certain distance apart". Hereafter, Euclidean distance is used as the distance.

Definition 5.3 (**New Interpretation of Superior Solution Set** $\mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)$)
Using the relationships defined so far, the superior solution set $\mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)$ is defined as the following Eq.(5.5).

$$\mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)=\{\boldsymbol{x}^{\star}\in\boldsymbol{X}\mid\forall\boldsymbol{x}\in\boldsymbol{X},\boldsymbol{x}\not\prec_{\varepsilon}\boldsymbol{x}^{\star}\wedge\boldsymbol{x}\not\prec_{\delta}\boldsymbol{x}^{\star}\}\tag{5.5}$$

□

The superior solution set $\mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)$ is a set of local optimal solutions, where the difference between the evaluation value and the global optimal solution within δ, and other solutions are at least some distance away from ε. We formulate a problem for determining the superior solution set $\mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)$ as a superior solution set search problem.

5.3 Proposal of Superior Relation in Superior Solution Set Search Problem

In proposing a search method based on the superior solution fitness, we pay attention to multi-objective optimization. The multi-objective optimization method that can consider multiple purposes at the same time and is used to support user decision-making. In this respect, it can be said that this characteristic has a high affinity with the superior solution set search problem. There are two types of multi-objective optimization methods: a method that searches for a unique solution by acquiring and using user preference information in advance, and a discovery approximation method that obtains multiple and diverse solution sets. The discovery approximation method of multi-objective optimization method corresponds to the latter, and is being actively researched as a promising optimization method. We focused on this evolutionary multi-objective optimization method from the viewpoint of affinity with the purpose of the

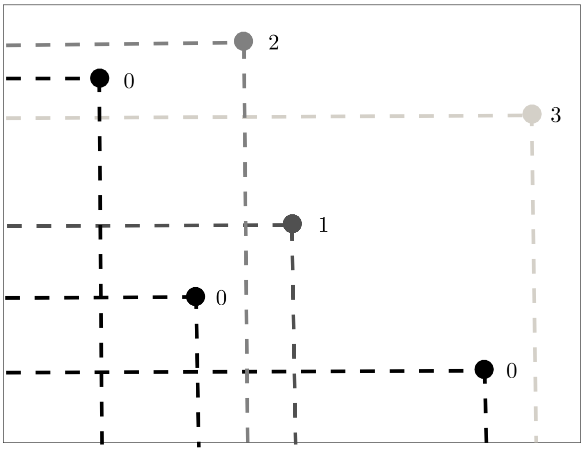


Fig. 5.1: Fitness Ranking in Multi-objective Optimization Problem

superior solution set search problem of acquiring multiple and diverse solution sets. Multi-objective optimization methods can be broadly classified into three types. There are three methods: "method based on superior superior relation", "method based on division", and "method based on Indicator". Above all, this time, we focus on the method based on superior relation. Typical methods based on superior relation include NSGA-II [76] and SPEA2 [77]. In these methods, a non-inferior solution set is searched by giving an ranking relation according to superior relation or inferiority to the search points and performing survival selection using the superior relation. An image of fitness ranking in multi-objective optimization problem is shown in Fig. 5.1.

In this paper, we propose an optimization method for the superior solution set search problem that explicitly includes the parameters (δ, ε). Specifically, first, the superior solution fitness is defined as an index that gives the superior relation of the solution using δ and ε. Next, we propose a superior solution set search method that follows the superior solution fitness.

5.3.1 Definition of Superior Solution Fitness

A superior solution fitness is defined utilizing the superior solution set search problem-specific parameters δ and ε and knowledge of multi-objective optimization. The superior solution fitness that gives superior relation to the solutions using the superior solution set search problem-specific parameters δ and ε based on the definitions in the superior solution set search problem is shown below.

Definition 5.4 (**Superior Solution Fitness** $fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$) Let $\boldsymbol{P}$ be the set of search points. Based on the definitions of the sets (definition of relations using constraints on evaluation values and definition of relations using distance constraints) for each search point $\boldsymbol{x} \in \boldsymbol{P}$, let superior solution fitness be the number of search points $\boldsymbol{y} \in \boldsymbol{P}$ that are superior to $\boldsymbol{x}$.

$$fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon) := \mathbf{Card}\{\boldsymbol{y} \in \boldsymbol{P} \mid \boldsymbol{y} \prec_\delta \boldsymbol{x} \vee \boldsymbol{y} \prec_\varepsilon \boldsymbol{x},\ \boldsymbol{y} \neq \boldsymbol{x} \in \boldsymbol{P}\} \tag{5.6}$$

Here, Card($\boldsymbol{A}$) is the cardinality of a finite set $\boldsymbol{A}$, which is the number of elements of a finite set. The above equation corresponds to the case where "the number of search points superior to itself" is given as the light intensity in multi-objective optimization. However, if there are duplicate search points, duplicate search points excluding one search point are counted as $f(\boldsymbol{x}) = \infty$. The smaller the value of superior solution fitness $fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$, the better the solution. When the superior solution fitness $fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$ values of two solutions are equal, the solution with the smaller evaluation is chosen (in minimazation problem). □

$fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$ includes parameters (δ, ε) specific to the superior solution set search problem in its definition. Therefore, the research task of utilizing parameters is achieved. This $fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$ gives the solutions a superior relation. Utilizing the knowledge of multi-objective optimization, we can propose a superior solution set search method based on this $fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$.

5.4 Superior Relation Based Firefly Algorithm for Superior Solution Set Search Problem

5.4.1 Proposal of Superior Relation Based Firefly Algorithm

From the discussions so far, it has been clarified that the superior solution set and the Pareto solution set share many properties. The multi-objective optimization method based on the superior relation is expected to converge to the Pareto frontier of the entire search point set by performing the ranking operation based on the superior relation. The method based on the superior relation solves the multi-objective optimization problem by utilizing the properties of the superior relation and the Pareto solution. On the other hand, the superior solution set is similar to the Pareto solution set, it is considered possible to apply the method based on the superior relation to the superior solution set search.

Based on Section 5.3.1, this paper proposes a superior solution set search method as a fundamental examination to search for a superior solution set based on superior relations. In FA, light intensity $\big($see the Eq.(3.23)$\big)$ is defined in terms of the objective function value. However, in the superior solution set searching method, the light intensity was defined based on relationships $(\prec_\delta, \prec_\varepsilon)$, such as the superior relation in multi-objective optimization. In the above definition, we are trying to improve the performance of the superior solution set search problem by ranking the search points.

On the other hand, in the superior solution set searching method based on the light intensity considering the distance, the light intensity was defined based on the relationships $(\prec_\delta, \prec_\varepsilon)$, such as the superior relation of the multi-objective optimization method. In the above definition, we are trying to improve the performance of superior solution set search problem by ranking the search points while strengthening the analogy of fireflies by the light intensity considering the distance.

In the proposed method, when ranking based on the relationships $(\prec_\delta, \prec_\varepsilon)$, we define the light intensity $I(\boldsymbol{x} \in \boldsymbol{P})$ of the search point $\boldsymbol{x} \in \boldsymbol{P} \subseteq \mathbb{R}^N$ according to the superior

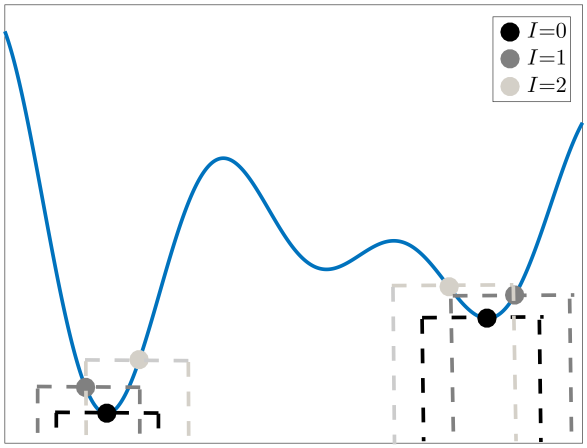


Fig. 5.2: Light Intensity Ranking in Superior Solution Set Search Problem

solution fitness $fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$ (see the Eq.(5.6)) by the Eq.(5.7).

$$I(\boldsymbol{x} \in \boldsymbol{P}) = fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon) \tag{5.7}$$

In multi-objective optimization based on superior relations, superiority and inferiority are compared using a contribution degree for maintaining diversity when there is no difference in the superior relations. However, in the superior solution set search, the relationship $\prec_\varepsilon$ already has embedded functions for maintaining diversity. Furthermore, since the superior solution set search problem is based on single objective optimization, you can judge superiority with the absolute evaluation value $|f(\boldsymbol{x})|$ if there is no difference in I.

This paper calculates the light intensity I of the search point using superior relations. An image of ranking the light intensity I in the search process is shown in Fig. 5.2. Then, the proposed method generates the best search point set $\boldsymbol{C}$ based on the information of the search point group $\boldsymbol{P}$ in each iteration, and shows it in the Eq.(5.8).

$$\boldsymbol{C} = \{\boldsymbol{x}_i \in \boldsymbol{P} \mid I(\boldsymbol{x}_i \in \boldsymbol{P}) = 0,\ i = 1, 2, \cdots, m\} \tag{5.8}$$

While advancing the search, the best search point set $\boldsymbol{C}$ gradually approaches the superior solution set $\mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)$. Here, not only all the best search point $\boldsymbol{x} \in \boldsymbol{C}$ are always $I(\boldsymbol{x} \in \boldsymbol{C}) = 0$, but all superior solutions $\hat{\boldsymbol{x}} \in \mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)$ are always $I(\hat{\boldsymbol{x}} \in \boldsymbol{X}) = 0$.

Superior relation based FA for the superior solution set search problem of the objective function $f(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathbb{R}^N)$ is shown by **Algrithm 5.1**.

Algorithm 5.1 Superior Relation Based Firefly Algorithm (SR-FA)

procedure SR-FA$(m, \alpha, \beta_0, \gamma, T_{\max}, \delta, \varepsilon)$

Step 1: Preparation

Set the maximum number of iterations $T_{\max}$, the number of search points m, and the parameters $\alpha > 0$, $\beta_0 > 0$, and $\gamma > 0$.

Set the parameters $\delta \geq 0$ and $\varepsilon \geq 0$ of superior solution set search problem.

Step 2: Initialization

Set the number of iterations $t = 1$.

In the feasible area $\boldsymbol{X} \subseteq \mathbb{R}^n$, randomly generate search point $\boldsymbol{x}_i^1$ $(i = 1, 2, \cdots, m)$, and the swarm set $\boldsymbol{P}^1 = \{\boldsymbol{x}_i^1 \mid i = 1, 2, \cdots, m\}$.

Step 3: Calculation of light intensity

Calculate the light intensity $I(\boldsymbol{x}_i^t \in \boldsymbol{P}^t)$ of each search point $\boldsymbol{x}_i^t$ and the best search point set $\boldsymbol{C}^t$ following equations.

$I(\boldsymbol{x}_i^t \in \boldsymbol{P}^t) = \mathrm{Card}\{\boldsymbol{x} \in \boldsymbol{P}^t \mid \boldsymbol{x} \prec_\delta \boldsymbol{x}_i^t \vee \boldsymbol{x} \prec_\varepsilon \boldsymbol{x}_i^t, \boldsymbol{x} \neq \boldsymbol{x}_i^t \in \boldsymbol{P}^t\}$

$\boldsymbol{C}^t = \{\boldsymbol{x}_i^t \in \boldsymbol{P}^t \mid I(\boldsymbol{x}_i^t \in \boldsymbol{P}^t) = 0,\ i = 1, 2, \cdots, m\}$

Each search point $\boldsymbol{x}_i^t \in \boldsymbol{P}^t$ is sorted in increasing order of $I(\boldsymbol{x}_i^t \in \boldsymbol{P}^t)$.

Save the reference solution $\boldsymbol{z}_i$ and solution set $\boldsymbol{V}$ following equations.

$\boldsymbol{z}_i = \boldsymbol{x}_i^t$ $(i = 1, 2, \cdots, m)$, $\boldsymbol{V} = \{\boldsymbol{z}_i \mid i = 1, 2, \cdots, m\}$

Set $i = 1$ and $j = 1$.

Step 4: Movement of search points

for $i = 1$ to m **do**

  if $I(\boldsymbol{x}_i^t \in \boldsymbol{P}^t) < I(\boldsymbol{x}_j^t \in \boldsymbol{P}^t)$ **then**

    Move the search point $\boldsymbol{x}_i^t \in \boldsymbol{P}^t$ referring solution $\boldsymbol{z}_j \in \boldsymbol{V}$ by the following equation.

    $\boldsymbol{x}_i^t := \boldsymbol{x}_i^t + \beta_0 e^{-\gamma \|\boldsymbol{z}_j - \boldsymbol{x}_i^t\|^2} (\boldsymbol{z}_j - \boldsymbol{x}_i^t) + \alpha \boldsymbol{R}$

    Here, $\boldsymbol{R} \in [-0.5, 0.5]^n$ is a uniform random vector.

    Let $j := j + 1$, until $j = m$ and move the search point $\boldsymbol{x}_{cbest}^t \in \boldsymbol{C}^t$ according to the following formula.

    $\boldsymbol{x}_{cbest}^t := \boldsymbol{x}_{cbest}^t + \alpha \boldsymbol{R}$

  Set $j := 1$.

end for

Step 5: Updating search points

Let $\boldsymbol{U} = \boldsymbol{P}^t \cup \boldsymbol{V}$, and calculate the light intensity $I(\boldsymbol{u}_s \in \boldsymbol{U})$ $(s = 1, 2, \cdots, 2m)$ and the elite solution set $\boldsymbol{B}$ according to the following equations.

$I(\boldsymbol{u}_s \in \boldsymbol{U}) = \mathrm{Card}\{\boldsymbol{y} \in \boldsymbol{U} \mid \boldsymbol{y} \prec_\delta \boldsymbol{u}_s \vee \boldsymbol{y} \prec_\varepsilon \boldsymbol{u}_s, \boldsymbol{y} \neq \boldsymbol{u}_s \in \boldsymbol{U}\}$

$\boldsymbol{B} = \{\boldsymbol{u}_s \in \boldsymbol{U} \mid rank_s \leq m,\ s = 1, 2, \cdots, 2m\}$

Here, $rank_s$ is the rank of solution based on $I(\boldsymbol{u}_s \in \boldsymbol{U})$.

Update the search point $\boldsymbol{x}_i^t$ to elite solutions $\boldsymbol{u}_i \in \boldsymbol{B}$ and the swarm set $\boldsymbol{P}^t$ according to following equations.

$\boldsymbol{x}_i^{t+1} = \boldsymbol{u}_i \in \boldsymbol{B}$ $(i = 1, 2, \cdots, m)$, $\boldsymbol{P}^{t+1} = \{\boldsymbol{x}_i^{t+1} \mid i = 1, 2, \cdots, m\}$

Step 6: Termination

if $t = T_{\max}$ **then**

  The algorithm is terminated.

else

  Return to **Step 4**, set $t := t + 1$.

end procedure

5.4.2 Numerical Experiment

(a) Numerical Experiment Conditions

Through numerical experiments, we compare the superior relation based FA (proposed method) with the original FA to evaluate its properties and performance for the superior solution set search problem. Please refer to Shekel's function in **Appendix A** for details of the benchmark function used in this experiments.

In this paper, we set the usable situation of benchmark function to verify the temporary setup with the proposed method, which is tentatively set up to correspond to the user's desired level. There are six types of (δ, ε) and superior solution set $\mathcal{Q}(\boldsymbol{X};\delta,\varepsilon)$, where ($\delta$, ε)=(7.5, 1), (7.5, 2), (8.5, 2), (10, 1), (10, 2), and (10, 5.5). We regard them as Cond. 1 $\sim$ Cond. 6. The criteria for setting parameters (δ, ε) are explained as follow.

- In the case of relatively allowing solution proximity and not allowing deterioration of evaluation value (Cond. 1)
- In the case of requiring relatively large solution diversity and allowing deterioration of the evaluation value (Cond. 4)
- In the case of requiring moderate diversity and evaluation values in the above two cases (Cond. 2, 3, 5, and 6)

The common conditions are as follows: the number of search points $m = 60$; and FA patemeters $\alpha = 0.05$, $\beta_0 = 1.0$; the maximum iteration time $T_{\max} = 100$; dimension $N = 2$ and 5. The initial solution was set randomly within the executable area $[-5,\ 5]^N$. The proposed method uses $\gamma = 1$ according to recommended parameter values, while the original FA uses the best γ of 20 types in the range $\gamma = 0.1, 0.2, \cdots, 2.0$.

In addition, we also did numerical experiments to compare AFA and the superior relation based FA on the Shekel's Function for 2, and 5 dimensions. The parameters of AFA were $C_{\max} = 10^{-6}$, $C_{\min} = 10^{-8}$, $\Delta C = 10^{-8}$ referring to points in same cluster, $P_{\max} = 10^{-8}$, $P_{\min} = 10^{-10}$, $\Delta P = 10^{-10}$ referring to points in different clusters, cluster number $K = 6$, $\theta = 0.3$, and $T = 5$. The number of trials was set to 50. The maximum number of iterations was 1000 and η was 0.5. In addition, the recommended values of other parameters β_0 was 1.0, $\alpha_{\max}$ was 1.0 and $\alpha_{\min}$ was 0.01. The other conditions for the original FA, FA-CI, FA-test, and AFA were shown below.

(b) Evaluation Index

In this numerical experiments, we evaluate the search performance of the superior solution using two evaluation indices Peak Ratio (PR) [78] and Convergence Ratio (CR).

Peak Ratio: PR is an index for evaluating the ratio of the acquired superior solution proposed by the authors and is expressed by the Eq.(5.9).

$$PR = \mathrm{Card}\{\boldsymbol{h}_g \mid ||\boldsymbol{h}_g - \hat{\boldsymbol{x}}_g|| \leq \eta,\ g = 1, 2, \cdots, G\} \tag{5.9}$$

Here, $\eta > 0$ is a threshold parameter, we set $\eta = 10^{-1}$ in all conditions, $\hat{\boldsymbol{x}}_g \in \mathcal{Q}(\boldsymbol{X}; \delta, \varepsilon)$ $(g = 1, 2, \cdots, G)$ is the superior solution, $G = \mathrm{Card}\big(\mathcal{Q}(\boldsymbol{X}; \delta, \varepsilon)\big)$ is the number of superior solutions, $\boldsymbol{h}_g$ is the search point $\boldsymbol{x}_i^{T\max} \in \boldsymbol{P}^{T\max}$ locating the nearest by $\hat{\boldsymbol{x}}_g$ at the last generation following by the Eq.(5.10).

$$\boldsymbol{h}_g = \mathop{\arg\min}_{\boldsymbol{x}_i^{T\max} \in \boldsymbol{P}^{T\max}} ||\boldsymbol{x}_i^{T\max} - \hat{\boldsymbol{x}}_g|| \tag{5.10}$$

PR indicates that it is assumed that the superior solution $\hat{\boldsymbol{x}}_g$ has been acquired if $\boldsymbol{h}_g$ exists satisfying $||\boldsymbol{h}_g - \hat{\boldsymbol{x}}_g|| \leq \eta$.

Convergence Ratio: CR is an index for evaluating convergence ratio of superior solution and is expressed by the Eq.(5.11).

$$CR = \frac{1}{G} \sum_{g=1}^{G} ||\boldsymbol{h}_g - \hat{\boldsymbol{x}}_g|| \tag{5.11}$$

CR has a large value if there are search points converging to the solutions that do not sufficiently converge to the superior solutions or is not the superior solution.

The mean (Mean) and the standard deviation (S.D.) of each index are compared after 50 runtimes using different initial solutions. The indices are described as follows:

- PR_{Mean} and $PR_{\text{S.D.}}$: The mean and standard deviation of PR.
- CR_{Mean} and $CR_{\text{S.D.}}$: The mean and standard deviation of CR.

(c) Results of the Numerical Experiment

Table 5.1 shows the results of numerical experiments to compare the proposed method with the original FA. The best of PR_{Mean} and CR_{Mean} are shown in bold. In Figs. 5.3∼5.8 show the search transition in the proposed method in six types of (δ, ε) in two-dimensional problem. In Figs. 5.3∼5.8, the ★ in the figure is the superior solution, while ◯ indicates the search point.

In all conditions, the proposed method shows higher PR values than the original FA. That is, the proposed method has found more superior solutions than the original FA. Moreover, under many initial conditions, the proposed method shows smaller CR values than the original FA. That is, in the original FA, many search points converge to the superior solution when the search point convergence of is not sufficient or is not the superior solution, whereas the proposed method is one in which many search points can be superior solutions. In this way, since the proposed method shows excellent results both in terms of PR and CR values for the superior solution, the proposed method acquires more superior solutions, and it can be said that the search point does not converge to any local optimal solution other than the superior solution. This can also be inferred from the transition of the search point. In the proposed method, the converged solution varies with changes in the parameters δ and ε, and many search points converge to the superior solution to be acquired.

Also, Table 5.2 shows the results of numerical experiments to compare the proposed method with AFA. The best of PR_{Mean} and CR_{Mean} are shown in bold. In all conditions, the proposed method shows higher PR values than the AFA. That is, the proposed method has found more superior solutions than the AFA. Moreover, under many initial conditions, the proposed method shows smaller CR values than the AFA. That is, in the AFA, many search points converge to the superior solution when the search point convergence of is not sufficient or is not the superior solution, whereas the proposed method is one in which many search points can be superior solutions. In this way, since the proposed method shows excellent results both in terms of PR and CR values for the superior solution, the proposed method acquires more superior solutions, and it can be said that the search point does not converge to any local optimal solution other than the superior solution.

(d) Discussion

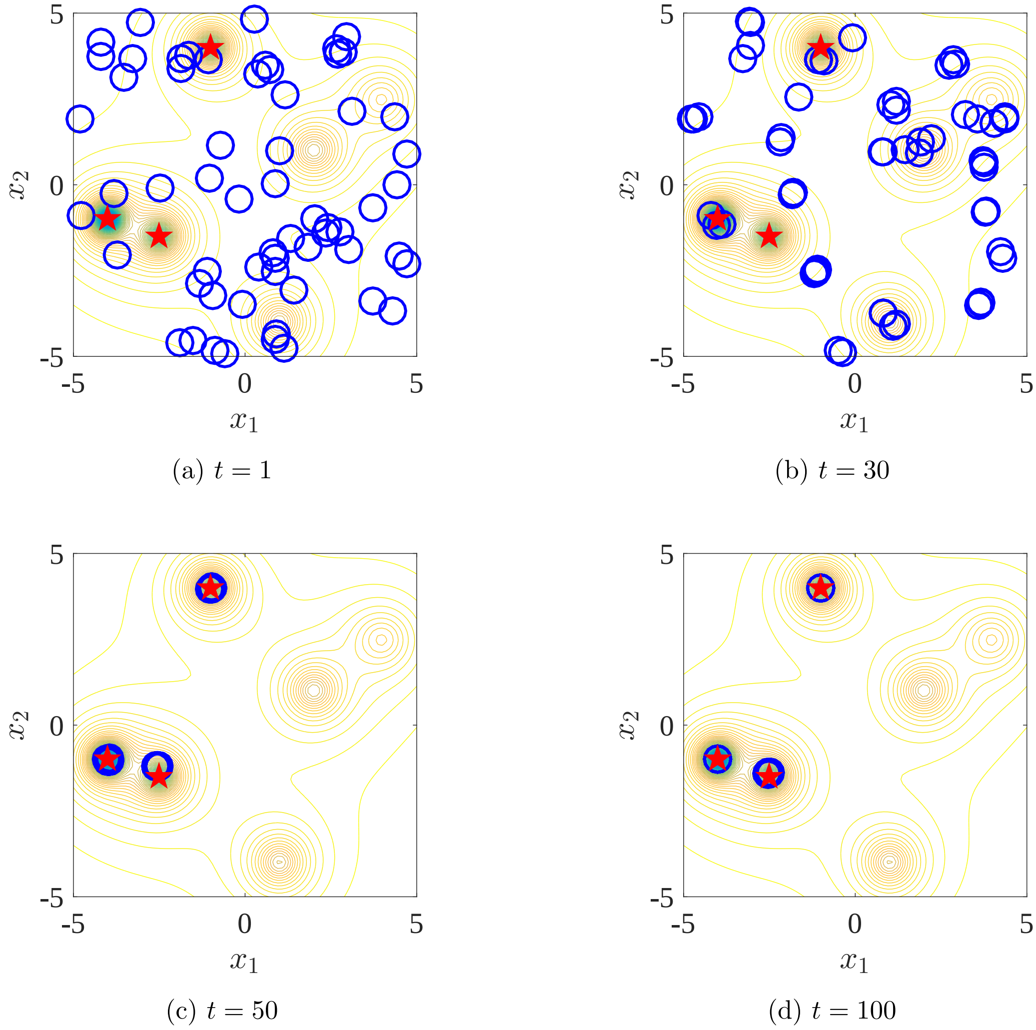
From the numerical experiment results, it can be seen that the original FA has lower search performance than the proposed method in the two-dimensional problem and five-dimensional problem. In addition, the original FA has much lower search performance in the five-dimensional problem than the two-dimensional problem. This is considered to be caused by the fact that the original FA moves the search point using Euclidean distance. That is, the original FA using Euclidean distance originally has a problem that the search performance is degraded for high dimensional problems. The proposed method also has lower search performance in the five-dimensional problem than the two-dimensional problem, but the decrease in performance is lower than in the original FA. Since the proposed method is based on the original FA, the Euclidean distance is used to move the search point. Furthermore, in the proposed method, the "superior relations" for strictly considering the parameters δ and ε that determine the superior solution set is evaluated based on the Euclidean distance. This is considered to be the cause of the decrease in the search performance in the high-dimensional problem of the proposed method as in the original FA. In order to improve the decrease in search performance in high-dimensional problems, improvement such as using something other than Euclidean distance for evaluation of the movement of the search point and the superior relations will be required.

Table 5.1: Experiment Results

Dimension	Problem				Proposed Method				Firefly Algorithm			
	Condition	δ	ε	Superior Solution	PR_{Mean}	$PR_{\text{S.D.}}$	CR_{Mean}	$CR_{\text{S.D.}}$	PR_{Mean}	$PR_{\text{S.D.}}$	CR_{Mean}	$CR_{\text{S.D.}}$
$N = 2$	Cond. 1	7.5	1	A, B, C	**3**	0	**0.0041**	0.0002	1.92	0.274	0.3406	0.1543
	Cond. 2	7.5	2	A, C	**2**	0	**0.0017**	0.0002	1.92	0.274	0.1581	0.3068
	Cond. 3	8.5	2	A, C, D, E	**4**	0	**0.0025**	0.0001	3.68	0.4712	0.2488	0.0676
	Cond. 4	10	1	A, B, C, D, E, F	**6**	0	**0.0120**	0.0011	3.92	0.274	0.4899	0.0752
	Cond. 5	10	2	A, C, D, E, F	**5**	0	**0.0094**	0.0002	3.88	0.3283	0.4644	0.0858
	Cond. 6	10	5.5	A, C, D, F	**4**	0	**0.0067**	0.0007	3	0.2857	0.6466	0.6185
$N = 5$	Cond. 1	7.5	1	A, B, C	**2**	0.699	**1.0355**	0.166	0.32	0.4712	3.3811	1.3833
	Cond. 2	7.5	2	A, C	**1**	0.8081	**1.812**	0.8871	0.15	0.3663	3.3205	1.6249
	Cond. 3	8.5	2	A, C, D, E	**1.36**	0.8981	**2.2083**	0.8965	0.625	0.5	4.0629	0.6614
	Cond. 4	10	1	A, B, C, D, E, F	**3.16**	0.9479	**1.9342**	0.6599	1	0	3.6909	0.4523
	Cond. 5	10	2	A, C, D, E, F	**2.26**	0.6642	**1.225**	1.0071	0.62	0.5488	3.9539	0.6468
	Cond. 6	10	5.5	A, C, D, F	**2.34**	0.7174	**1.6431**	0.6358	0.25	0.5	4.176	0.5853

Table 5.2: Experiment Results

Dimension	Problem				Proposed Method				AFA			
	Condition	δ	ε	Superior Solution	PR_{Mean}	$PR_{\text{S.D.}}$	CR_{Mean}	$CR_{\text{S.D.}}$	PR_{Mean}	$PR_{\text{S.D.}}$	CR_{Mean}	$CR_{\text{S.D.}}$
$N = 2$	Cond. 1	7.5	1	A, B, C	**3**	0	**0.0041**	0.0002	2.02	0.123	0.3821	0.1687
	Cond. 2	7.5	2	A, C	**2**	0	**0.0017**	0.0002	1.90	0.321	0.1842	0.2975
	Cond. 3	8.5	2	A, C, D, E	**4**	0	**0.0025**	0.0001	3.88	0.3323	0.2345	0.0576
	Cond. 4	10	1	A, B, C, D, E, F	**6**	0	**0.0120**	0.0011	3.90	0.235	0.3985	0.0687
	Cond. 5	10	2	A, C, D, E, F	**5**	0	**0.0094**	0.0002	3.70	0.2395	0.3743	0.0952
	Cond. 6	10	5.5	A, C, D, F	**4**	0	**0.0067**	0.0007	2.88	0.3237	0.6325	0.5154
$N = 5$	Cond. 1	7.5	1	A, B, C	**2**	0.699	**1.0355**	0.166	1.32	0.5213	2.9812	1.5634
	Cond. 2	7.5	2	A, C	**1**	0.8081	**1.812**	0.8871	1	0.3563	2.523	1.4923
	Cond. 3	8.5	2	A, C, D, E	**1.36**	0.8981	**2.2083**	0.8965	1.02	0.3202	2.0923	0.3212
	Cond. 4	10	1	A, B, C, D, E, F	**3.16**	0.9479	**1.9342**	0.6599	2.32	0.2351	2.2301	0.3255
	Cond. 5	10	2	A, C, D, E, F	**2.26**	0.6642	**1.225**	1.0071	1.36	0.8536	1.3762	0.2485
	Cond. 6	10	5.5	A, C, D, F	**2.34**	0.7174	**1.6431**	0.6358	2.02	1.2323	1.2373	0.3823



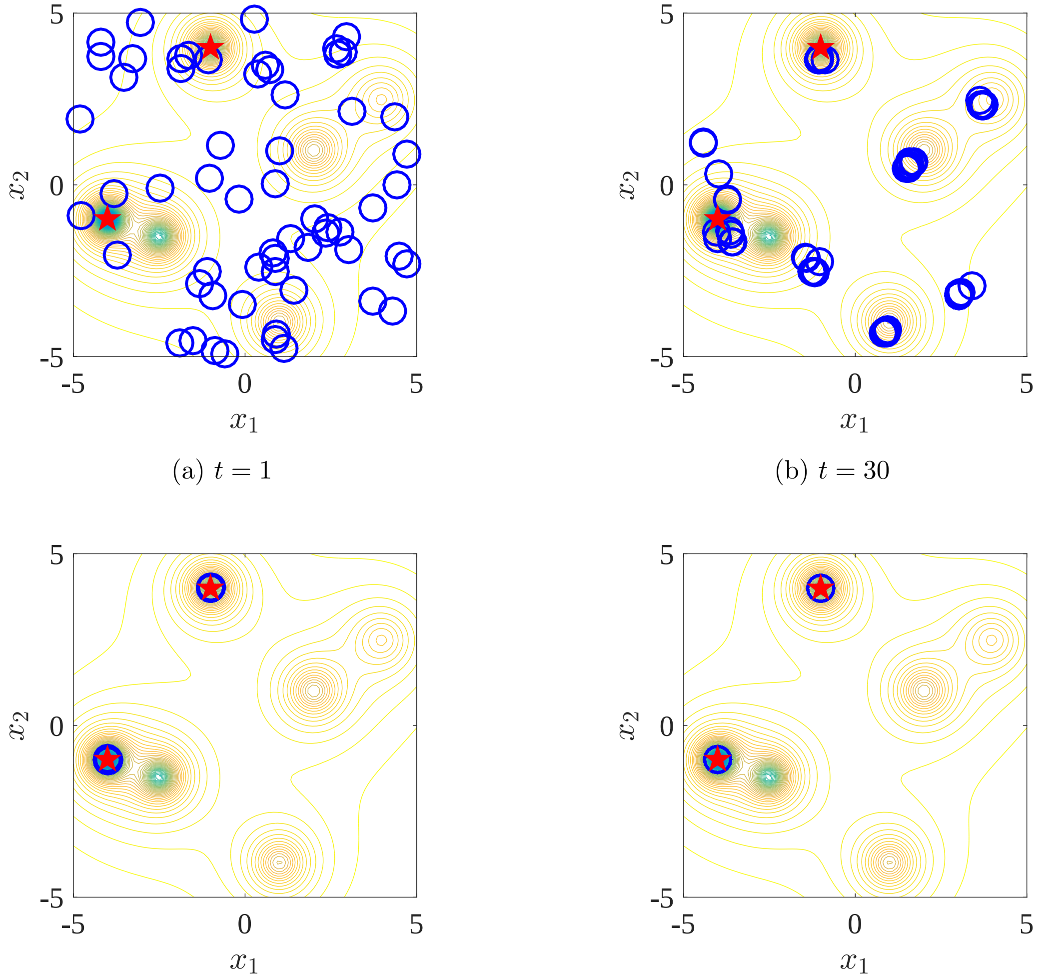
(a) $t = 1$

(b) $t = 30$

(c) $t = 50$

(d) $t = 100$

Fig. 5.3: Transition of Search Points for Search State of Cond. 1 (δ, ε= $7.5, 1$) in Superior Relation Based Firefly Algorithm ($N = 2$)



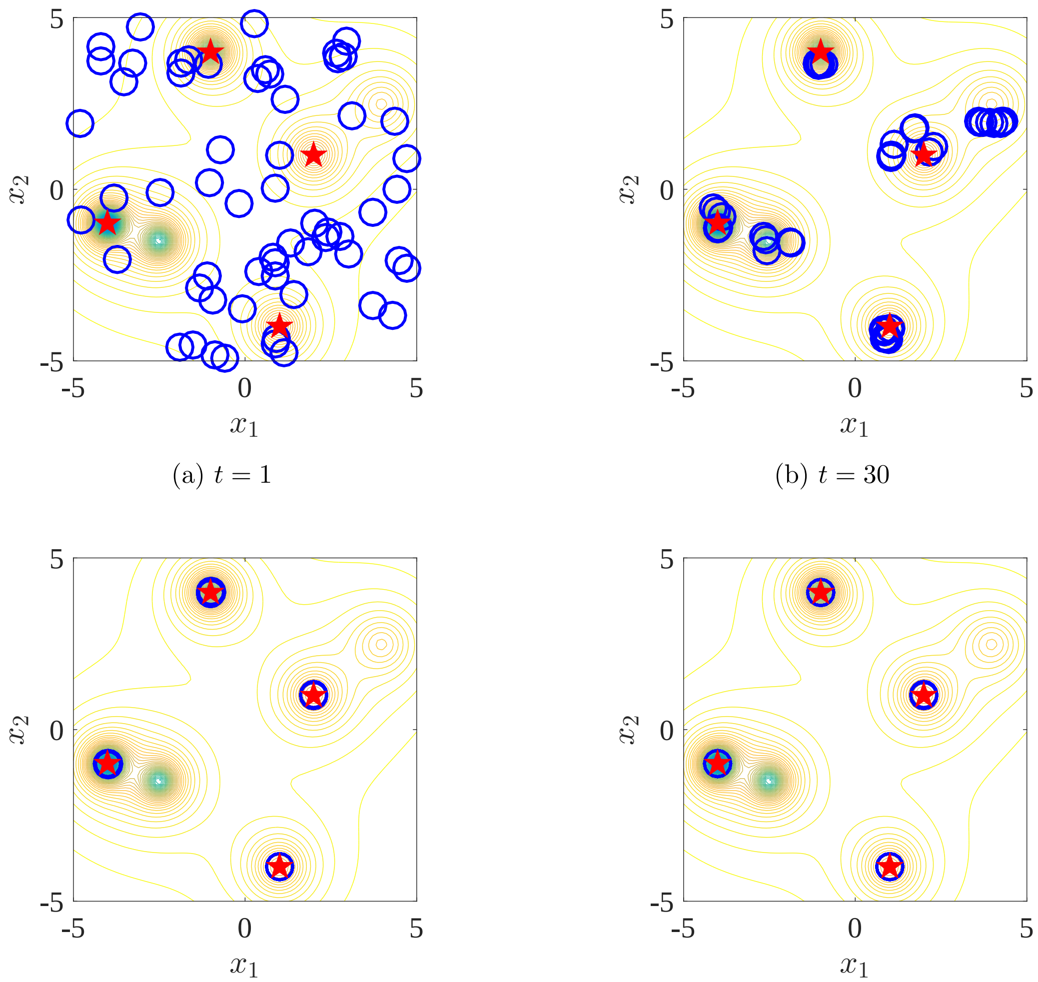
(a) $t = 1$

(b) $t = 30$

(c) $t = 50$

(d) $t = 100$

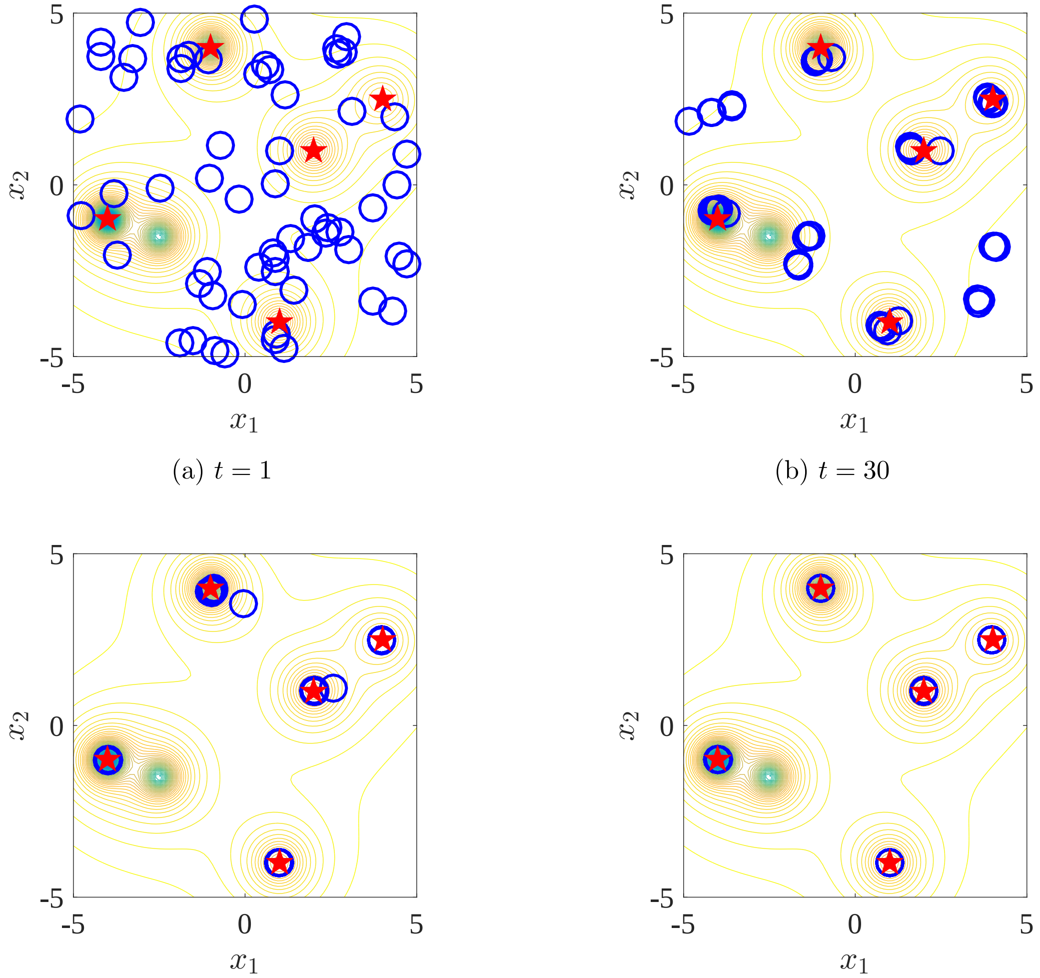
Fig. 5.4: Transition of Search Points for Search State of Cond. 2 (δ, ε= $7.5, 2$) in Superior Relation Based Firefly Algorithm ($N = 2$)



(a) $t = 1$ (b) $t = 30$

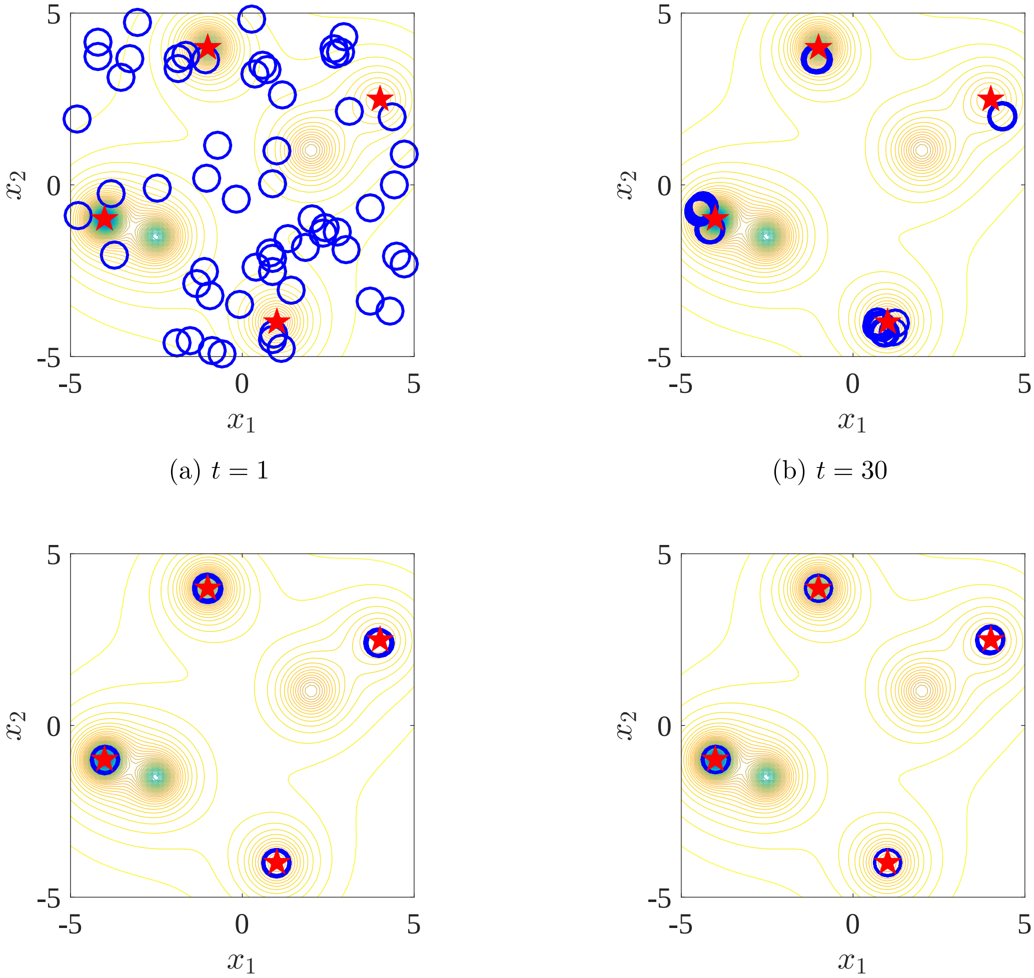
(c) $t = 50$ (d) $t = 100$

Fig. 5.5: Transition of Search Points for Search State of Cond. 3 (δ, ε= $8.5, 2$) in Superior Relation Based Firefly Algorithm ($N = 2$)



(a) $t = 1$ (b) $t = 30$ (c) $t = 50$ (d) $t = 100$

Fig. 5.6: Transition of Search Points for Search State of Cond. 4 (δ, ε= $10, 1$) in Superior Relation Based Firefly Algorithm ($N = 2$)



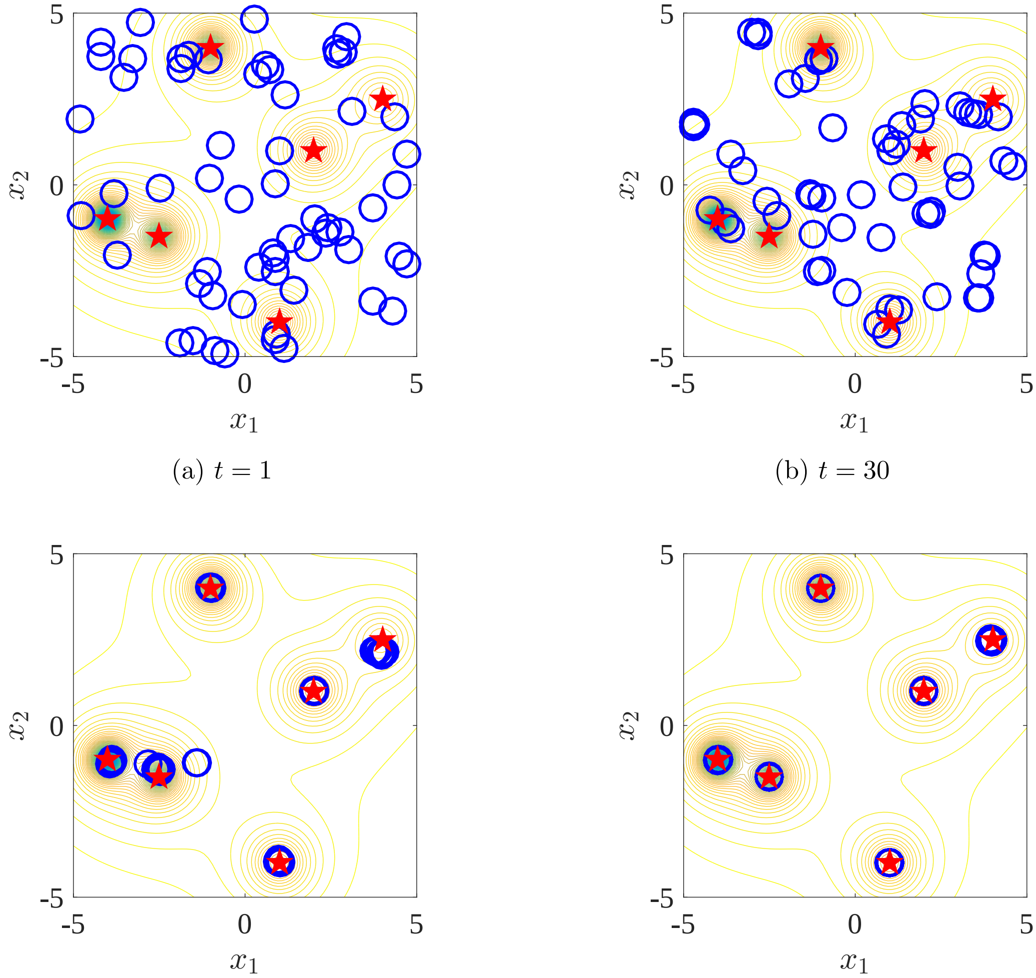
(a) $t = 1$

(b) $t = 30$

(c) $t = 50$

(d) $t = 100$

Fig. 5.7: Transition of Search Points for Search State of Cond. 5 (δ, ε= $10, 2$) in Superior Relation Based Firefly Algorithm ($N = 2$)



(a) $t = 1$ (b) $t = 30$

(c) $t = 50$ (d) $t = 100$

Fig. 5.8: Transition of Search Points for Search State of Cond. 6 (δ, ε= $10, 5.5$) in Superior Relation Based Firefly Algorithm ($N = 2$)

5.5 Superior Relation Based Genetic Algorithm for Superior Solution Set Search Problem

5.5.1 Proposal of Superior Relation Based Genetic Algorithm

In a multi-objective optimization problem, there is no single optimal solution that is optimal for all objectives. Instead, there is a Pareto optimal solution set that forms the Pareto front, which is the optimal trade-off between objectives. The superior solution set is defined as the difference between the evaluation values and the global optimal solution falls within a certain range and the distance from other local optimal solutions that is greater than a certain distance. So the structural similarities between the superior solution set search problem and multi-objective optimization problem.

Evolutionary computation is attracting attention as a means to obtain a Pareto optimal solution set in the multi-objective optimization problem [79, 80]. Evolutionary computation is a general term for probabilistic multipoint search methods based on solution populations, such as Differential Evolution, Particle Swarm Optimization, and Artificial Bee Colony Algorithm, starting with Genetic Algorithm (see **Algorithm C.4**) and Evolution Strategy. Beginning with the proposal of VEGA (Vector Evaluated Genetic Algorithm) [81] by Schaffer in 1985, NSGA-II (Fast Elitist Non-dominated Sorting Genetic Algorithm) [76] by Deb et al. Has been applied as the most famous algorithm. It is frequently used in. Thus, evolutionary multi-objective optimization is maturing for some of the multi-objective optimization problems. So we can also propose a new superior solution set search method based on GA. Specifically, first, the superior solution fitness is defined as an index that gives the superior relation of the solution using δ and ε in Section 5.5. Next, we propose a GA with the superior solution fitness.

Superior relation based GA for the superior solution set search problem of the objective function $f(\boldsymbol{x})$ $(\boldsymbol{x} \in \mathbb{R}^N)$ is shown by **Algrithm 5.2**.

Algorithm 5.2 Superior Relation Based Genetic Algorithm (GA-4S)

procedure GA-4S$(m, p_c, p_m, \delta, \varepsilon, T_{max})$
Step 1: Initialization
  Give initial solutions $\boldsymbol{P}^1$ $(|\boldsymbol{P}^1| = m)$, Set $t = 1$
Step 2: Generation of New Solutions
  $\boldsymbol{Q}^t = \emptyset$
  for $i = 1 \dots m$ **do**
    Choose randomly $\boldsymbol{x}_a, \boldsymbol{x}_b \in \boldsymbol{P}^t$ $(a \neq b)$
    $\{\boldsymbol{y}_a, \boldsymbol{y}_b\} =$ SIMULATED BINARY CROSSOVER$(\boldsymbol{x}_a, \boldsymbol{x}_b)$ ▷ See Algorithm C.1
    Choose randomly $\boldsymbol{y} \in \{\boldsymbol{y}_a, \boldsymbol{y}_b\}$
    $\boldsymbol{q} =$ POLYNOMIAL MUTATION$(\boldsymbol{y})$ ▷ See Algorithm C.3
    $\boldsymbol{Q}^t := \boldsymbol{Q}^t \cup \{\boldsymbol{q}\}$
  end for
Step 3: Superior Solution Fitness Assignment
  $\boldsymbol{U}^t = \boldsymbol{P}^t \cup \boldsymbol{Q}^t$ $(|\boldsymbol{U}^t| = 2m)$
  for each $\boldsymbol{x} \in \boldsymbol{U}^t$ **do**
    for each $\boldsymbol{z} \in \boldsymbol{U}^t$ **do**
      $fit(\boldsymbol{x} \in \boldsymbol{P}^t, \delta, \varepsilon) = \text{Card}\{\boldsymbol{z} \mid \boldsymbol{z} \prec_\delta \boldsymbol{x} \vee \boldsymbol{z} \prec_\varepsilon \boldsymbol{x}\}$
    end for
  end for
Step 4: Superior Solution Fitness Based Selection
  Sort $\boldsymbol{U}^t$ in ascending order using $fit(\boldsymbol{x} \in \boldsymbol{U}^t, \delta, \varepsilon)$
  $\boldsymbol{P}^{t+1} := \boldsymbol{U}^t[1 : m]$
Step 5: Termination
  if $t < T_{\max}$ **then**
    $t := t + 1$
    Go to **Step 2**
  else
end procedure

5.5.2 Numerical Experiment

(a) Numerical Experiment Conditions

Through numerical experiments, we solved the superior solution set search problem which consist of aforementioned target function with a $fit(\boldsymbol{x} \in \boldsymbol{P}, \delta, \varepsilon)$-based elite selection GA called "GA-4S". We evaluated the usefulness of GA-4S compared with a basic GA. Please refer to Shekel's function in **Appendix A** for details of the benchmark function used in this experiments.

In each experiment, we used Simulated Binary Crossover (SBX) [76, 86] (see **Algorithm C.1** and **Algorithm C.2**) and Polynomial Mutation (PM) [76, 86] (see **Algorithm C.3**). The parameters for these operations were crossover rate $p_c = 1$ and mutation rate $p_m = 0.5$. The values of the distribution adjustment variables for these operations were set to 25 different values of η_c and $\eta_m = [2, 5, 10, 20, 50]$. The number of search points m was 30 and the initial solution was randomized within a feasible region($[-5, 5]^N$). The maximum number of generations $T_{\max}$ was 100 and the dimension N was 2. We ran 50 trials with different initial solutions for each trials under the above conditions.

In this paper, we set the usable situation of benchmark function to verify the temporary setup with the proposed method, which is tentatively set up to correspond to the user's desired level. There are six types of (δ, ε) and superior solution set $\mathcal{Q}(\boldsymbol{X}; \delta, \varepsilon)$, where ($\delta$, ε)=(7.5, 1), (7.5, 2), (8.5, 2), (10, 1), (10, 2), and (10, 5.5). We regard them as Cond. 1 $\sim$ Cond. 6. The criteria for setting parameters (δ, ε) are explained as follow.

- In the case of relatively allowing solution proximity and not allowing deterioration of evaluation value (Cond. 1)
- In the case of requiring relatively large solution diversity and allowing deterioration of the evaluation value (Cond. 4)
- In the case of requiring moderate diversity and evaluation values in the above two cases (Cond. 2, 3, 5, and 6)

(b) Evaluation Index

In this numerical experiments, we evaluate the search performance of the superior solution using two evaluation indices Peak Ratio (PR) and Convergence Ratio (CR), which are defined in Section 5.4.2 by the Eqs.(5.9) and (5.10). Here, we set the threshold parameter $\eta = 10^{-1}$ in all conditions to obtain a superior solution in the Eq.(5.10).

The mean (Mean) and the standard deviation (S.D.) of each index are compared after 50 runtimes using different initial solutions. The indices are described as follows:

- PR_{Mean} and $PR_{\mathrm{S.D.}}$: The mean and standard deviation of PR.
- CR_{Mean} and $CR_{\mathrm{S.D.}}$: The mean and standard deviation of CR.

(c) Results of the Numerical Experiment

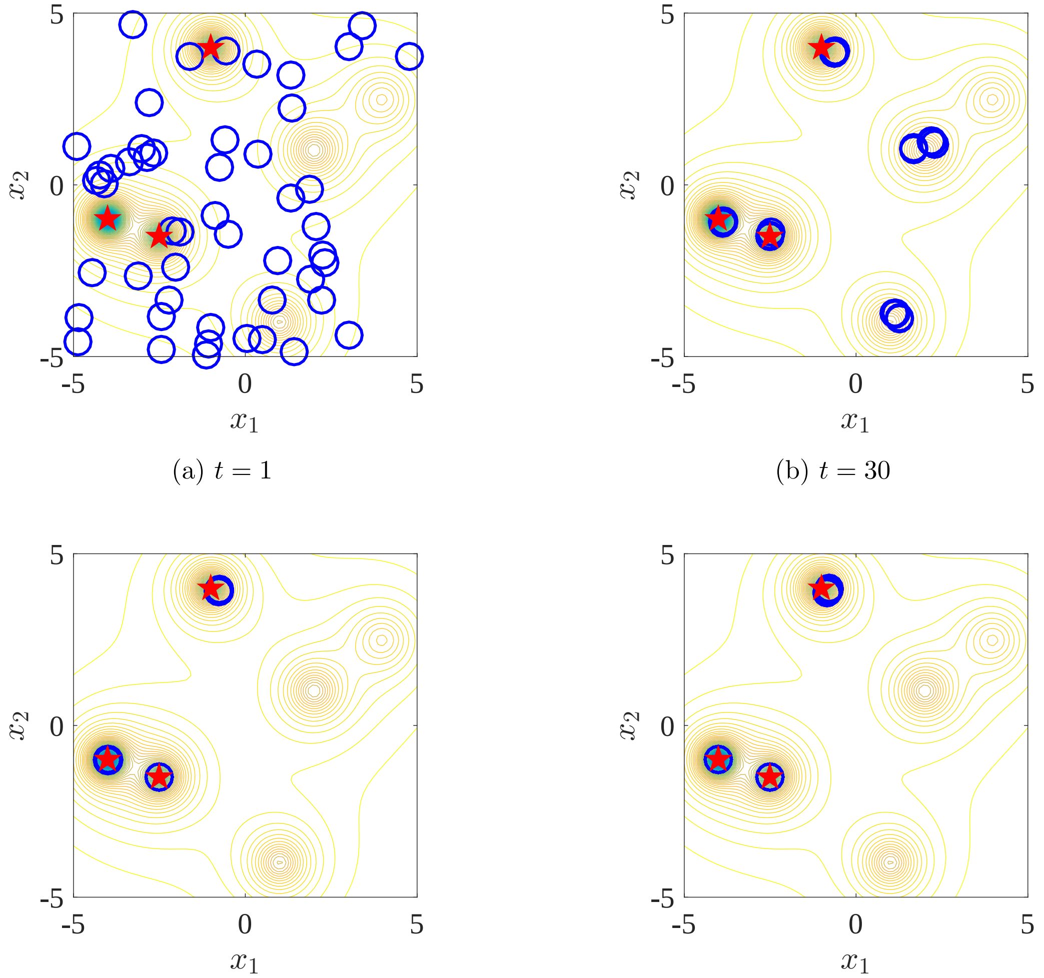
Table 5.3 shows the results of numerical experiments. The best of PR_{Mean} and CR_{Mean} are shown in bold. In Figs. 5.9∼5.14 show the search transition in the proposed method in six types of (δ, ε) in two-dimensional problem. In Figs. 5.9∼5.14, the ★ in the figure is the superior solution, while ◯ indicates the search point. In all conditions, the GA-4S shows higher PR values than the basic GA. That is, the GA-4S has found more superior solutions than the basic GA. Moreover, under many initial conditions, the proposed method shows smaller CR values than the basic GA. That is, in the basic GA, many search points converge to the superior solution when the search point convergence of is not sufficient or is not the superior solution, whereas the GA-4S is one in which many search points can be superior solutions. In this way, since the GA-4S shows excellent results both in terms of PR and CR values for the superior solution, the GA-4S acquires more superior solutions, and it can be said that the search point does not converge to any local optimal solution other than the superior solution. This can also be inferred from the transition of the search point. In the GA-4S, the converged solution varies with changes in the parameters δ and ε, and many search points converge to the superior solution to be acquired.

(d) Discussion

From the numerical experiment results, it can be seen that the basic GA has lower search performance than the proposed method. This is considered to be caused by the fact that the basic GA is a group of search points gathered in one promising area.

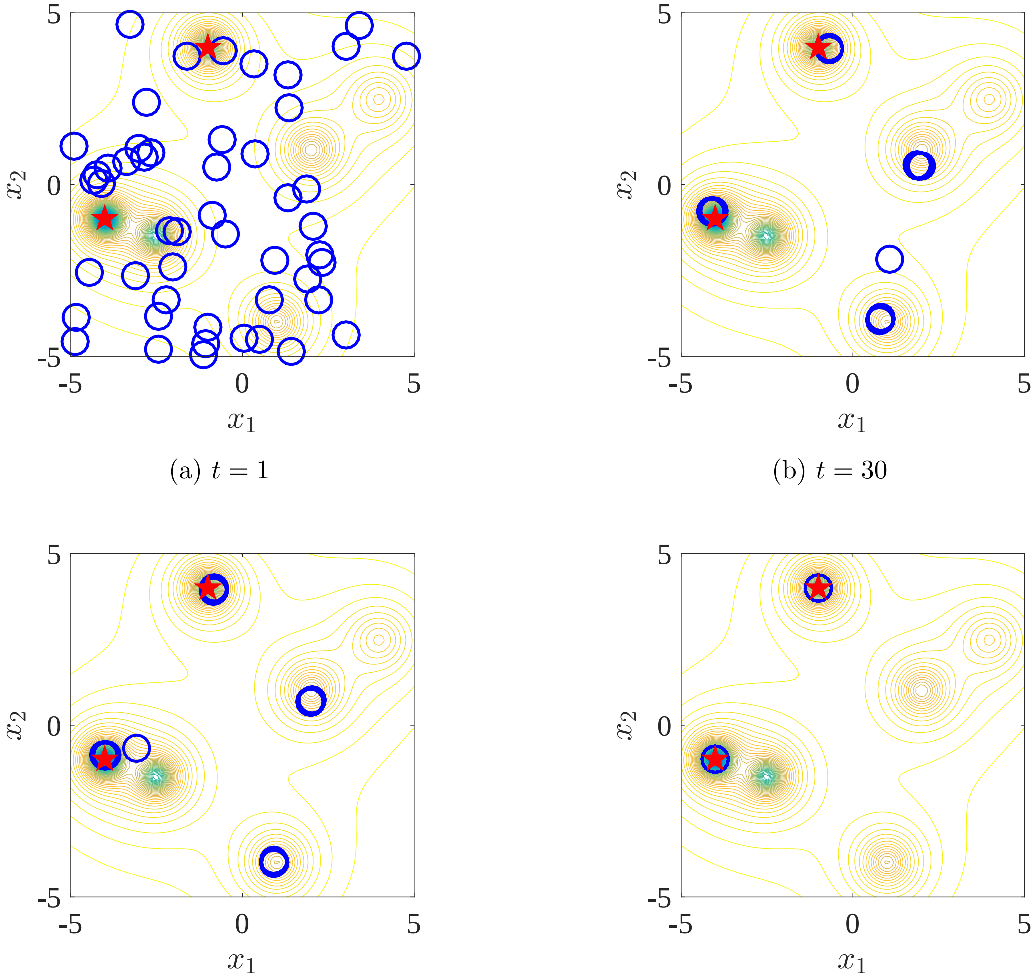
Table 5.3: Experiment Results

Dimension	Problem				GA-4S				Genetic Algorithm			
	Condition	δ	ε	Superior Solution	PR_{Mean}	$PR_{\text{S.D.}}$	CR_{Mean}	$CR_{\text{S.D.}}$	PR_{Mean}	$PR_{\text{S.D.}}$	CR_{Mean}	$CR_{\text{S.D.}}$
$N = 2$	Cond. 1	7.5	1	A, B, C	**2.98**	0.328	**0.0056**	0.0021	1.02	0.182	0.1062	0.0738
	Cond. 2	7.5	2	A, C	**1.96**	0.198	**0.0087**	0.0015	0.98	0.141	0.1180	0.0680
	Cond. 3	8.5	2	A, C, D, E	**3.92**	0.285	**0.0103**	0.0031	1	0	0.0892	0.0765
	Cond. 4	10	1	A, B, C, D, E, F	**5.88**	0.238	**0.0120**	0.0032	1.04	0.198	0.0848	0.0853
	Cond. 5	10	2	A, C, D, E, F	**4.96**	0.198	**0.0145**	0.0052	1	0	0.0643	0.0785
	Cond. 6	10	5.5	A, C, D, F	**3.94**	0.274	**0.0167**	0.0038	0.96	0.153	0.0865	0.0685



(a) $t = 1$ (b) $t = 30$ (c) $t = 50$ (d) $t = 100$

Fig. 5.9: Transition of Search Points for Search State of Cond. 1 (δ, $\varepsilon = 7.5, 1$) in Superior Relation Based Genetic Algorithm ($N = 2$)



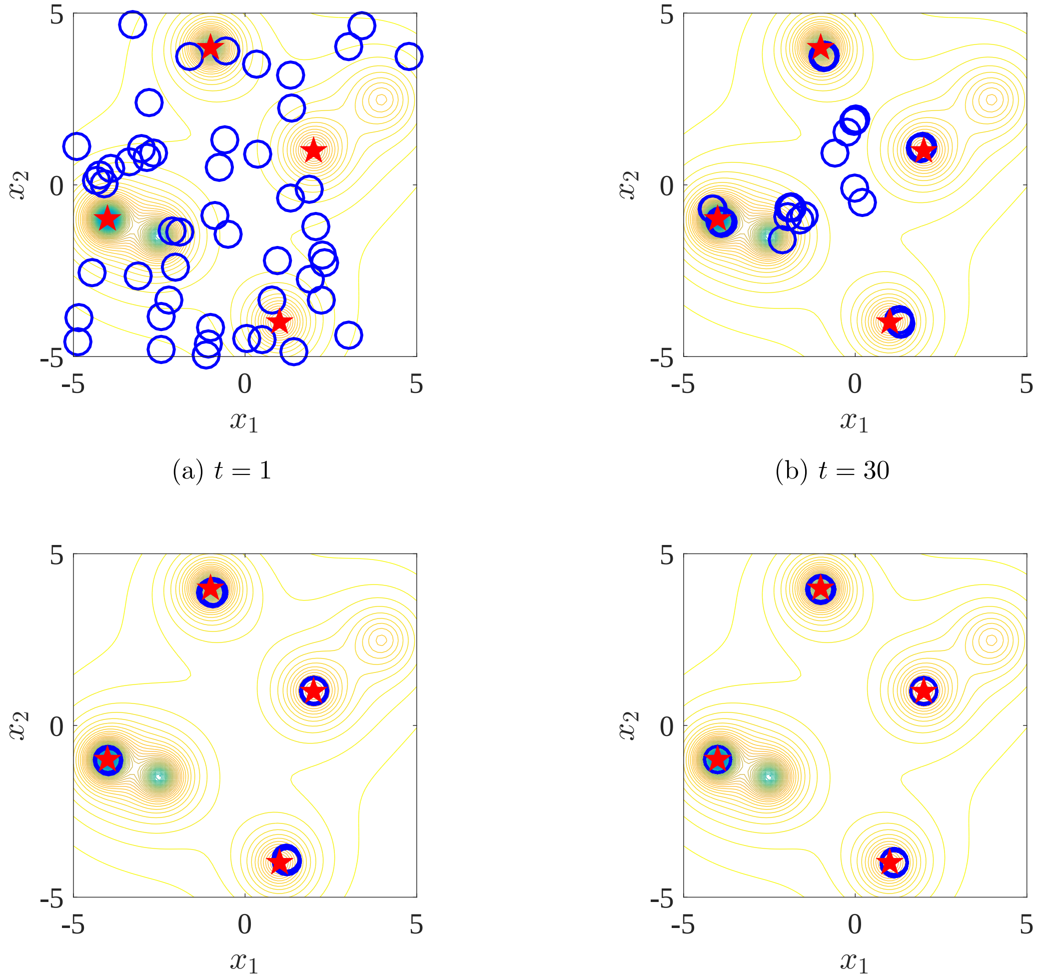
(a) $t = 1$

(b) $t = 30$

(c) $t = 50$

(d) $t = 100$

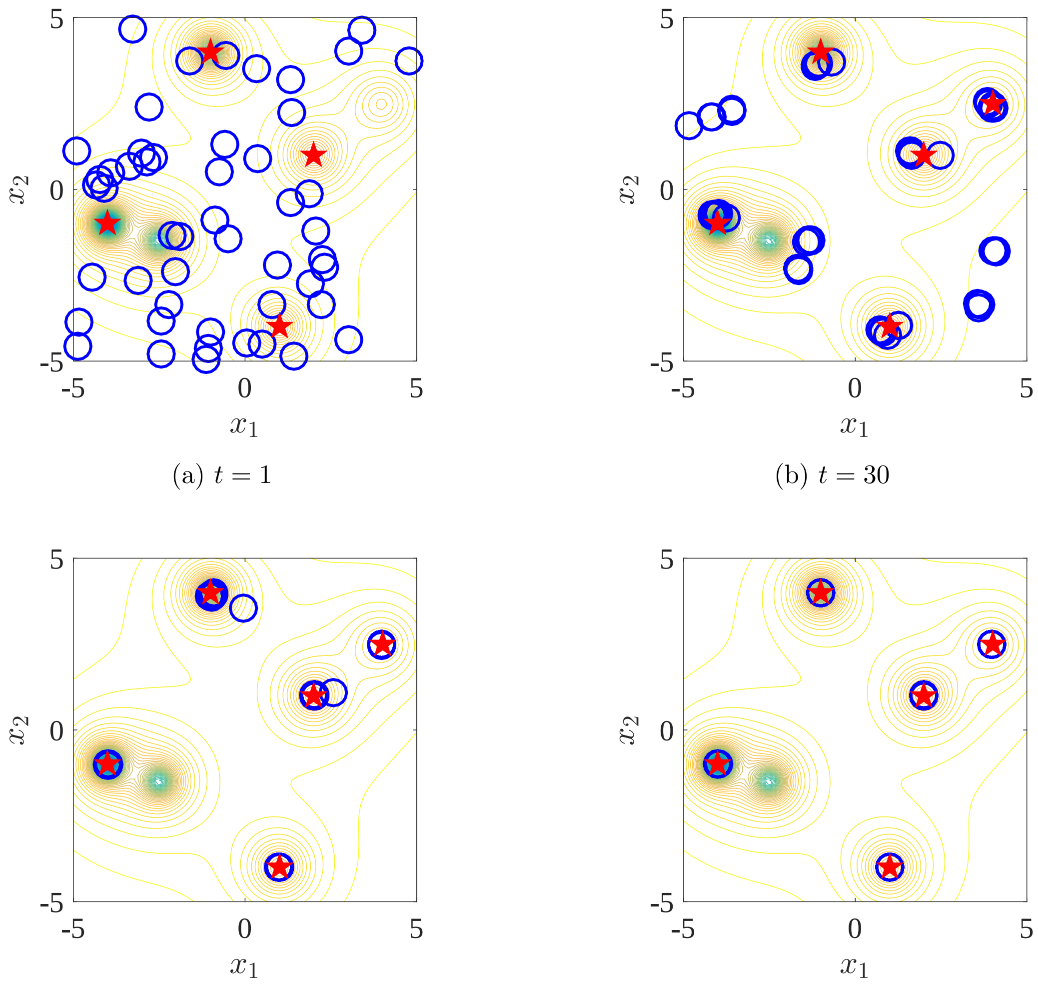
Fig. 5.10: Transition of Search Points for Search State of Cond. 2 (δ, ε= $7.5, 2$) in Superior Relation Based Genetic Algorithm ($N = 2$)



(a) $t = 1$ (b) $t = 30$

(c) $t = 50$ (d) $t = 100$

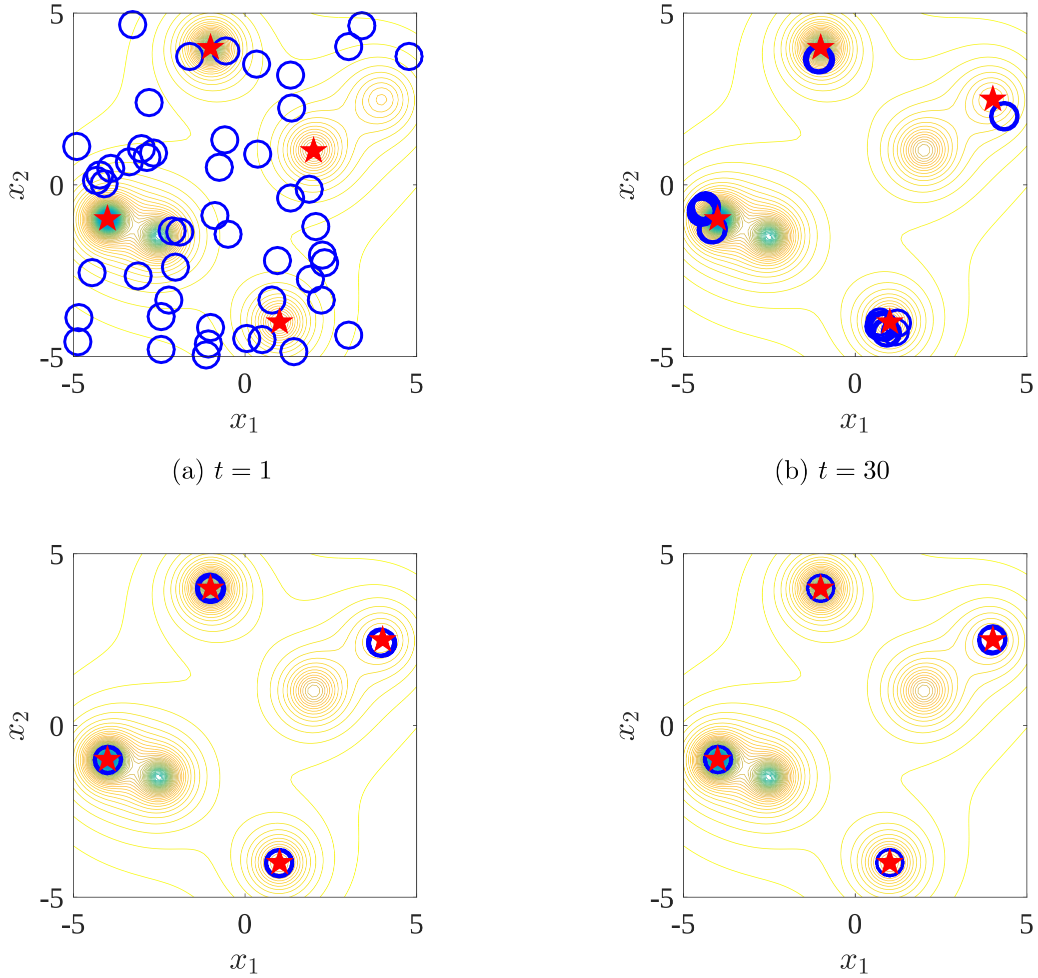
Fig. 5.11: Transition of Search Points for Search State of Cond. 3 (δ, ε= $8.5, 2$) in Superior Relation Based Genetic Algorithm ($N = 2$)



(a) $t = 1$ (b) $t = 30$

(c) $t = 50$ (d) $t = 100$

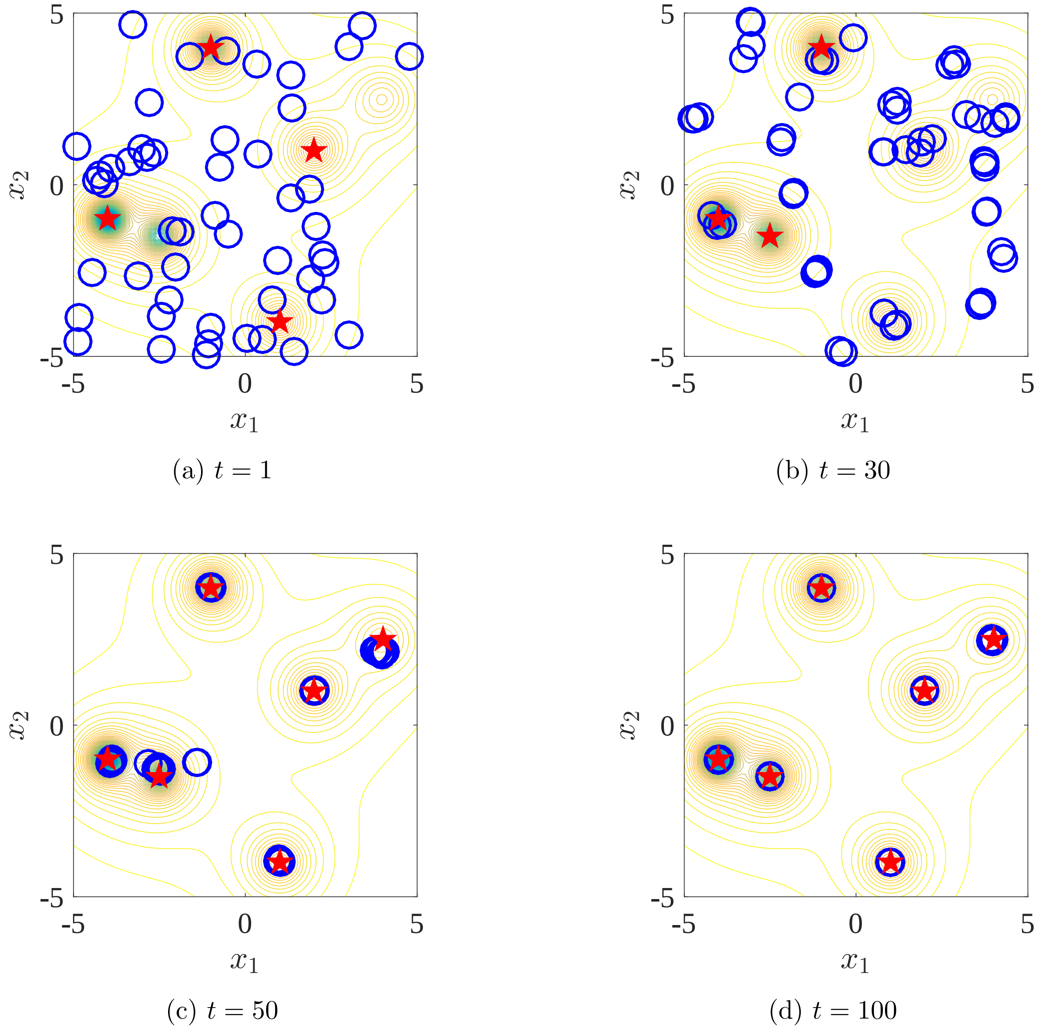
Fig. 5.12: Transition of Search Points for Search State of Cond. 4 (δ, ε= $10, 1$) in Superior Relation Based Genetic Algorithm ($N = 2$)



(a) $t = 1$ (b) $t = 30$

(c) $t = 50$ (d) $t = 100$

Fig. 5.13: Transition of Search Points for Search State of Cond. 5 (δ, ε= $10, 2$) in Superior Relation Based Genetic Algorithm ($N = 2$)



(a) $t = 1$ (b) $t = 30$ (c) $t = 50$ (d) $t = 100$

Fig. 5.14: Transition of Search Points for Search State of Cond. 6 (δ, ε= $10, 5.5$) in Superior Relation Based Genetic Algorithm ($N = 2$)

5.6 Summary

In this chapter, we pointed out the structural similarities between the superior solution set search problem and the multi-objective optimization problem. We proposed a search strategy based on the superior relation of the superior solution set by utilizing the analyzed properties and the user's desire level for the search. The proposed superior solution set search method explicitly includes the definition of the superior solution set based on the superior relation. Numerical experiments verified the usefulness of FA based on the proposed superior relation.

6 Conclusion

This Chapter summarizes the content of the paper and describes future issues and research prospects.

6.1 Summary

The results of this research are shown below.

- The superior solution set was defined based on the single-objective optimization problem by a mathematical formula as a set of local optimal solutions whose evaluation values are superior by a certain value and the distance between solutions is more than a certain distance, and the superior solution set search problem was defined based on this set. The superior solution set search problem can meet difficult requirements, which cannot be considered in usuals optimization, were useful.
- We compared typical metaheuristics such as Particle Swarm Optimization, Differential Evolution, and Artificial Bee Colony Algorithm. Then, we pointed out Firefly Algorithm (FA) has a mechanism for adjusting the amount of movement based on the distance from the reference point. From this feature, it was clarified that FA can obtain multiple local optimum solutions. FA was a basic method

for searching for the superior solution set, which is a subset of the local optimum solution set.

- It is effective to utilize cluster information for the superior solution set search problem, we proposed an FA-based superior solution set search method (FA-CI) that utilizes the search points with the best objective function values in each cluster. The proposed FA-CI has verified the excellent search performance by numerical experiments.
- Discuss the differences in the problem structure between the superior solution set search problem and the single-objective optimization problem from the viewpoint of the search strategy (diversification and intensification) obtained through structural analysis for the conventional single-objective optimization problem. Based on that discussion, we constructed a search strategy (diversification and intensification) for the superior solution set search problem. From the viewpoint of diversification and intensification for the superior solution set search problem, the relationship between FA parameters and diversification and intensification was clarified. After the above analysis, we proposed an Adaptive FA that preliminarily performs parameter adjustment according to set target value schedule, evaluated indicators of diversification and intensification for the superior solution set search problem, and verified their usefulness by conducting a numerical experiment using benchmark functions.
- We analyzed the properties of the superior solution set search problem, and we pointed out the structural similarities between the superior solution set search problem and multi-objective optimization problem. Based on the analyzed properties, we proposed a new FA based on superior relations, which can search for the superior solution set by using the user's quantitative desire level as a search strategy. By introducing the distance of the problem space into the moving mechanism, FA makes it possible to search for multiple local optimal solutions by dividing the solution set into multiple groups. By analyzing this property possessed of FA, we clarified the affinity between FA and the superior solution set search problem. We subsequently analyzed the superior solution set search

problem and FA, and we discussed its properties together with a similar problem setting. We proposed an FA based on superior relations as a new optimization technique for the superior solution set problem based on these analyses. Numerical experiments are then conducted using the superior solution set search problem, and the usefulness of the proposed method was demonstrated while comparing the performance of the proposed method with the conventional FA.

6.2 Future Issues

In addition, the following items can be mentioned as future major issues in this field. Outline of future issues is shown in Fig.6.1.

(a) Issues 1

The research approach so far has developed the superior solution set search method based on the Firefly Algorithm (FA). From the viewpoint of the FA search mechanism and search performance, we point out the lack of FA for high-dimensional superior solution set problems. The neighborhood generation of FA is expressed by the Eq.(3.25). From the Eq.(3.25), the neighborhood generation is combining the difference vector between the search point $\boldsymbol{x}_i^t$ and the reference point $\boldsymbol{z}_j$, which is better than $\boldsymbol{x}_i^t$, and the perturbed random number $\alpha\boldsymbol{R}$.

Here, FA uses the Euclidean distance (the L_2 norm) to generate the neighborhood. The L_k norm distance function, which is defined as the Eq.(6.1), is also susceptible to the dimensionality curse for many classes of data distributions. Reference [82] discusses the general behavior of the commonly used L_k norm in high dimensional space.

$$L_k(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{N}(\left\|\boldsymbol{x}^i - \boldsymbol{y}^i\right\|^k)^{1/k},\ \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^N,\ k \in \mathbb{Z} \tag{6.1}$$

Table 6.1: Notations and Basic Defintions

Notation	Defintion
N	Dimensionality of the data space
m	Number of data points
$\mathcal{F}$	1-dimensional data distribution in $[0, 1]$
$\boldsymbol{R}_N$	Data point from $\mathcal{F}^d$ with each coordinate drawn from $\mathcal{F}$
$dist_N^k(\boldsymbol{x}, \boldsymbol{y})$	Distance between $(x^1, \cdots, x^N)$ and $(y^1, \cdots, y^N)$
$\lVert\cdot\rVert_k$	Distance of a vector to the origin $(0, \cdots, 0)$ using the function $dist_N^k(\cdot, \cdot)$
$Dmax_N^k$	Farthest distance of the m points to the origin using the distance metric L_k
$Dmin_N^k$	Nearest distance of the m points to the origin using the distance metric L_k
$E[\boldsymbol{R}], var[\boldsymbol{R}]$	Expected value and variance of a random variable $\boldsymbol{R}$
$\boldsymbol{Y} \to_p c$	A vector sequence $(Y_1, \cdots, Y_N)$ converges in probability to
	a constant vector c if : $\forall \epsilon > 0 \; \underset{d\to\infty}{limit} \; P[dist_N(Y_d, c) \leq \epsilon] = 1$

Theorem 6.1 **(Adapted for L_k metric)**

$$If \lim_{d\to\infty} var\left(\frac{\lVert\boldsymbol{R}_N\rVert_k}{E\left[\lVert\boldsymbol{R}_N\rVert_k\right]}\right) = 0, \; then \; \frac{Dmax_d^k - Dmin_N^k}{Dmin_N^k} \to_p 0$$

□

See Reference [83] for proof of a more general version of this result. Table 6.1 shows notations and basic defintions of Theorem 6.1.

According to the result of Theorem 6.1, the distance L_k is the distance between m search points, the distance $Dmax_N^k$ between the two search points is the farthest in m search points, and the distance $Dmin_N^k$ between the two search points is the nearest in m search points. The higher the number of dimensions N, the closer the distance $Dmax_N^k$ and the distance $Dmin_N^k$ is. In other words, when the number of dimensions N becomes infinite, the distance L_k between all search points tends to be the same, and the distance L_k between search points is meaningless and unstable.

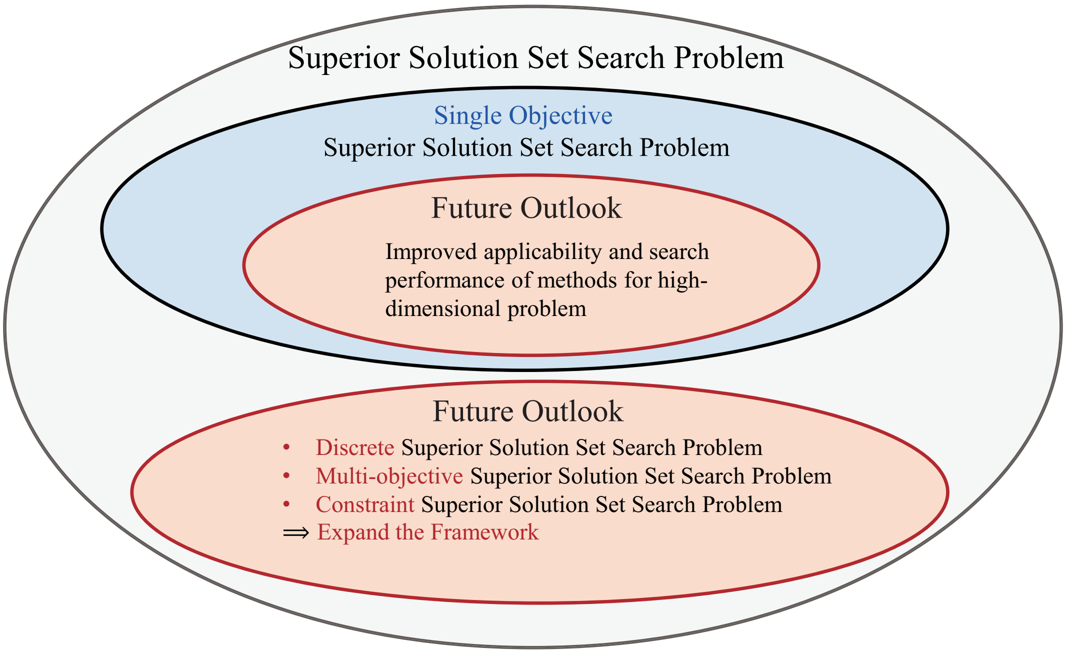


Fig. 6.1: Outline of Future Issues

In this paper, it is difficult to adjust the movement amount of the difference vector in the high-dimensional search space by the parameter γ because the Euclidean distance (the norm L_2) is the same. In other words, the distance between each search point $\boldsymbol{x}_i^t$ and each reference point $\boldsymbol{z}_j$ in the high-dimensional search space is almost the same, so the adjustment of the movement amount becomes almost the same according to the distance. By this effect, the norm L_2 of the difference vector is almost the same as the effect of reducing the coefficient $\beta_{i,j}$ $(= \beta_0 \mathrm{e}^{-\gamma||\boldsymbol{z}_j - \boldsymbol{x}_i^t||_2^2})$. It is conceivable to lose the property for basic superior solution set search that the search point group is divided into multiple parts.

On the other hand, since FA moves absolutely, the search point $\boldsymbol{x}_i^t$ is sure to update to the neighborhood solution $\hat{\boldsymbol{x}}_i^t$ $(= \boldsymbol{x}_i^t + \beta_{i,j}(\boldsymbol{z}_j - \boldsymbol{x}_i^t) + \alpha \boldsymbol{R})$. This does not always move the search point to the position where it improves, and allows deterioration. Further, since the L_2 norm of the difference vector in high-dimensional search space is extremely large, even if somewhat large difference vector, since the coefficient $\beta_{i,j}$ is exponentially small, the difference vector is approximately **0**, and the substantially depends only on the random vector. Due to this effect, it is also considered bad to search for the superior solution set.

So we need propose a new superior solution set search method that enables more efficient search and examination of superior solution set search problem in high dimension to deal with the issues 1.

(b) Issues 2

We proposed the most basic superior solution set search method that uses the same selection operation as the multi-objective optimization method based on the superior relation. Through basic numerical experiments, it was shown that the proposed method has the property that the local solution that converges changes according to the change of the parameters that determine the superior solution set. However, the experiment only using one benchmark problem were performed. The evaluation indexes for the search performance of the superior solution set search method and proposals for benchmark problems are also required to evaluate the superior solution set search method.

(c) Issues 3

We think that there is room for consideration in formulating the superior solution set search problem. The superior solution set search problem and the superior solution set search method proposed in this study are closely linked, and by conducting research focusing on both the problem and the method, the development of an optimization method that meets the needs of practical applications can be expected.

The approach of superior relation used this time can be applied to all of the multi-objective optimization method based on the superior relation proposed so far. In this study, we have completed the examination of one of the most basic methods, but the application of the multi-objective optimization method based on other superior relation to the superior solution set search problem is a future task.

(d) Issues 4

Based on the multi-objective optimization problem, discrete optimization problem, or constraint optimization problem, a multi-objective superior solution set, a discrete superior solution set, or a constraint superior solution set is defined by a mathematical formula as a set of Pareto solutions whose evaluation values are superior to the objective function space by a certain amount or more and the solutions are separated from each other by a certain value or more in the determinant variable space. Based on the defined problem, the development of method is also an important issue to extend the framework of superior solution set search problems / methods.

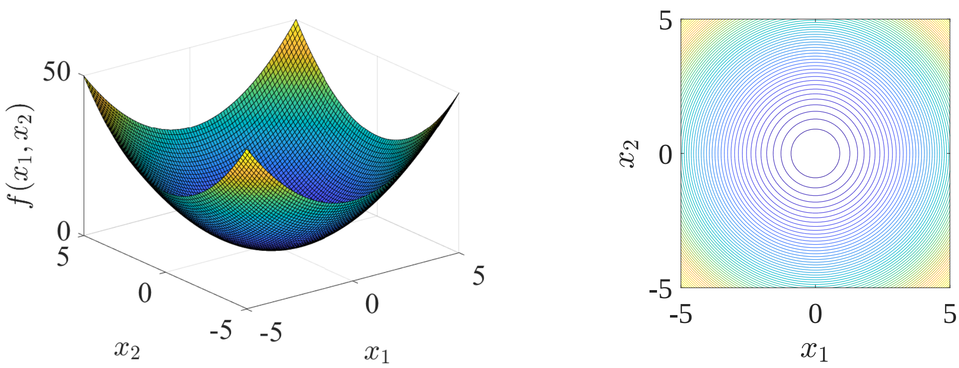
A Benchmark Function

The benchmark function used in this paper is described.

1. Sphere Function

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} x_i^2$$

$$\boldsymbol{x}^*[0, 0, \cdots, 0]^N \qquad f(\boldsymbol{x}^*) = 0$$



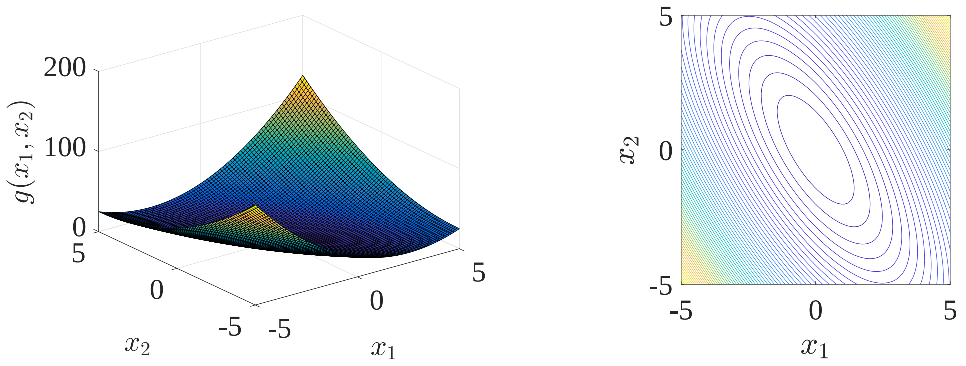
(a) Outline (b) Contour

Fig. A.1: Sphere Function

2. Schwefel Functuon

$$g(\boldsymbol{x}) = \sum_{i=1}^{N} \Big(\sum_{j=1}^{i} x_j \Big)^2$$

$$\boldsymbol{x}^* = [0, 0, \cdots, 0]^N \qquad g(\boldsymbol{x}^*) = 0$$



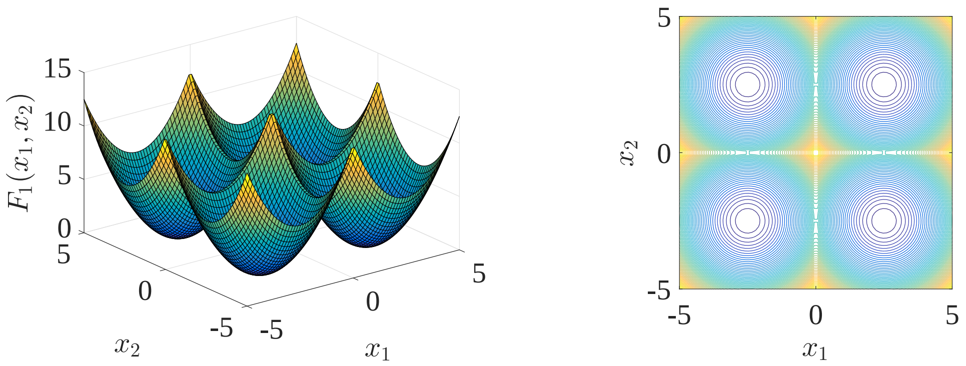
(a) Outline (b) Contour

Fig. A.2: Schwefel Functuon

3. Functuon 1

$$F_1(\boldsymbol{x}) = \min \Big(f(\boldsymbol{x}+\boldsymbol{Y}),\ f(\boldsymbol{x}-\boldsymbol{Y}),\ f(\boldsymbol{x}+\boldsymbol{Z}),\ f(\boldsymbol{x}-\boldsymbol{Z}) \Big)$$

$$\boldsymbol{Y} = [2.5, 2.5, \cdots, 2.5, 2.5]^N \qquad \boldsymbol{Z} = [2.5, -2.5, \cdots, 2.5, -2.5]^N$$



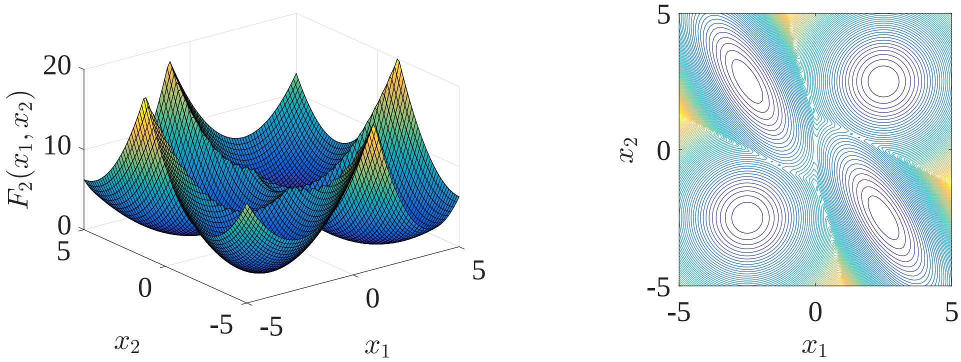
(a) Outline (b) Contour

Fig. A.3: Functuon 1

4. Functuon 2

$$F_2(\boldsymbol{x}) = \min\Big(f(\boldsymbol{x}+\boldsymbol{Y}),\ g(\boldsymbol{x}+\boldsymbol{Z}),\ f(\boldsymbol{x}-\boldsymbol{Y}),\ g(\boldsymbol{x}-\boldsymbol{Z})\Big)$$

$$\boldsymbol{Y} = [2.5, 2.5, \cdots, 2.5, 2.5]^N \qquad \boldsymbol{Z} = [2.5, -2.5, \cdots, 2.5, -2.5]^N$$



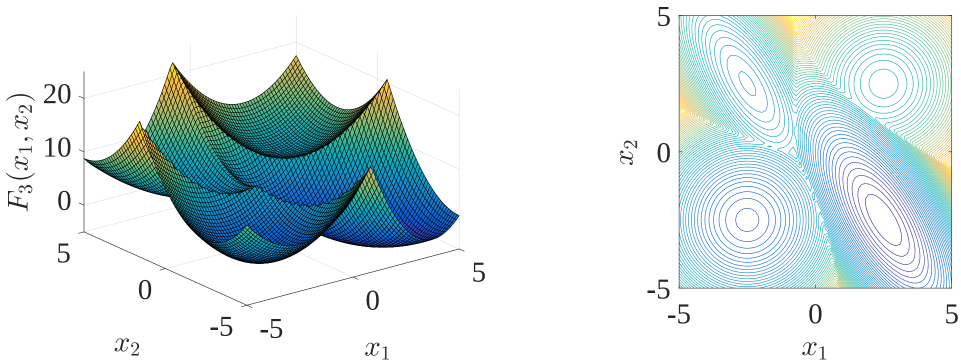
(a) Outline (b) Contour

Fig. A.4: Functuon 2

5. Functuon 3

$$F_3(\boldsymbol{x}) = \min\Big(f(\boldsymbol{x}+\boldsymbol{Y})+\mathrm{E},\ g(\boldsymbol{x}+\boldsymbol{Z})-\mathrm{E},\ f(\boldsymbol{x}-\boldsymbol{Y})+\mathrm{F},\ g(\boldsymbol{x}-\boldsymbol{Z})-\mathrm{F}\Big)$$

$$\boldsymbol{Y} = [2.5, 2.5, \cdots, 2.5, 2.5]^N \qquad \boldsymbol{Z} = [2.5, -2.5, \cdots, 2.5, -2.5]^N \qquad \mathrm{E} = 2.5 \qquad \mathrm{F} = 5$$



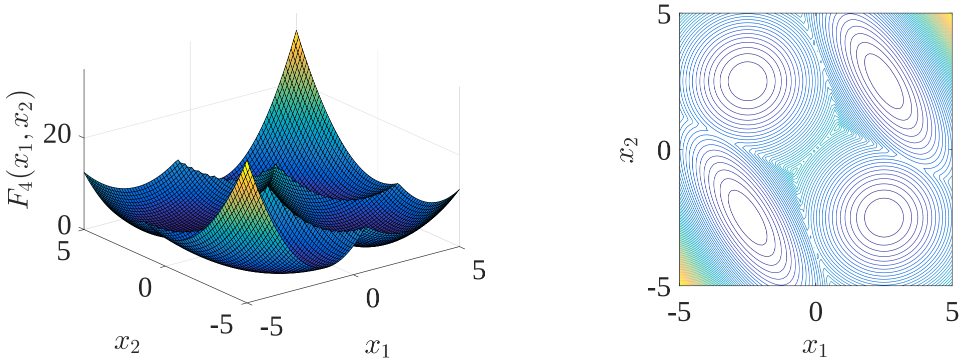
(a) Outline (b) Contour

Fig. A.5: Functuon 3

6. Functuon 4

$$F_4(\boldsymbol{x}) = \min\Big(g(\boldsymbol{x}+\boldsymbol{Y}),\ f(\boldsymbol{x}+\boldsymbol{Z}),\ g(\boldsymbol{x}-\boldsymbol{Y}),\ f(\boldsymbol{x}-\boldsymbol{Z})\Big)$$

$$\boldsymbol{Y} = [2.5, 2.5, \cdots, 2.5, 2.5]^N \qquad \boldsymbol{Z} = [2.5, -2.5, \cdots, 2.5, -2.5]^N$$



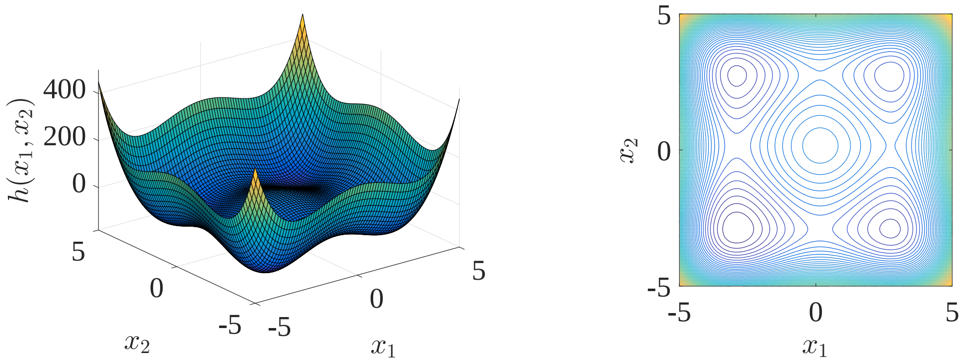
(a) Outline (b) Contour

Fig. A.6: Functuon 4

7. 2^N minima Function

$$h(\boldsymbol{x}) = \sum_{i=1}^{N} \{x_i^4 - 16x_i^2 + 5x_i\}$$

$$\boldsymbol{x}^* \approx [-2.90, -2.90, \cdots, -2.90]^N \qquad h(\boldsymbol{x}^*) \approx -78n$$

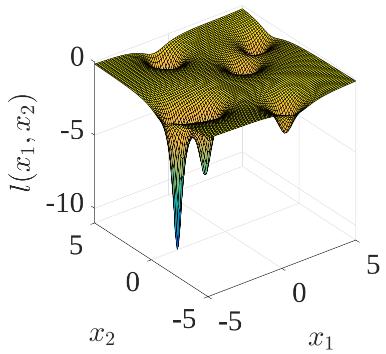


(a) Outline (b) Contour

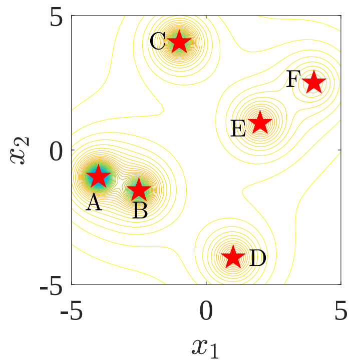
Fig. A.7: 2^N minima Function

8. Shekel's Function

$$\begin{cases} l(\boldsymbol{x}) = -\sum_{i=1}^{6} \left[(\boldsymbol{x} - \boldsymbol{a}_i)(\boldsymbol{x} - \boldsymbol{a}_i)^\top + c_i\right]^{-1} \\ \boldsymbol{x}^* = \boldsymbol{a}_i, \quad l(\boldsymbol{x}^*) \cong \frac{1}{c_i} \end{cases} \tag{A.1}$$



(a) Outline



(b) Contour

Fig. A.8: Shekel's Function

Table A.1: Each Parameter of the Eq.(A.1)

i	$\boldsymbol{a}_i$	c_i	Superior Solution	$l(\boldsymbol{a_i})$
1	$[-4, -1, \cdots, -4, -1]^N$	0.1	A	-10.47
2	$[-2.5, -1.5, \cdots, -2.5, -1.5]^N$	0.2	B	-5.52
3	$[-1, 4, \cdots, -1, 4]^N$	0.2	C	-5.16
4	$[1, -4, \cdots, 1, -4]^N$	0.4	D	-2.65
5	$[2, 1, \cdots, 2, 1]^N$	0.4	E	-2.79
6	$[4, 2.5, \cdots, 4, 2.5]^N$	0.7	F	-1.66

Table A.1 shows the parameters $\boldsymbol{a}_i$, c_i $(i = 1, 2, \cdots, 6)$, and Fig. A.8 shows a contour line of this function. This function has superior solution set changing for the parameters δ and ε. In Fig. A.8, this function has six local optimal solutions with A $\sim$ F marked by ★, and the global optimal solution is A.

B

k-means Clustering

B.1 Overview of k-means Clustering

Clustering is the division of a set to be classified into subsets that achieve internal cohesion and external isolation [62, 63, 64]. Classification is to make the similarity of elements belonging to the same cluster as large as possible, while the similarity of elements between different clusters is as small as possible.

The k-means clustering is one of the typical algorithms for clustering proposed by Macqueen belonging to the hard clustering method [62]. The definition of hard clustering refers to dividing a database $\boldsymbol{X}$ consisting of m objects into K clusters, and each cluster $\boldsymbol{U}_k (k = 1, 2, \cdots, K)$ satisfies as follows conditions.

- All objects always belong to one cluster.
- An object cannot belong to more than one cluster.
- There is no cluster that contains no objects.

Furthermore, the hard clustering can be described by the following Eq.(B.1).

$$\boldsymbol{X} = \boldsymbol{U}_1 \cup \boldsymbol{U}_2 \cup \cdots \cup \boldsymbol{U}_K, \quad \boldsymbol{U}_i \cap \boldsymbol{U}_j = \phi \quad (i \neq j) \tag{B.1}$$

B.2 Algorithm of k-means Clustering

Given a set of observations $(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n)$, where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into $k(\leq n)$ sets $\boldsymbol{U} = (\boldsymbol{U}_1, \boldsymbol{U}_2, \cdots, \boldsymbol{U}_K)$ so as to minimize the within-cluster sum of squares (WCSS). Formally, the objective is to find by Eq.(B.2)

$$\text{WCSS} = \underset{\boldsymbol{U}}{\arg\min} \sum_{k=1}^{K} \sum_{\boldsymbol{x}_p \in \boldsymbol{U}_k} ||\mathcal{G}_k - \boldsymbol{x}_p||^2 \tag{B.2}$$

where $\mathcal{G}_k$ is the mean of points in $\boldsymbol{U}_k$ according to Eq.(B.3).

$$\mathcal{G}_k = \frac{1}{|\boldsymbol{U}_k|} \sum_{\boldsymbol{x}_p \in \boldsymbol{U}_k} \boldsymbol{x}_p \tag{B.3}$$

Assign each observation $\boldsymbol{x}_p$ into the cluster U_k with the nearest centroid $\mathcal{G}_k$ according to Eq.(B.4).

$$\boldsymbol{U}_k = \left\{ \boldsymbol{x}_p | \, \|\boldsymbol{x}_p - \mathcal{G}_k\|^2 \geq \|\boldsymbol{x}_p - \mathcal{G}_j\|^2 \;\; \forall j, 1 \leq j \leq K \right\} \tag{B.4}$$

The following **Algorithm B.1** shows the k-means clustering.

Algorithm B.1 k-means clustering

```
procedure k-MEANS CLUSTERING(K, ϵ)
  Step 1: Preparation
    Set the number of clusters K and the termination condition ς.
    Set the number of iterations t = 1.
  Step 2: Initialization
    Give a set of observations (x_1, x_2, ··· , x_m).
    Randomly select K observations from the set of observations as the centroid 𝒢_k^1 (k = 1, ··· , K)
     of each cluster U_k^1 (k = 1, 2, ··· , K).
  Step 3: Cluster assignment
    for k = 1 to K do
      for p = 1 to m do
        Assign each observation x_p into the cluster U_k^t with the nearest centroid G_k^t.
        U_k^t = { x_p | ||x_p − 𝒢_k^t||^2 ≥ ||x_p − 𝒢_j^t||^2  ∀j, 1 ≤ j ≤ K }
      end for
    end for
  Step 4: Update
    for k = 1 to K do
      Recalculate centroid 𝒢_k^{t+1} for observations assigned to each cluster.
      𝒢_k^{t+1} = 1/|U_k^t| Σ_{x_p ∈ U_k^t} x_p
    end for
  Step 6: Termination
    Calculate the minimize the within-cluster sum of squares WCSS.
    WCSS = arg min_U Σ_{k=1}^{K} Σ_{x_p ∈ U_k^t} ||𝒢_k^t − x_p||^2
    if ϵ ≤ WCSS then
      The algorithm is terminated.
    else
      Return to Step 3.
end procedure
```

In this paper, numerical experiments were conducted with the value of ϵ as 10^{-6}. In addition, the k-means clustering requires the user to set the number of clusters in advance. Since the x-means clustering [85] that does not require this has been proposed, please refer to it when using the clustering method in advancing the research.

C Crossover and Mutation for Real Genetic Algorithm

C.1 Simulated Binary Crossover

Simulated Binary Crossover (SBX) [86] is a crossover method proposed by Deb et al. There are various implementation methods, but in this paper, according to nsga2-gnuplot-v1.1.6 published in the Reference [87], the implementation considers the constraints of the coefficient of determination. The algorithm of Simulated Binary Crossover is shown in **Algorithm C.1** and **Algorithm C.2**.

C.2 Polynomial Mutation

Polynomial Mutation (PM) [76] is also a mutation method proposed by Deb et al. In this paper, the implementation takes into account the constraints of the coefficient of determination according to nsga2-gnuplot-v1.1.6 published in the Reference [87]. The algorithm of Polynomial Mutation is shown in **Algorithm C.3**.

Algorithm C.1 Simulated Binary Crossover I (SBX I)

```
procedure SBX I(parent_1, parent_2, p_c, η_c, x_l, x_u)
%parent_1, parent_2 : Parent individual, p_c : Crossover rate, η_c : Distribution parameter
%x_l : Lower limit vector, x_u : Upper limit vector, RAND() ∈ [0, 1] : a uniform random number
    child_1 = child_2 = 0                                    ▷ Initialization of child
    if RAND() < p_c then
        for j = 1, . . . , n do
            if RAND() < 0.5 then
                if parent_1j = parent_2j then
                    child_1j = parent_1j
                    child_2j = parent_2j
                else
                    if parent_1j < parent_2j then
                        y_1 = parent_1j
                        y_2 = parent_2j
                    else
                        y_1 = parent_2j
                        y_2 = parent_1j
                    (c_1, c_2)=SBX II(y_1, y_2, x_lj, x_uj, η_c)     ▷ See Algorithm C.2
                if RAND()< 0.5 then
                    child_1j = c_2
                    child_2j = c_1
                else
                    child_1j = c_2
                    child_2j = c_1
            else
                child_1j = parent_1j
                child_2j = parent_2j
        end for
    else
        child_1 = parent_1
        child_2 = parent_2
end procedure
```

Algorithm C.2 Simulated Binary Crossover II (SBX II)

procedure SBX II$(y_1, y_2, p_c, \eta_c, x_l, x_u)$
    $r = \text{RAND}()$
    $\beta = 1 + 2 * \frac{c_1 - x_{lj}}{y_2 - y_1}$
    $\alpha = 2 - \beta^{-\eta_c+1}$
    if $r \leq \frac{1}{\alpha}$ **then**
        $\beta_q = (r \times \alpha)^{\frac{1}{\eta_c+1}}$
    else
        $\beta_q = (\frac{1}{2 - r \times \alpha})^{\frac{1}{\eta_c+1}}$
    $c_1 = 0.5((y_1 + y_2) - \beta_q(y_2 - y_1))$

    $\beta = 1 + 2 * \frac{x_{uj} - y_2}{y_2 - y_1}$
    $\alpha = 2 - \beta^{-\eta_c+1}$
    if $r \leq \frac{1}{\alpha}$ **then**
        $\beta_q = (r \times \alpha)^{\frac{1}{\eta_c+1}}$
    else
        $\beta_q = (\frac{1}{2 - r \times \alpha})^{\frac{1}{\eta_c+1}}$
    $c_2 = 0.5((y_1 + y_2) - \beta_q(y_2 - y_1))$

    if $c_1 < x_l$ **then**
        $c_1 = x_l$
    if $c_2 < x_l$ **then**
        $c_2 = x_l$
    if $c_1 > x_u$ **then**
        $c_1 = x_u$
    if $c_2 > x_u$ **then**
        $c_2 = x_u$
end procedure

Algorithm C.3 Polynomial Mutation

procedure POLYNOMIAL MUTATION($\boldsymbol{parent}, p_m, \eta_m, \boldsymbol{x}_l, \boldsymbol{x}_u$)
%$\boldsymbol{parent}$: Parent individual, p_m : Mutation rate, η_m : Distribution parameter
%$\boldsymbol{x}_l$: Lower limit vector, $\boldsymbol{x}_u$: Upper limit vector, RAND()$\in [0,1]$: a uniform random number
  for $j = 1, \ldots, n$ **do**
    if RAND() $< p_m$ **then**
      $y = parent_j$
      $\delta_1 = \frac{y - x_{lj}}{x_{uj} - x_{lj}}$
      $\delta_2 = \frac{x_{uj} - y}{x_{uj} - x_{lj}}$
      $r =$ RAND()
      if $r \leq 0.5$ **then**
        $a = 2r + (1 - 2r) \times (1 - \delta_1)^{\eta_m + 1}$
        $\delta_q = a^{\frac{1}{\eta_m + 1}} - 1$
      else
        $a = 2(1 - r) + 2(r - 0.5) \times (1 - \delta_2)^{\eta_m + 1}$
        $\delta_q = 1 - a^{\frac{1}{\eta_m + 1}}$
      $y = y + \delta_q(x_{uj} - x_{lj})$
      if $y < x_{lj}$ **then**
        $y = x_{lj}$
      if $y > x_{uj}$ **then**
        $y = x_{uj}$
      $parent_j = y$
  end for
end procedure

C.3 Real Genetic Algorithm

Genetic Algorithm (GA) with Simulated Binary Crossover and Polynomial Mutation for the minimization problem of the objective function $f(\boldsymbol{x})$ ($\boldsymbol{x} \in \mathbb{R}^N$) is shown by **Algorithm** C.4.

Algorithm C.4 Genetic Algorithm (GA)

---

procedure GA(m, p_c, p_m, T_{max})
  Step 1: Initialization
    Give initial solutions $\boldsymbol{P}^1$ ($|\boldsymbol{P}^1| = m$), Set $t = 1$
  Step 2: Generation of New Solutions
    $\boldsymbol{Q}^t = \emptyset$
    for $i = 1 \dots m$ **do**
      Choose randomly $\boldsymbol{x}_a, \boldsymbol{x}_b \in \boldsymbol{P}^t$ ($a \neq b$)
      $\{\boldsymbol{y}_a, \boldsymbol{y}_b\}$ = SIMULATED BINARY CROSSOVER($\boldsymbol{x}_a, \boldsymbol{x}_b$) ▷ See Algorithm C.1
      Choose randomly $\boldsymbol{y} \in \{\boldsymbol{y}_a, \boldsymbol{y}_b\}$
      $\boldsymbol{q}$ = POLYNOMIAL MUTATION($\boldsymbol{y}$) ▷ See Algorithm C.3
      $\boldsymbol{Q}^t := \boldsymbol{Q}^t \cup \{\boldsymbol{q}\}$
    end for
  Step 3: Fitness Assignment
    $\boldsymbol{U}^t = \boldsymbol{P}^t \cup \boldsymbol{Q}^t$ ($|\boldsymbol{U}^t| = 2m$)
    for each $\boldsymbol{x}^t \in \boldsymbol{U}^t$ **do**
      for $i = 1 \dots 2m$ **do**
        $fit_i^t = f(\boldsymbol{x}_i^t)$
      end for
    end for
  Step 4: Selection
    Sort $\boldsymbol{U}^t$ in ascending order using fit_i^t
    $\boldsymbol{P}^{t+1} := \boldsymbol{U}^t[1:m]$
  Step 5: Termination
    if $t < T_{\max}$ **then**
      $t := t + 1$
      Go to **Step 2**
    else
end procedure

References

1. Reference

[1] J. Tsuchiya, T. Suzuki, K. Yasuda: "Application of Integrated Optimization to Optimal Design of Surface Motor," IEEJ Transaction C: Electronics, Information & Systems, Vol.130, No.1, pp.108-109 (2010) (in Japanese)

[2] A. Takezawa, S. Nishiwaki, and M. Kitamura: "Shape and Topology Optimization Based on the Phase Field Method and Sensitivity Analysis," Journal of Computational Physics, Vol.229, No.7, pp.2697-2718 (2010)

[3] K. Morikawa and N. Nakamura: "An Optimization Approach for Job Shop Planning and Scheduling," Journal of Japan Industrial Managemengt Association, Vol.51, No.3, pp.168-176 (2000) (in Japanese)

[4] K. Fujita, G. Emoto, T. Takeshita, and T. Sanaka: "Application of Optimization Technology in the Chemical Industry," Operations Research as a Management Science Research, Vol.48, No.8, pp.549-554 (2003) (in Japanese)

[5] H. Kise: "Production Scheduling-A State of the Art," Systems, Control and Information Vol.41, No.3, pp.92-99 (1997)

[6] K. Masakazu: "Survey of Optimization Methods for Power Systems Operation," The transactions of the Institute of Electrical Engineers of Japan. B, A publication of Power and Energy Society, Vol.121, No.2, pp.160-163 (2001) (in Japanese)

[7] M. Hiroyuki: “Recent Trends on Applications of Meta-heuristics in Power Systems,” The transactions of the Institute of Electrical Engineers of Japan. B, A publication of Power and Energy Society, Vol.123, No.10, pp.1120-1123 (2003) (in Japanese)

[8] F. Yoshikazu: “Applications of Meta-heuristics to Power and Energy Fields,” The Trans. of the Institute of Electrical Engineers of Japan. B, A publication of Power and Energy Society, Vol.124, No.5, pp.679-682 (2004) (in Japanese)

[9] S. Tokui: “The Idea of Deep Learning from the Perspective of Optimization,” Operations Research as a Management Science Research, Vol.60, No.4, pp.191-197 (2015) (in Japanese)

[10] H. Konno and H. Yamazaki: “Mean-Absolute Deviation Portfolio Optimization Model and Its Application to Tokyo Stock Market,” Management Science, Vol.37, No.5, pp.519-531 (1991)

[11] R. Wakai, T. Shinzato, and Y. Shimazaki: “Random Matrix Approach for Portfolio Optimization Problem,” Journal of Japan Industrial Management Association, Vol.65, No.1, pp.17-28 （2014） (in Japanese)

[12] G. Dantzig: “Linear Programming and Extensions,” Princeton University Press (1998)

[13] K. Tanaka: “Convex Analysis and Optimization Theory,” Makino Shoten (1994)

[14] H. Konno and H. Yamashita: “Nonlinear Programming,” JUSE (1978)

[15] K. Shimizu and E. Aiyoshi: “Mathematical Programming,” Shokodo (1984)

[16] Masatoshi Sakawa: “Optimization of Nonlinear Systems -From One Objective to Multi-objective-,” Morikita Publishing (1986)

[17] H. Konno: “Introduction to Mathematical Determination Method -Campus OR-,”Asakura Shoten (1992)

[18] M. Fukushima: “Basics of Nonlinear Optimization,”Asakura Shoten (2001)

[19] H. Tamaki: “System Optimization,” Ohmsha (2005)

[20] H. Yabe: “Optimization and Its Applications: Engineering Basics,” SAIENSU-SHA (2006)

[21] K. Fujisawa and S. Umetani: “50 Optimization Problems to Help You Apply,” Asakura Shoten (2009)

[22] E. Aiyoshi and K. Yasuda: “Metaheuristics and Their Applications,” Tokyo: The Institute of Electrical Engineers of Japan, Ohmsha (2007)

[23] K. Yasuda: “The Present and The Future of Metaheuristics,” Journal of the Society of Instrument and Control Engineers, Vol.47, No.6, pp.453-458 (2008) (in Japanese)

[24] K. Yasuda: “Evolutionary Computation and Metaheuristics,” Journal of the Society of Instrument and Control Engineers, Vol.122, No.3, pp.320-323 (2002) (in Japanese)

[25] I. Boussaïd, J. Lepagnot, and P. Siarry: “A Survey on Optimization Metaheuristics,” Information Sciences, Vol.237, pp.82-117, Elsevier Publisher (2013)

[26] X. S. Yang: “Nature-Inspired Metaheuristic Algorithms,” Second Edition, Luniver Press (2010)

[27] J. Kennedy and R. C. Eberhart: “Particle Swarm Optimization,” Proc. of IEEE International Conference on Neural Networks, Vol.4, pp.1942-1948 (1995)

[28] Shi Y., Eberhart R.C.: “Modified Particle Swarm Optimizer,” Proc. of IEEE International Conference on Evolutionary Computation, pp.69–73 (1998)

[29] J. Kennedy: “The Particle Swarm: Social Adaptation of Knowledge,” Proc. of 1997 IEEE International Conference on Evolutionary Computation (ICEC’97),

pp.303-308 (1997)

[30] R. Poli: "Analysis of the Publications on the Applications of Particle Swarm Optimisation," Journal of Artificial Evolution and Applications (2008)

[31] J. H. Holland: "Adaptation in Natural and Artificial Systems," University of Michigan Press (1975)

[32] D. E. Goldberg: "Genetic Algorithm in Search, Optimization, and Machine Learning," Addison-Wesley (1989)

[33] P. A. Vikhar: "Evolutionary algorithms: A Critical Review and Its Future Prospects," Proc. of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), Jalgaon, pp.261–265 (2016)

[34] R. Storn and K. V. Price: "Differential Evolution: A Simple and Efficient Heuristic for Global Optimization over Continuous Space," Journal of Global Optimization, Vol.11, No.4, pp.341-359 (1997)

[35] K. Price, R. Storn, and JA. Lampinen: "Differential Evolution: A Practical Approach to Global Optimization," Springer Science & Business Media (2006)

[36] D. Karaboga: "An Idea Based on Honey Bee Swarm for Numerical Optimization," Technical Report TR06, Computer Engineering Department, Engineering Faculty, Erciyes University (2005)

[37] A. Bahriye and D. Karaboga: "A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization," Information Sciences, 192, pp.120-142 (2012)

[38] X. S. Yang: "Firefly Algorithms for Multimodal Optimization," Stochastic Algorithms: Foundations and Applications, Lecture Notes in Computer Sciences, Vol.5792, pp.169-178 (2009)

[39] X. S. Yang and X. S. He: "Firefly Algorithms: Recent Advances and Applica-

tions," International Journal of Swarm Intelligence, Vol.1, No.1, pp.36-50 (2013)

[40] X. S. Yang: "Engineering optimization: an introduction with metaheuristic applications," John Wiley & Sons, 2010

[41] K. Yasuda, A. Ishigame: "Nonlinear Programming Algorithm : From the Practical Viewpoint," Journal of the Society of Instrument and Control Engineers, Vol.50, No.9, pp.344-349 (2006) (in Japanese)

[42] Y. Fukuyama: "Applications of Metaheuristics to Power and Energy Fields," IEEJ Trans. on Power and Energy, Vol. 124, No.5, pp.550-557 (2004) (in Japanese)

[43] J. Inagaki, M. Haseyama, and H. Kitajima:"A Note on Deciding Various Solutions by Genetic Algorithm for Routing Applications," IEICE technical Report, Circuits and Systems, Vol.82, pp.1102-1111 (1999) (in Japanese)

[44] F. Satoshi, H. Shigeto, and Y. Yusuke: "Application of an Optimization Technique to Production and Logistics Scheduling," JEE Technical Reports, No.14, pp.49-54 (2006) (in Japanese)

[45] W. Daisuke and H. Toshio: "Decent Solution Search and Diversity of Solution in Form Decision for Free Surface Shell Structure," Journal of Structural Engineering. B, Vol.28B, pp.453-460 (2012) (in Japanese)

[46] L. Teng, T. Isumi, X. Lu, F. Wakui: "A Method of the Optimum Route Search by Fuzzy-AHP," IEEJ Trans. on Electronics, Information and Systems, Vol.133, No.6, pp.1269-1276 (2013) (in Japanese)

[47] S. Obayashi: "Multiobjective Optimization for Aerodynamic Design of Aircraft," ISCIE Trans., Vol. 47, No.6, pp.253-258 (2003) (in Japanese)

[48] S. Watanabe and R. Minato: "Development of a Design Support System that Can Efficiently Utilize Non-dominated Solutions -Through an Example of Tutbojet Engine Design-," Trans. of the Japanese Society for Artificial Intelligence, Vol.24,

No.1, pp.1-12 (2009) (in Japanese)

[49] S. Kuninobu and K. Handa: "A Parts Arrangement Algorithm for Note PC -An Algorithm for 3D Bin Packing Problem with Arrangement Constraints," Proc. of 23th International Conference on Circuits/Systems Computers and Communications 2008, pp.329-332 (2008)

[50] K. Tagawa, K. Mizutani, K. Inoue, and H. Haneda: "An Imanishism-based Genetic Algorithm for Seeking Various Optimal Solutions of the Module Placement Problem," Trans. of the Institute of Systems, Control and Information Engineers, Vol.14, No.10, pp.467-474 (2001) (in Japanese)

[51] K. Ishikawa, K. Masuda, and T. Sekozawa: "Multiple Optimal Solutions Search by Using Swarm Division Particle Swarm Optimization Algorithm which Utilizes the Distribution of Personal Best Solutions," IEEJ Trans. on Electronics, Information and Systems, Vol.134, No.9, pp.1373-1383 (2014) (in Japanese)

[52] X. Li: "Niching Without Niching Parameters: Particle Swarm Optimization Using a Ring Topology," IEEE Trans. on Evolutionary Computation, Vol.14, No.1, pp.150-169 (2010)

[53] K. Yasuda, N. Iwasaki, G. Ueno and E. Aiyoshi: "Particle Swarm Optimization: A Numerical Stability Analysis and Parameter Adjustment Based on Swarm Activity," IEEJ Trans. on Electrical and Electronic, Vol.3, No.6, pp.642-659 (2008)

[54] C.H.Papadimitriou and K.Steiglitz: "Combinatorial Optimization," Dover (1998)

[55] G.L.Nemhauser and L.A.Wolsey: "Integer and Combinatrial Optimization," Wiley (1988)

[56] L.A.Wolsey: "Integer Programming," Wiley (1998)

[57] T.A.Ciriani and R.C.Leachman Eds: "Optimization in Industry/Optimization

in Industry 2," Wiley (1993)

[58] Richard Kipp Martin: "Large Scale Linear and Integer Optimization," Kluwer (1999)

[59] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen: "Evolutionary Algorithms for Solving Multi-Objective Problems," 2nd ed., Springer (2007)

[60] R. Oosumi, W. Kumagai, K. Tamura, and K. Yasuda: "A Superior Solution Set Search Problem for Single-Objective Optimization and A Firefly Algorithm Based Optimization Method, " IEEJ Trans. on Electronics, Information and Systems, Vol.136, No.10, pp.1947-1948 (2016) (in Japanese)

[61] R. Oosumi, K. Tamura, and K. Yasuda:"Novel Single-objective Optimization Problem and Firefly Algorithm-based Optimization Method," Proc. of IEEE International Conference on Systems, Man, and Cybernetics 2016, pp.1011-1015 (2016)

[62] J.Macqueen: "Some Methods for Classification and Analysis of Multivariate Observations," Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, pp.281-297 (1967)

[63] S. P. Lloyd: "Least square quantization in PCM," IEEE Trans. on Information Theory, Vol.IT-28, NO.2, pp.129-137 (1982)

[64] B.S.Everitt: "Cluster Analysis," Edward Arnold, 5th edition (2011)

[65] W. Kumagai, K. Tamura, J. Tsuchiya, and K. Yasuda: "Adaptive Cuckoo Search Based on Evaluation and Control," IEEJ Trans. on Electronics, Information and Systems, Vol.136, No.11, pp.1596-1609 (2016) (in Japanese)

[66] Y. Mutsunori and I. Toshihide: "Combinatorial Optimization-Focus On Meta-Strategy-," Asakura Publishing (2001)

[67] A. Ido and K. Yasuda: "A Basic of The Adaptive Particle Swarm Optimization," IEEJ Trans. on Electronics, Information and Systems, Vol.124, No.2, pp.550-557

(2004) (in Japanese)

[68] Hu, X. and R.C. Eberhart: "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems," Proc. of The IEEE Congress on Evolutionary Computation (CEC 2002), pp.1666-1670 (2002)

[69] Y. Zhang, L. Wang, and Q. Wu: "Modified Adaptive Cuckoo Search (MACS) Algorithm and Formal Description for Global Optimisation," International Journal of Computer Applications in Technology, Vol.44, No.2, pp.1-6 (2012)

[70] S. Walton, O. Hassan, K., Morgan, and M. R. Brown: "Modified Cuckoo Search: A New Gradient Free Optimisation Algorithm," Chaos, Solitons & Fractals, Vol.44 No.9, pp.710-718 (2011)

[71] E. Valian, S. Mohanna, and S. Tavakoli: "Improved Cukoo Search Algorithm for Global Optimization," International Journal of Communications and Information Technology, Vol.1, No.1, pp.31-44 (2011)

[72] X. Hu and R.C. Eberhart: "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems," Proc. of The IEEE Congress on Evolutionary Computation (CEC 2002), pp.1666-1670 (2002)

[73] K. Yasuda, N. Iwasaki, G. Ueno, and E. Aiyoshi: "Particle Swarm Optimization: A Numerical Stability Analysis and Parameter Adjustment Based on Swarm Activity," IEEJ Trans. on Electrical and Electronic, Vol.3, No.6, pp.642-659 (2008)

[74] M. Naik, M. R. Nath, A. Wunnava, S. Sahany, and R. Panda: "A new adaptive Cuckoo search algorithm," Proc. of 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), pp.1-5 (2015)

[75] P. Ong: "Adaptive Cuckoo Search Algorithm for Unconstrained Optimization," The Scientific World Journal, pp.1-6 (2014)

[76] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan: "A Fast and Elitist Multiob-

jective Genetic Algorithm: NSGA-II", IEEE Trans. on Evolutionary Computation, Vol.6, No.2, pp.182-197 (2002-4)

[77] E. Zitzler, M. Laumanns, and L. Thiele: "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich (2001)

[78] R. Thomsen: "Multimodal Optimization Using Crowding-based Differential Evolution," Proc. of 2004 IEEE Congress on Evolutionary Computation, pp.1382-1389 (2004)

[79] K. Deb: "Multi-objective Optimization Using Evolutionary Algorithms," John Wiley & Sons, 2001

[80] C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen: "Evolutionary Algorithms for Solving MultiObjective Problems," 2nd edition, Springer, 2007

[81] J. D. Schaffer: "Multiple objective optimization with vector evaluated genetic algorithms," In Proceedings of the First International Conference on Genetic Algorithms, pp.93–100 (1985)

[82] B. Kevin, J. Goldstein, R. Ragha, and S. Uri: "When is "nearest neighbor meaningful?", International conference on database theory, Springer, Berlin, Heidelberg, pp.217-235 (1999)

[83] A. Charu C, H. Alexander, and K. Daniel A: "On the Surprising Behavior of Distance Metrics in High Dimensional space", International conference on database theory, Springer, Berlin, Heidelberg, pp.420-434 (2010)

[84] H. Ishikawa, N. Niimi, Y. Shiraishi, and S. Yokoyama: "Data Mining and Collective Intelligence-From Basics to Web and Social Media-," Kyoritsu Shuppan (2012)

[85] D.Pelleg and A.Moore: "X-means: Extending K-means with Efficient Estimation

of the Number of Clusters," ICML. Vol.1, pp.723-734 (2000)

[86] K. Deb and R. B. Agrawal: "Simulated Binary Crossover for Continuous Search Space," Complex Systems, Vol.9, No.2, pp.115-148 (1995)

[87] Kanpur Genetic Algorithms Laboratory
http://www.iitk.ac.in/kangal/index.shtml

2. Author's Publications

(a) Full Papers

[88] R. Oosumi, H. Wang, K. Tamura, and K. Yasuda : "Cluster-structured Firefly Algorithm for Superior Solution Set Search Problem," IEEJ Trans. on Electronics, Information and Systems, Vol.137 No.10 pp.1431-1432 (2017.6)

[89] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Adaptive FA Based on Evaluation and Control of Search State for Superior Solution Set Search Problem," IEEJ Trans. on Electrical and Electronic Engineering, Vol.13, No.12, pp.1783-1794 (2018.6)

[90] R. Fukushima, H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Dominance Relation-based Genetic Algorithm for Superior Solution Set Search Problem," IEEJ Trans. on Electrical and Electronic Engineering, Vol.14, No.6, pp.959-960 (2019.2)

[91] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Superior Relation-Based Firefly Algorithm for Superior Solution Set Search Problem," IEEJ Trans. on Electrical and Electronic Engineering, Vol.14, No.12, pp.1796-1804 (2019.12)

(b) International Conferences

[92] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Firefly Algorithm Using Cluster Information for Superior Solution Set Search," 2017 IEEE International Conference on Systems, Man, and Cybernetics, pp.3695-3699 (2017.10)

[93] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Adaptive Firefly Algorithm Based on Diversification and Intensification for Superior Solution Set Search," 2018 IEEE International Conference on Systems, Man, and Cybernetics, pp.2794-2801 (2018.10)

(c) Japanese National Conferences

[94] H. Wang, R. Oosumi, K. Tamura, J. Tsuchiya, and K. Yasuda: "A Basic Study of Firefly Algorithm Using the Cluster Information in the Superior Solution Set Search," Proc. of Symposium on Evolutionary Computation 2016, P2-06, p.176-183 (2016.12)

[95] H. Wang, R. Oosumi, K. Tamura, J. Tsuchiya, and K. Yasuda: "Adaptive Firefly Algorithm Based on Evaluation and Control of Cluster Search State for Superior Solution Set Search Problem," 2017 IEEJ Conference on Electronics, Information and Systems, GS2-2，pp.1232-1237 (2017.9)

[96] H. Wang, R. Oosumi, K. Tamura, J. Tsuchiya, and K. Yasuda: "A Basic Study of Firefly Algorithm Utilizing Cluster Information for Superior Solution Set Search Problem," 2017 IEEJ Conference on Electronics, Information and Systems, SS1-5, pp.1647-1678 (2017.9)

[97] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Adaptive Firefly Algorithm: Evaluation and Control of Search State for Superior Solution Set Search Problem," Proc. of SICE Symposium on Systems and Information 2017, SS11-5，pp.769-774 (2017.11)

[98] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Basic Study of a Superior Solution Set Search Method based on Ranking of Superior Solutions," 2018 IEEJ Conference on Electronics, Information and Systems, MC1-7, pp.913-918 (2018.9)

[99] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Adaptive Firefly Algorithm Based on Diversification and Intensification in Superior Solution Set Search Problem," 2018 IEEJ Conference on Electronics, Information and Systems, SS3-1, pp.1633-1634 (2018.9)

[100] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Firefly Algorithm Based on Superior Relations in Superior Solution Set Search Problem," Proc. of SICE Symposium on Systems and Information 2018, GS01-07 (2018.11)

[101] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Firefly Algorithm Based on Superior Relation in Superior Solution Set Search Problem," The 32nd Workshop on Circuits and Systems, D1-1, pp. 37-42 (2019.8)

[102] H. Wang, K. Tamura, J. Tsuchiya, and K. Yasuda: "Superior Relation Based on Firefly Algorithm in Superior Solution Set Search," 2019 IEEJ Conference on Electronics, Information and Systems, SS3-8, pp. 1592-1593 (2019.9)

3. Author's Award History

[103] SICE Symposium on Systems and Information 2017, Received the SSI Excellent Paper Award
Award-Winning Paper: "Adaptive Firefly Algorithm: Evaluation and Control of Search State for Superior Solution Set Search Problem"
Award Date: November 7, 2017
https://www.sice.or.jp/org/SSI2017/awards.php

[104] 2019 Annual Conference on Electronics, Information and Systems, I.E.E. of

Japan, Received Student Competition Session: Outstanding Student Presentation Award

Award-Winning Paper: "Superior Relation Based on Firefly Algorithm in Superior Solution Set Search"

Award Date: September 6, 2019

http://denki.iee.jp/eiss/wp-content/uploads/eiss/conf2019/2019_ss_award_list.pdf