Examination of Multi-objective Optimization Method for Global Search Using DIRECT and GA

Luyi WANG, Hiroyuki ISHIDA, Tomoyuki HIROYASU, Mitsunori MIKI

***Abstract*— A number of multi-objective genetic algorithms (MOGAs) have been developed to obtain Pareto optimal solutions for multi-objective optimization problems. However, as these methods involve probabilistic algorithms, there is no guarantee that the global search will be conducted in the design variable space. In such cases, there are unsearched areas in the design variable space, and the obtained Pareto solutions may not be truly optimal. In this paper, we propose an optimization method called NSDIRECT-GA to conduct a global search over as much as possible of the design variable space, which improves the reliability of the obtained Pareto solutions. The effectiveness of NSDIRECT-GA was examined through numerical experiments. By NSDIRECT-GA, not only the optimal solutions but also the information of the landscape can be determined, and it is possible to obtain Pareto solutions with higher reliability than with normal MOGAs.**

I. INTRODUCTION

Multi-objective optimization problems are those in which multiple objectives must be optimized. As these multiple objectives often conflict with each other, a single optimum solution is often not available. Therefore, the goal in such cases is to obtain Pareto optimal solutions, which are solutions with objective values that cannot be simultaneously improved without degradation of at least one other value.

There have been many recent studies of the adaptation of Genetic Algorithms (GAs) to multi-objective optimization [1], [2], [3], [4], [5], [6]. This is because a GA search is performed by multiple individuals, and it is capable of obtaining Pareto solutions in a single search. Adaptation of GAs to multi-objective optimization is called Multi-objective Genetic Algorithm (MOGA). Of the many MOGAs that have been developed to date, NSGA-II [5] and SPEA2 [6] are known to perform well.

However, GAs and MOGAs are probabilistic algorithms, and there is no guarantee that a global search of the design variable search will be conducted. A MOGA search is strongly influenced by non-dominated solutions, and tends to concentrate in the area in which the non-dominated solutions exist, resulting in an uneven search. Furthermore, when an individual with high fitness value that overwhelms other individuals is created in a MOGA search, its genetic information will quickly spread throughout the population and result in early convergence. There will be unsearched areas in a MOGA search due to the high influence of the non-dominated solutions, and the obtained Pareto solutions lack reliability with regard to being truly optimal. Therefore, it is necessary to consider a mechanism to perform a global search of the design variable space as much as possible, and to improve the reliability of the obtained Pareto solutions.

In this paper, we propose an optimization method called NSDIRECT-GA, which is a combination of a global search algorithm of DIRECT [7] and MOGA. NSDIRECT-GA executes a global search of the design variable space as much as possible, and reduces unsearched areas. It also makes it possible to roughly understand the landscape of the search space when the optimization is finished.

Luyi Wang is with the Department of Knowledge Engineering and Computer Science, Doshisha University, JAP (phone: 81-774-65-6924; fax: 81-774-65-6780; email: lou@mikilab.doshisha.ac.jp).

II. GLOBAL SEARCH METHOD

To improve the reliability of the solutions derived by a MOGA, it is necessary to search throughout the design variable space without leaving any areas unsearched. Global search becomes important for this purpose, and a single objective optimization method called DIRECT [7] has attracted a great deal of attention in this field. However, it has been confirmed that the convergence of DIRECT is slow when applied to multimodal functions [8].

In this paper, we propose an optimization method called NSDIRECT-GA (Non-dominated Sorting DIRECT-GA), which adapts DIRECT to multi-objective optimization and combines it with MOGA. The following sections will explain DIRECT and the modifications made to adapt DIRECT to multi-objective optimization.

A. DIRECT

DIRECT (DIviding RECTangle) is a method that searches for the optimal solution by considering design variable space to be an N-dimension hyper-cube (referred to as “boxes” in this paper), dividing the hyper-cube, and sampling the center points. The DIRECT algorithm is given follows:

1) Normalization
2) Division of the Hyper-cube
3) Potentially Optimal Hyper-rectangles
4) Division of the Hyper-rectangles
5) Repeat 3, 4 until the termination rule is satisfied

Each operation is described in the following sections.

1) Normalization and Division of the Hyper-cube: DIRECT begins the search by transforming the domain of the target problem into the unit hyper-cube:

$$\overline{\Omega} = \{x \in R^N : 0 \leq x_i \leq 1\} \tag{1}$$

Then, the center point of the hyper-cube c_1 is sampled. Next, DIRECT divides this space by evaluating the function values at the points $c_1 \pm \delta \overrightarrow{e_i} (i = 1, ..., N)$, where δ is one-third the side length of the hyper-cube, and $\overrightarrow{e_i}$ is the ith

Euclidean base-vector. That is, a hyper-cube is divided into three hyper-rectangles in each dimension.

The sequence of the dimensions to be divided is determined by w_i, which is shown in (2), and the first division is performed in the dimension with the smallest w_i.

$$w_i = min(f(c_1 + \delta e_i), f(c_1 - \delta e_i)) \quad (2)$$
$$i = 1, ..., N$$

This operation is repeated for all dimensions on the box with the point c_1, choosing the next dimension with the next smallest w_i. Fig. 1 illustrates the search space after the initial divisions. The numbers in the figure on the left of Fig. 1 show the function values at each point. In this case, w_1 = 60.0 and w_2 = 150.1, so the first division is performed along the direction of x_1.

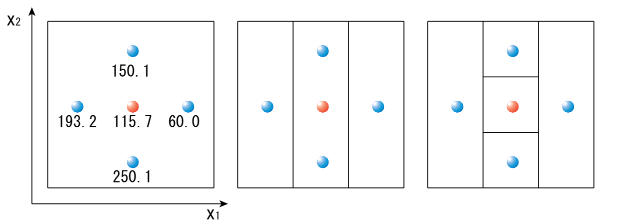


Fig. 1. Design variable space after the first division

2) Potentially Optimal Hyper-rectangles: DIRECT divides all of the hyper-rectangles that satisfy the definition of potentially optimality:

Let $\epsilon > 0$ be a positive constant and let f_{min} be the current best function value. A hyper-rectangle j is potentially optimal if there exists some $K > 0$ such that

$$f(c_j) - Kd_j \leq f(c_i) - Kd_i, \forall i, and \quad (3)$$
$$f(c_j) - Kd_j \leq f_{min} - \epsilon|f_{min}|$$

In (3), c_j is the center point of the hyper-rectangle j, and d_j defines a measure for this hyper-rectangle. Jones et al. used the distance from center point c_j to its vertices for this measure. Jones also concluded that a good value for ϵ is 1.0×10^{-4}. Fig. 2 illustrates this definition.

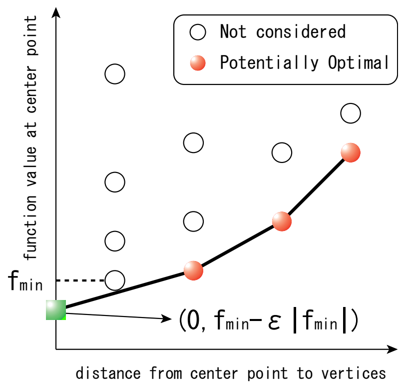


Fig. 2. Selection of hyper-rectangles to be divided

3) Division of the Hyper-rectangles: DIRECT divides the hyper-rectangles by performing division only in the dimensions with the longest side length. The sequence of the dimensions to be divided is determined by w_j by small sequence:

$$w_j = min(f(c_i + \delta_i e_j), f(c_i - \delta_i e_j)) \quad (4)$$
$$j \in I$$

Where I is the set of dimensions with the longest side length, and δ_i is one-third the length of the longest side of hyper-rectangle i. DIRECT performs the division for all dimensions in I.

The DIRECT search is performed by repeating the above operations. Several iterations of DIRECT search are shown in Fig. 3.

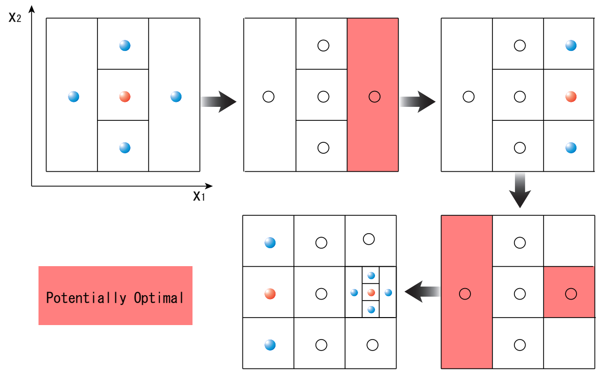


Fig. 3. Several iterations of DIRECT search

B. NSDIRECT-GA

As DIRECT mentioned in the previous section is a single objective optimization method, some modifications are necessary to apply it to multi-objective optimization problems. To improve the slow convergence of DIRECT in multi-modal functions, MOGA is also utilized. In this paper, we propose NSDIRECT-GA (Non-dominated Sorting DIRECT-GA) which combines the modified DIRECT and MOGA. Several modifications have been made in NSDIRECT-GA from the original DIRECT, including a change in the division of the hyper-cube and hyper-rectangles, determination of potentially optimal hyper-rectangles, and utilization of MOGA. NSDIRECT-GA also utilizes Non-dominated Sort and Crowding Distance used in NSGA-II [5]. The modifications are described in the following sections.

1) Changes in Division of the Hyper-cube: In DIRECT, a cube is divided according to the values of $f(c_1 \pm \delta\overrightarrow{e_i})(i = 1, ..., N)$ in each dimension at points $c_1 \pm \delta\overrightarrow{e_i}(i = 1, ..., N)$. In NSDIRECT-GA, the division is executed on the following basis:

1) Rank at the center point of the each box (Rank)
2) Crowding Distance at the center point of the each box (CD)

NSDIRECT-GA divides this space according to Rank and CD at points $c_1 \pm \delta\overrightarrow{e_i}(i = 1, ..., N)$, where δ is one-third the side length of the hyper-cube, and $\overrightarrow{e_i}$ is the ith Euclidean base-vector. That is, a hyper-cube is divided into three hyper-rectangles in each dimension.

The order in which the dimensions are divided is determined by R_i, as shown in (5), and the first division is performed in the dimension with the smallest R_i. However, when there are multiple dimensions with equal R, the sequence is determined by C_i, as shown in (6). In this case, the dimension with the longest C_i is divided first.

$$R_i = min(Rank(c_1 + \delta e_i), Rank(c_1 - \delta e_i)) \quad (5)$$
$$i = 1, ..., N$$
$$C_i = max(CD(c_1 + \delta e_i), CD(c_1 - \delta e_i)) \quad (6)$$
$$i = 1, ..., N$$

2) Change in Determination of Potentially Optimal Hyper-rectangles: In NSDIRECT-GA we divide all of the hyper-rectangles that are potentially optimal as in DIRECT. The definition of potentially optimal in NSDIRECT-GA is as follows:

Let $\epsilon > 0$ be a positive constant. A hyper-rectangle j is potentially optimal if there exists some $K > 0$ such that:

$$Rank(c_j) - Kd_j \leq Rank(c_i) - Kd_i, and$$
$$Rank(c_j) - Kd_j \leq 1 - \epsilon, and$$
$$if(d_j = d_i and Rank(c_j) = Rank(c_i))$$
$$then(CD(c_j) \geq CD(c_i)), \forall i \quad (7)$$

In (7), c_j is the center point of the hyper-rectangle j, and d_j is the distance between the center point c_j and its vertices. In this paper, we used 1.0×10^{-4} for ϵ . Fig. 4 illustrates this definition.

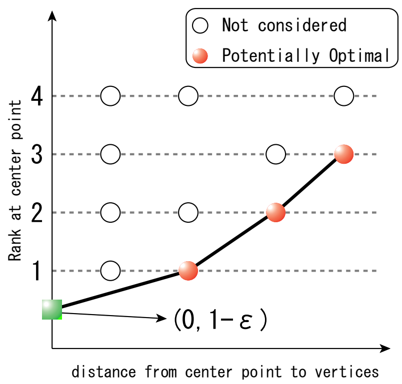


Fig. 4. Selection of hyper-rectangles to be divided

In Fig. 4, the horizontal and vertical axis represent d in (7), and $Rank(c)$, respectively. From Fig. 4, it can be seen that the potentially optimal boxes are located at the bottom of each column, forming a convex front with all other boxes located above it.

Moreover, the hyper-rectangles with $Rank = 1$ were not always potentially optimal. Therefore, the value of ϵ controls the balance of local and global search.

3) Changes in Division of the Hyper-rectangles: DIRECT divides the hyper-rectangles only in the dimensions with the longest side length in the hyper-rectangles. The sequence of dimensions to be divided is determined by R_j shown in (8), in ascending order. However, when there exist multiple dimensions with equal R, the sequence is determined by C_j as shown in (9), in descending order.

$$R_j = min(Rank(c_i + \delta_i e_j), Rank(c_i - \delta_i e_j)) \quad (8)$$
$$j \in I$$
$$C_j = max(CD(c_i + \delta_i e_j), CD(c_i - \delta_i e_j)) \quad (9)$$
$$j \in I$$

In (8) and (9) I is the set of dimensions with the longest side length, and δ_i is one-third the length of the longest side of hyper-rectangle i. NSDIRECT-GA performs the division for all dimensions in I.

4) Utilization of MOGA: In NSDIRECT-GA, MOGA is utilized after having divided the design variable space with DIRECT into the number of hyper-rectangles decided beforehand. Then, hyper-cubes are created with DIRECT according to the solutions obtained by MOGA search.

As shown in Fig. 5, the design variable space of MOGA search is divided into hyper-cubes with side length being d, where d is the side length of the smallest hyper-rectangle obtained by the DIRECT search. Only the center points of these divided hyper-cubes are used as the candidate solutions of the MOGA.

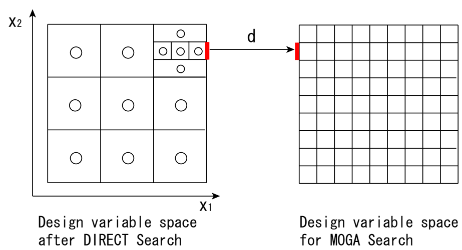


Fig. 5. Division of the design variable space for MOGA search

Non-dominated solutions are obtained from the MOGA search. These solutions are center points of the hyper-cubes with the side length of d. The hyper-cubes containing the non-dominated solutions are added to the candidate hyper-rectangles to be divided by DIRECT as shown in Fig. 6.

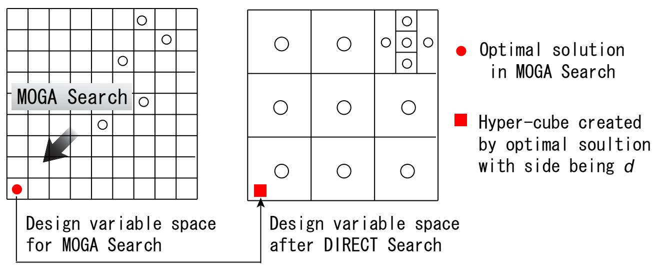


Fig. 6. Addition of optimal solution to DIRECT search

However, these hyper-cubes containing the non-dominated solutions cannot be simply added to the candidates, as this would result in the creation of shapes other than hyper-rectangles in the design variable space. To prevent this, it is necessary to make adjustments by dividing the design variable space so that it consists only of hyper-rectangles.

The following method of the adjusting division is conducted for all of the added solutions.

Step 1 Specify the hyper-rectangle in which the added solution exists.

Step 2 End if the hyper-rectangle is a hyper-cube with side length d. If not, go to Step 3.

Step 3 Divide the hyper-rectangle using the method shown in Section II-B.3. Then, return to Step 1.

The process of adjusting division is shown in Fig. 7 .

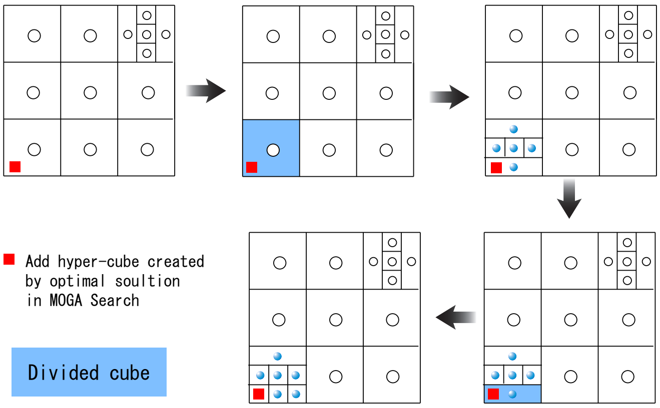


Fig. 7. Adjusting division of the design variable space

III. Numerical Experiment

A. Experimental Overview

The search history of the design variable space by the proposed NSDIRECT-GA was compared with that of NSGA-II [5].

DIRECT is known to be effective for unimodal problems [8], and so it was expected that the proposed method utilizing DIRECT would also be effective for unimodal problems. Although the convergence of DIRECT is slow when applied to multimodal problems, the proposed method was expected to resolve this issue by combining MOGA with DIRECT. To verify the effectiveness of the proposed method, it was applied to both unimodal and multimodal test problems in the experiment.

To confirm what type of search was performed by each method in the design variable space, the search historys were compared with the actual landscape of the test problem. The MOGA utilized in NSDIRECT-GA was NSGA-II [5].

B. Test Problem

The test problems used in the experiment were ZDT2 and ZDT4. The landscapes of the test problems were expressed by the function $g(x)$, which represents the distance from the Pareto optimal front. Areas in the design variable space with small values of $g(x)$ are closer to the Pareto optimal front. Fig. 8 shows the relations of function $g(x)$ and the Pareto optimal front. As shown in Fig. 8, the Pareto optimal front has a $g(x)$ value of 0; therefore, areas in the design variable space with $g(x)$ of 0 are where the Pareto optimal solutions are found.

The test problems used are described in the following sections.

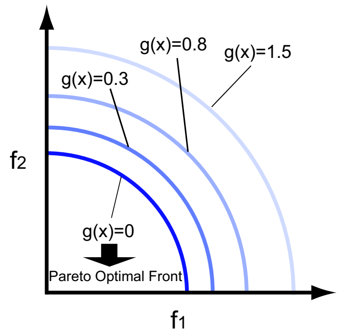


Fig. 8. Relation of function $g(x)$ and Pareto optimal front

1) ZDT2: ZDT2 is a 2-objective minimization problem where both single modal at $f_1(x)$ and $f_2(x)$ are unmoral. The Pareto optimal front is non-convex and the equation of ZDT2 is shown in (10). Its landscape is also shown in Fig. 9.

$$\begin{cases} \min \quad f_1(x) = x_1 \\ \min \quad f_2(x) = g(x_2, \cdots, x_n) \cdot h(f_1, g) \\ \quad g(x_2, \cdots, x_n) = 1 + 9(\sum_{i=2}^{n} x_i)/(n-1) \\ \quad h(f_1, g) = 1 - (f_1/g)^2 \\ \text{subject to} \quad x_i \in [0,1], \quad i = 1, \ldots, n, \quad n = 2 \end{cases} \tag{10}$$

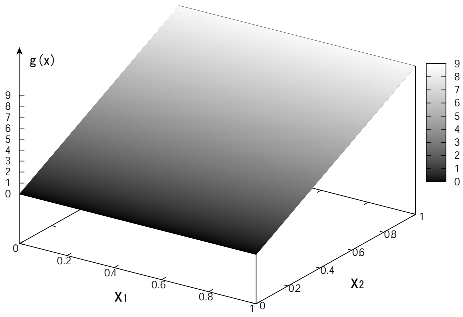


Fig. 9. Landscape of ZDT2

2) ZDT4: ZDT4 is a 2-objective minimization problem with unmodal $f_1(x)$ and multi-modal $f_2(x)$. The Pareto optimal front is convex and the equation of ZDT4 is shown in (11). In the experiment, the design variable space is shifted using the variable d because the Pareto optimal front of ZDT4 exists in the center of the design variable space and is obtained by the DIRECT algorithm in its initialization process. The value of d in this experiment was 0.5. The landscape of ZDT4 is shown in Fig. 10.

$$\begin{cases} \min \quad f_1(x) = x_1 \\ \min \quad f_2(x) = g(x_2, \cdots, x_n) \cdot h(f_1, g) \\ \quad g(x_2, \cdots, x_n) = 1 + 10(n-1) \\ \qquad\qquad + (\sum_{i=2}^{n}((x_i - d)^2 - 10cos(4\pi(x_i - d))) \\ \quad h(f_1, g) = 1 - \sqrt{f_1/g} \\ \text{subject to} \quad x_1 \in [0,1] \\ \qquad\qquad x_i \in [-5,5], \quad i = 2, \ldots, n, \quad n = 2 \end{cases} \tag{11}$$

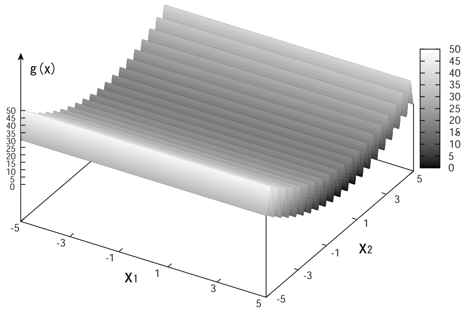


Fig. 10. Landscape of ZDT4

C. Evaluation Method

Many methods are available to evaluate the obtained Pareto solutions; the Generational Distance (GD) was used here. GD measures the distance between the obtained Pareto solutions and the Pareto optimal front. Pareto solutions with small GD value are closer to the Pareto optimal front. GD was calculated by (12).

$$GD = \left(\frac{1}{n_{PF}} \sum_{i=1}^{n_{PF}} d_i^2 \right)^{\frac{1}{2}} \tag{12}$$

Here, n_{PF} is the number of individuals in the obtained Pareto solutions and d_i is the Euclidean distance between the individual i of the obtained Pareto solutions and the Pareto optimal front in the objective space.

D. Parameters

The parameters of NSDIRECT-GA and of NSGA-II used in the experiment are shown in Tables I and II, respectively.

TABLE I
NSDIRECT-GA PARAMETER SETTINGS.

Number of boxes when MOGA is applied	5000
Maximum Generation	50
Population size	100
Crossover Method	2 point crossover
Crossover Rate	1.0
Gene length	10*Dimension
Mutation Rate	1.0/Gene length
Crowding tournament size	2

TABLE II
NSGA-II PARAMETER SETTINGS.

Maximum Generation	100
Population size	100
Crossover Method	2 point crossover
Crossover Rate	1.0
Gene length	10*Dimension
Mutation Rate	1.0/Gene length
Crowding tournament size	2

E. Results

The search results of ZDT2 and ZDT4 in 30 trials are shown in Figs. 11 and 12, respectively. GD values shown here are the medians of 30 trials.

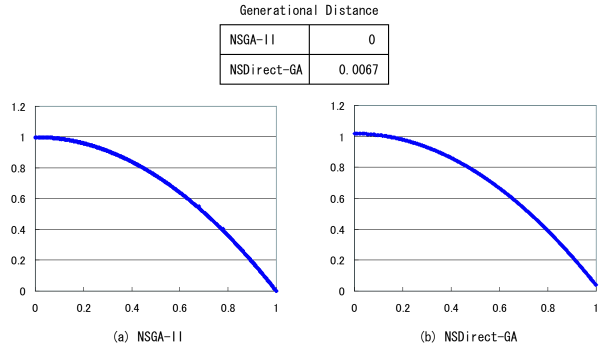


Fig. 11. Search Results and GD of ZDT2 in 30 trials

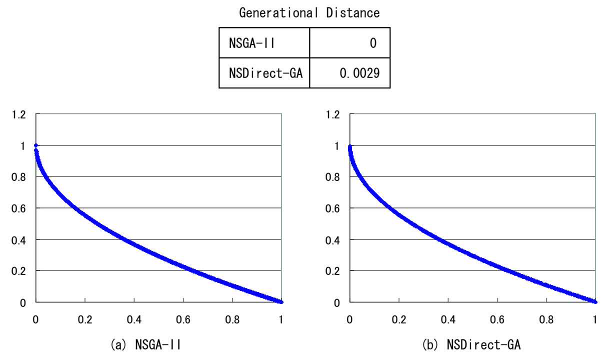


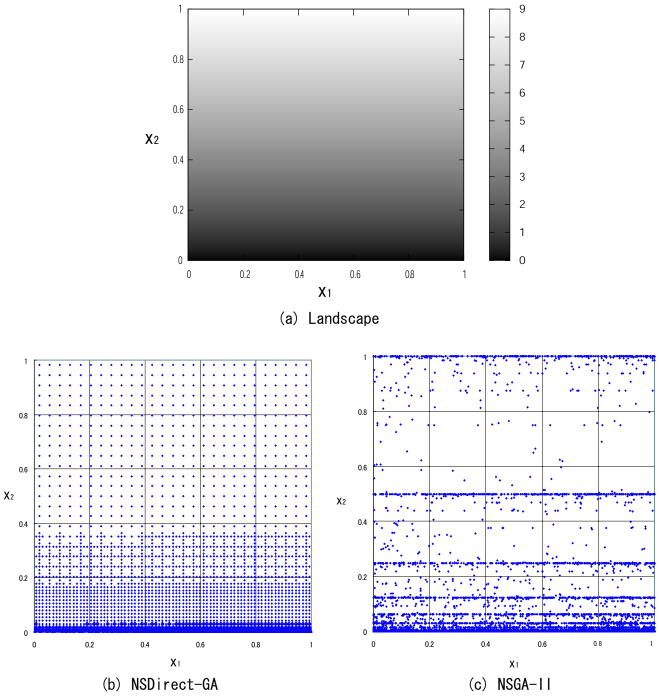
Fig. 12. Search Results and GD of ZDT4 in 30 trials

From Figs. 11 and 12, it can be seen that in both ZDT2 and ZDT4 test problem, NSGA-II and NSDIRECT-GA obtained similar solutions. The GD values showed that in both test problems the solutions obtained by NSGA-II reached the Pareto optimal front. On the other hand, although the solutions obtained by NSDIRECT-GA were not truly optimal, as shown by the GD value, they were sufficiently close to the Pareto optimal front. In NSDIRECT-GA, the obtained solutions only exist at the center of the hyper-rectangles. Therefore, solutions located on the sides of the hyper-rectangles cannot be obtained. With both test problems, the Pareto optimal solutions were located on the sides of the hyper-rectangles, and this is why NSDIRECT-GA was unable to obtain the Pareto optimal front. These results indicate that the accuracy of NSDIRECT-GA is similar to that of NSGA-II.

From Figs. 11 and 12 indicate that in both ZDT2 and ZDT4 test problems, NSDIRECT-GA obtained similar solutions to the Pareto optimal front. Therefore, NSDIRECT-GA was effective for both unimodal and multimodal problems.

The search histories of the design variable spaces of ZDT2 and ZDT4 by each method are shown in Figs. 13 and 14, respectively.

Figs. 13 and 14 show that in both ZDT2 and ZDT4 test problems, NSDIRECT-GA conducted global search in the design variable space with few unsearched areas, and searched close to the area with the Pareto optimal front. On the other hand, although NSGA-II was successful in finding the Pareto optimal front, many unsearched areas remained

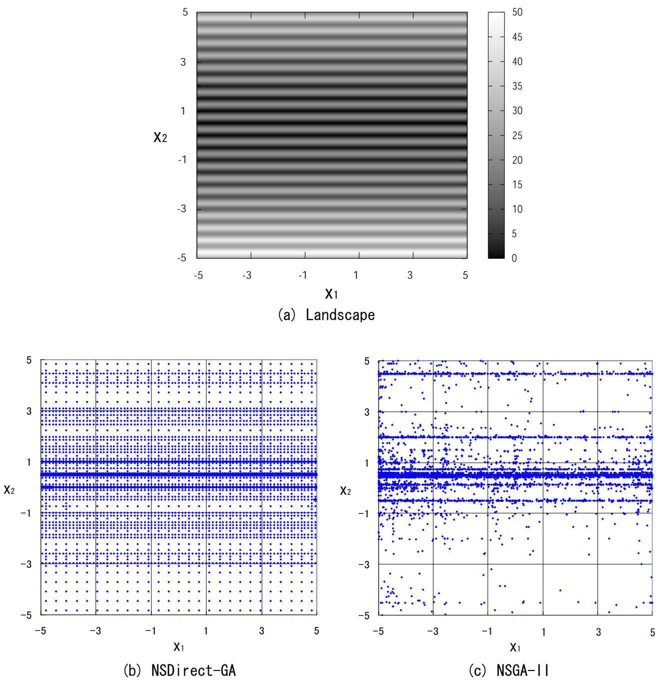


(a) Landscape

(b) NSDirect-GA

(c) NSGA-II

Fig. 13. Search histories in design variable space of ZDT2



(a) Landscape

(b) NSDirect-GA

(c) NSGA-II

Fig. 14. Search histories in design variable space of ZDT4

in the design variable space. Therefore, when solutions with similar accuracy are obtained by both methods, those obtained by NSDIRECT-GA are considered to be more reliable than those obtained by NSGA-II.

The search history of NSDIRECT-GA not only showed the area in which the Pareto optimal front was located, but also the locations of the local Pareto fronts. Comparison of the search history of NSDIRECT-GA and the test problem landscapes in Figs. 13 and 14 indicated that NSDIRECT-GA provided useful information regarding the solution space. This information regarding solution space is useful for verifying the reliability of the optimized results.

IV. Conclusions

In this paper, we proposed an optimization method called NSDIRECT-GA, which consists of a combination of DIRECT and MOGA. NSDIRECT-GA was designed to conduct a global search of the design variable space as much as possible.

The obtained solutions of the proposed NSDIRECT-GA and NSGA-II along with their search histories were compared through numerical experiments. The experimental results indicated that the accuracy of NSDIRECT-GA is similar to that of NSGA-II with less unsearched areas revealed by comparing the landscapes of the test problems, and the search history of NSDIRECT-GA was successful in providing information of the solution space.

As NSDIRECT-GA leaves less areas unsearched as compared to MOGA and provides information regarding the solution space, solutions obtained by NSDIRECT-GA are more reliable. In future research, adaptation of NSDIRECT-GA to test problems with higher dimensions will be considered.

References

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, (Addison-Wesly, Boston, 1989).

[2] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multi-objective optimization: Formulation, discussion and generalization", Proceedings of the 5th international coference on genetic algorithms, pp. 416-423, (1993).

[3] E. Zitzler and L. Thiele, "multi-objective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach", IEEE Transactions on Evolutionary Computation, Vol. 3, No. 4, pp. 257-271, (1999).

[4] M. Erickson, A. Mayer and J. Horn, "The Niched Pareto Genetic Algorithm 2 Applied to the Design of Groundwater Remediation Systems", First International Conference on Evolutionary Multi-Criterion Optimization, No. 1993, pp. 681-695, (2000).

[5] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II", KanGAL report 200001, Indian Institute of Technology, Kanpur, India, (2000).

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

[7] C.D. Perttunen Lones, D.R. and B.R. Stuckman. Lipschitzian optimization without the Lipschitz constant. Joural of Optimization Theory and Applications, Vol.79,No.1,pp.157-181,(1993).

[8] S.Hiwa, "A Hpbrid Optimization Approach for Global Exploration", Master's Thesis, Graduate School of Engineering, Doshisha University, 2007.