myjournal manuscript No.
(will be inserted by the editor)

Multi-Objective Genetic Programming with Partial Sampling and Its Extension to Many-Objective

Makoto Ohki

Received: date / Accepted: date

Abstract This paper describes a technique on an optimization of tree-structure data by of multi-objective evolutionary algorithm(MOEA), or multi-objective genetic programming (MOGP). GP induces bloat of the tree structure as one of the major problem. The cause of bloat is that the tree structure obtained by the crossover operator grows bigger and bigger but its evaluation does not improve. To avoid the risk of bloat, a partial sampling(PS) operator is proposed as a mating operator. The size of the tree and a structural distance(SD) are introduced into the measure of the tree-structure data as the objective functions in addition to the index of the goodness of tree structure. GP is defined as a three-objective optimization problem. SD is also applied for the ranking of parent individuals instead to the crowding distance of the conventional NSGA-II. When the index of the goodness of tree-structure data is two or more, the number of objective functions in the above problem becomes four or more. We also propose an effective many-objective EA applicable to such the many-objective GP. We focus on NSGA-II based on Pareto partial dominance(NSGA-II-PPD). NSGA-II-PPD requires beforehand a combination list of the number of objective functions to be used for Pareto partial dominance(PPD). The contents of the combination list greatly influence the optimization result. We propose to schedule a parameter r meaning the subset size of objective functions for PPD and to eliminate individuals created by the mating having the same contents as the individual of the archive set.

Keywords Many-Objective Genetic Programming · Partial Sampling · Tree Structural Distance · Pareto Partial Dominance · Subset Size Scheduling · Elimination of Duplicates

Tottori University
4, 101 Koyama-Minami, Tottori, Tottori, 6800945 Japan
Tel.: +81-857-5688
Fax: +81-857-0880
E-mail: mohki@tottori-u.ac.jp
ORCID ID: 0000-0002-8682-6238

1 Introduction

A technique of genetic programming (GP) [17,18] is an algorithm to optimize structured data based on a evolutionary algorithm (EA) [11,25]. GP is applied to various fields such as program synthesis[5], function generations[14] and rule set discoveries[30]. Although GP is very effective for optimizing structured data, it has several problems such as getting into a bloat, inadequate optimization of constant nodes, being easily captured in local optimal solution area when applied to complicated problems. The main cause of the bloat is a crossover operator which exchanges partial trees of parent individuals[27,2,3,7], where this paper focuses on the optimization of tree-structure data by means of GP. Several techniques to reduce the bloat have been proposed by improving the simple crossover operation[18,6,26,13,19,10]. Although these methods have successfully inhibited bloat to a certain extent, effective search has not necessarily been performed. Moreover, there is no theoretical basis that crossover is effective for optimizing the tree-structure data.

Apart from reduction of the bloat, a search method for optimizing the graph structure has been proposed[15]. Although this method is suitable for searching various size of the structured data consisting of nodes and branches, the algorithm is complicated and the computation cost is very high. In this paper, we exclude the crossover operator which is the cause of the bloat in GP, and propose a partial sampling (PS) operator [29] as a new mating operator. In PS operator, first of all, a partial sample of a partial tree structure is extracted from several individuals of a parent individual group by a procedure of a proliferation. Next, the partial tree structure obtained by the proliferation is combined with a new tree structure by a metastasis. In this paper, two types of metastasis are prepared for GP, one that depends on the original upper node and the other that does not. Repeating the proliferation and the metastasis regenerates a new tree-structure data for the next generation.

Moreover, a multi-objective EA (MOEA) technique for suppressing the bloat problem and acquire many kinds of various tree-structure data is applied for GP by adding two more objective functions. One of the newly added objective functions is the size of the tree-structure data. Furthermore, the relative position of the target individual in the population in terms of the structural distance (SD) is also evaluated as an objective function. Then, the optimization of the tree-structure data is formulated as a multi-objective optimization problem (MOP) based on these three objective functions. NSGA-II [9,8] is applied to this MOP. In the conventional NSGA-II, a crowding distance (CD) is applied for ranking the front set overflowing from the parent group. Because the conventional NSGA-II sorts such the individuals of the overflowing front set with CD and selects extreme solution, diversity about tree structure is not maintained. In this paper, SD is applied, instead of CD, for ranking the overflowing front set from the parent group.

The proposed technique and the conventional techniques are applied to a double spiral problem [6,38] for verification. This problem is a classification problem containing two classes of point sets arranged on a spiral shape to be classified with a function. This problem is well known as one of difficult problem to solve with a neural network.

The number of the index of the goodness of the tree-structure data When the index of the goodness of tree-structure data is two or more, the number of objective functions in the above problem becomes four or more. We also propose an effective many-objective EA (MaOEA) applicable to such the many-objective GP. Many-objective optimization problems (MaOPs) are difficult to solve and is tackled by many researchers [41,39,40,9,8,4]. Although SPEA2 [41,39,40] and NSGA-II [9,8] are well known as powerful algorithm for MOPs, they do not work so effectively for MaOPs [33, 12,1]. In this paper, we handle the case of solving an MaOP by NSGA-II based algorithm.

When applying NSGA-II or SPEA2 to MaOP, as the objective number increases, most of the solutions in the solution set, or population, become a relation that is not superior or inferior to each other. This relation is called non-dominated (ND) relationship. As a result, the convergence of the obtained set of Pareto Optimal Solutions ($\mathcal{POS}$) to the optimum Pareto front remarkably decreases. Sato et al. have proposed a concept of Pareto partial dominance that makes it easier to determine the superiority/inferiority relationship between solutions by using several objective functions instead of all objective functions as an algorithm for such MaOP [34]. Since NSGA-II based on Pareto partial dominance (NSGA-II-PPD) focuses on a relatively small number of objectives, solutions are easy to decide superiority/inferiority even on MaOP, and an effective selection pressure can be expected.

SPEA2 with a shift-based density estimation (SDE) strategy [23,21,20] is also very strong algorithm to solve multi-objective optimization problems. This method requires a lot of computational cost to forcibly rank individual subsets in non-dominant relationships. Also when optimizing the tree structure by SPEA2 technique with SDE, it has been difficult to suppress bloat. Therefore, this research focuses on CD which is advantageous in terms of simplicity and less computational cost. And this paper proposes SD for the purpose of suppressing the bloat.

NSGA-II-PPD has the following three problems. The first problem is that a combination list of the number of objects to be used for Pareto partial dominance must be specified before the optimization. The second one is that an appropriate number of selected objectives according to the complexity of the problem in undecided. Moreover, the contents of the combination list greatly influence the optimization result. NSGA-II-PPD performs ND sorting using all objective functions at a specific generation cycle, and preserves parents as an archive set for the next generation. This process generates child individuals having the same contents as the already existing individual in the archive set in some cases. As a result, the same individuals increases in the first front set, which disturbs effective ranking in the front selection. This is the third problem. By consideration of these problems, this paper proposes a simple scheduling technique of partial objective set used for Pareto partial dominance and a technique of killing individuals having the same contents in preserving the archive set [28]. In order to verify the effectiveness of the proposed techniques, we examine a many-objective 0/1 knapsack problem[41].

2 Partial Sampling Operator for Mating

One of the main causes of the bloat is the crossover operator generally applied in the conventional GPs, used for regenerating a new tree-structure data. This paper proposes to exclude the crossover operator from the conventional GP and to apply PS operator for regeneration of a new tree-structure data instead of the crossover operator. The PS operator creates a new tree-structure data by partially sampling tree structures from a parent individual and joining them together. This procedure is called a proliferation. The proliferation is terminated according to the probability, p_t. Partially sampled subtree structures by the proliferation are combined together by a metastasis. Two types of the metastasis are prepared, one that depends on the original upper node and the other that does not. We call the the former as an upper node depend metastasis and the latter as a random metastasis.

In the initial proliferation, a root node, $n_{i,\mathrm{root}}$, of an individual, indiv$_i$, randomly selected from a parent group $\mathbf{P}^g$ is copied to a set $\mathbf{T}_{\mathrm{sub}}$ as shown in Fig.1. The initial proliferation is started from the root node, $n_{i,\mathrm{root}}$, of the individual, indiv$_i$. In this example, the starting root node contains an identification, A. Let $\mathbf{N}_{\mathrm{cand}}$ be a set of all lower nodes under the node of $\mathbf{T}_{\mathrm{sub}}$, where that node is not selected as a node of $\mathbf{T}_{\mathrm{sub}}$ yet. One node is randomly selected from $\mathbf{N}_{\mathrm{cand}}$ and copied to $\mathbf{T}_{\mathrm{sub}}$. The proliferation terminates according to the proliferation terminate probability, p_{t}, or when $\mathbf{N}_{\mathrm{cand}} = \emptyset$. When the proliferation is terminated, the set $\mathbf{T}_{\mathrm{sub}}$ thus obtained is copied to $\mathbf{T}_{\mathrm{new}}$, where is a set of nodes as a new tree-structure data. The set $\mathbf{T}_{\mathrm{sub}}$ is initialized to $\emptyset$. Furthermore, the root node of the partial tree structure $\mathbf{T}_{\mathrm{new}}$ in the initial proliferation is randomly generated in a low probability on the initial proliferation.

In the conventional GP with variable structure length, small partial structures are assembled by an regenerating operator, for example, crossover or mutation, and these partial structures are combined to generate a new tree-structure data of a large size[31,32]. When the conventional GP increases the average size of the tree-structure data, the size of the partial structure also preserved for the next generation increases. Therefor, scheduling the probability, p_{t}, as follows prevents the size of the partial tree structure from explodingly increasing.

$$p_{\mathrm{t}}^0 = \frac{1}{\mathrm{AverageSize}(\mathbf{R}^g)}, \tag{1}$$

$$p_{\mathrm{t}}^{g+1} = \frac{\mathrm{Succ}(\mathbf{R}^g) - p_{\mathrm{t}}^0 \cdot \mathrm{Succ}(\mathbf{P}^g)}{\mathrm{Succ}(\mathbf{P}^g) - p_{\mathrm{t}}^0 \cdot \mathrm{Succ}(\mathbf{R}^g)} \left(p_{\mathrm{t}}^g - p_{\mathrm{t}}^0\right) + p_{\mathrm{t}}^0, \tag{2}$$

where $\mathbf{R}^g$ denotes the population at the g-th generation, $\mathbf{P}^g \subset \mathbf{R}^g$ denotes the parent set at the g-th generation, AverageSize($\mathbf{R}^g$) denotes a function returning the average size of each tree structure of the population, and Succ($\cdot$) denotes a function returning the average size of the partial tree structure that the argument set takes over from the previous generation.

A partial tree structure is grown by applying one of two kinds of metastasis to the partial tree structure obtained by the initial proliferation. One of two kinds of metastasis is a random metastasis. The random metastasis activates according to a metastasis selection probability, p_{met}. The other one metastasizes depending on the upper node. The upper node depend metastasis activates according to the probability, $1 - p_{\mathrm{met}}$. The partial tree structure $\mathbf{T}_{\mathrm{new}}$ shown in the Fig.1 has three empty branched numbered as 1, 2 and 3. The branch 1 has the upper node A, and the branches 2 and 3 have upper node D. Now, suppose that the branch 1 is selected as a target of the upper node depend metastasis. In the next proliferation, a node having the upper node A is selected from the parent group, $\mathbf{P}^g$. On the other hand, if the random metastasis is applied to the partial tree structure $\mathbf{T}_{\mathrm{new}}$, the beginning node for the next proliferation is randomly selected from the parent group, $\mathbf{P}^g$.

A new node is selected from the parent group, $\mathbf{P}^g$, according to the decided metastasis type. This node is not necessarily a root node. The proliferation is started from the selected node again.

By repeating the proliferation and the metastasis, new tree-structure data is generated as shown in Fig.2. However, when the metastasis applied to only one parent individual, or when a parent individual having the same structure as the generated tree structure, the generated tree structure is eliminated and PS operator is performed again. The terminal nodes are given as a random number in a low probability, where this is based on the conventional mutation idea.

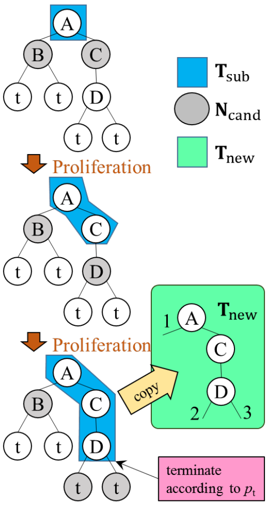

Fig. 1 The initial proliferation in PS operator.

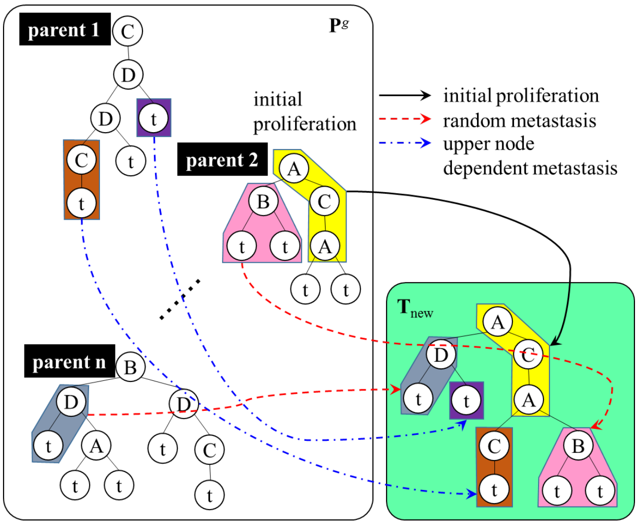

Fig. 2 Outline of how a new tree structure is created by PS operator.

3 Multi-Objective GP with Structural Distance

Optimizing the tree-structure data based only on the index of its own goodness brings problems that causes the bloat but also that the optimization is caught in a local optimum region. Depending on the structure of the local optimum region, the optimization stagnates, causing an illusion as if the obtained solution(s) were ultimate optimal. To avoid the risk of such the problems, this paper, therefor, proposes a technique to optimize the tree-structure data based on the size of the tree structure and SD in the population in addition to the index of the goodness of tree structure.

In this paper, three objective functions, h_1, h_2 and h_3, are defined as follows to be used for the multi-objective optimization . An objective function according to the goodness of an individual, indiv$_i$, is described by the following equation.

$$h_1(\text{indiv}_i) = \text{performance}(\text{root}_i), \tag{3}$$

where root$_i$ denotes a root node of the individual, indiv$_i$, and performance(root$_i$) denotes a function that returns value of the goodness of the tree structure beginning from the root node, root$_i$.

An objective function according to the size of tree structure is defined by the following equation.

$$h_2(\text{indiv}_i) = \frac{1}{\text{Size}(\text{root}_i)}, \tag{4}$$

where Size(root$_i$) denotes a function that returns the number of the nodes of the tree structure beginning from the root node, root$_i$.

An objective function according to average of SD in the population is defined by the following equation.

$$h_3(\text{indiv}_i) = \frac{1}{N_\text{pop}} \sum_{k=1}^{N_\text{pop}} \text{SD}(\text{root}_i, \text{root}_k), \tag{5}$$

where N_pop denotes the size of the population, and SD(root$_i$, root$_k$) denotes a function that returns SD between indiv$_i$ and indiv$_k$. In order to calculate SD, weights are given to all the nodes of the tree structured data by means of the following steps, when the tree structured data is initially generated. An example of giving weights to the tree structure is shown in Fig.3.

(Step 1) Give weight 1 to the root node.

(Step 2) Assume that W is a weight given to the current node.

(Step 3) Equally distribute weights to the lower nodes of the current node so that the total is $W/2$.

Two tree structures are compared in order from the root node to check conformity of both nodes as shown in Fig.3. The distance, SD(root$_i$, root$_k$), is initialized as zero. When different nodes are found in the conformity comparison, the weight of that node is added to the distance. The lower nodes below the different node are all ignored. Especially, when the tree structures of both are completely different, Distance(root$_i$, root$_k$) is given 1 as the maximum value.

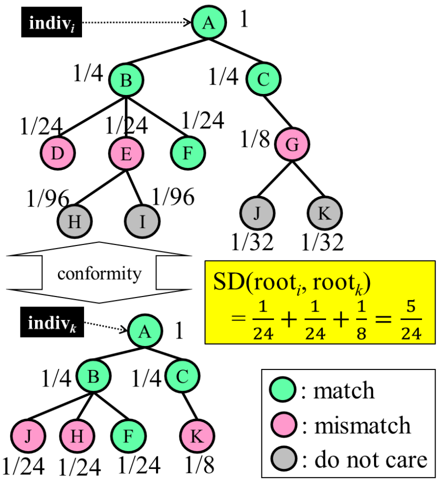

Fig. 3 An example of giving weights to the tree structure and an example of computation of SD.

Now, we have defined the three-objective optimization problem. NSGA-II shown in Fig.4 is applied to solve this problem. When several objective functions mean goodness of the tree structure and should not be joined together, the number of them and the two objective functions indicated above, h_2 and h_3, are the total number of objective functions in the proposed method in this paper. NSGA-II selects parent individuals by using non-dominated sorting and ranking with CD. Since tree-structure data is to be optimized in this paper, CD based only on the value of the objective function does not necessarily bring the diversity of the tree structure. Therefore, this paper propose to use SD when selecting parents from the rank set overflowing from the parent group. A

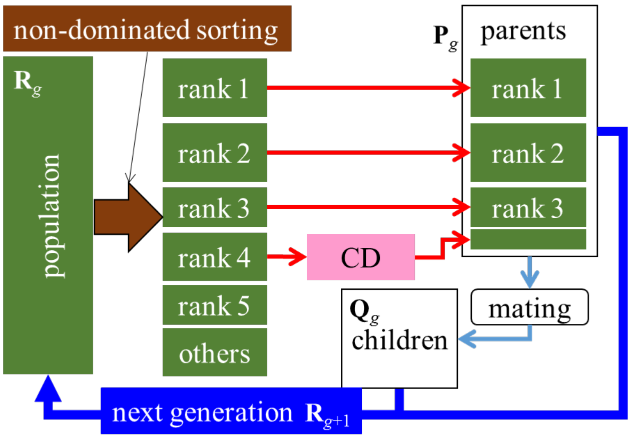

Fig. 4 Conventional NSGA-II with CD.

block chart of the modified NSGA-II with SD is shown in Fig.5.

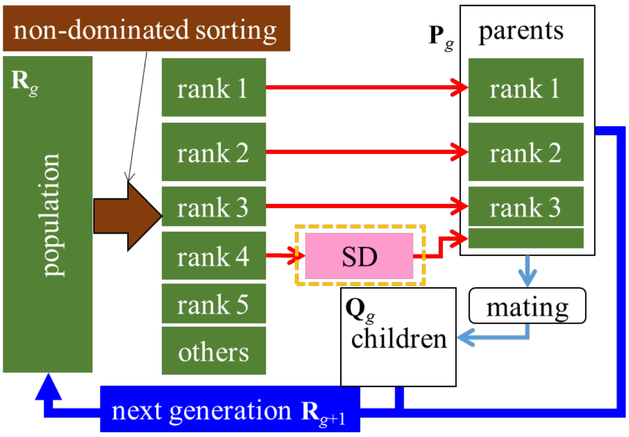

Fig. 5 Modified NSGA-II with SD.

4 MANY-OBJECTIVE EVOLUTIONARY ALGORITHM FOR MAOGP

MOP is a problem that optimizes, or maximizes in this paper, multiple objective functions under several constraints. Since the objective functions are in a trade-off relationship with each other, it is not possible, in general, to obtain the only one solution that completely satisfies all the objective functions. Therefore, we require to obtain $\mathcal{POS}$ of compromised solutions without superiority or inferiority to each other. For the objective function vector **f** consisting m objective functions, f_i, the problem of finding the variable vector **x** that maximizes the value of f_i in the feasible region S in the solution space is defined as follows.

$$\begin{cases} \text{max.} & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})]^T \\ \text{s.t.} & \mathbf{x} \in \mathbf{S} \end{cases} \tag{6}$$

When the values of the objective function, f_i, of two solutions x and y satisfy the following relation, we say that the solution x dominates the solution y.

$$\mathbf{f}(\mathbf{x}) \succ \mathbf{f}(\mathbf{y}) \triangleq \\ \forall i \in \mathbf{M} : f_i(\mathbf{x}) \geqq f_i(\mathbf{y}) \wedge \exists i \in \mathbf{M} : f_i(\mathbf{y}) > f_i(\mathbf{y}) \tag{7}$$

where **M** denotes a set of the indexes for the objective function, $\{1, 2, \ldots, m\}$. When there is no solution dominates a solution **x**, the solution **x** is called non-inferior solution. A set of such the non-inferior solutions is defined as the following $\mathcal{POS}$.

$$\mathcal{POS} = \{\mathbf{x} \in \mathbf{S} | \neg \exists \mathbf{y} \in \mathbf{S}.\mathbf{f}(\mathbf{y}) \succ \mathbf{f}(\mathbf{x})\} \tag{8}$$

A Pareto front showing the the trade-off relation between the objective functions is defined as follows.

$$\mathcal{FRONT} = \{\mathbf{f}(\mathbf{x}) | \mathbf{x} \in \mathcal{POS}\} \tag{9}$$

Several effective studies [41,39,40,9,8,4] have been made on MOP as defined by Eq.(6). NSGA-II shown in Fig.4 is a powerful multi-objective optimization scheme as a method proposed on one of these studies. NSGA-II applies non-dominated sorting (ND sorting) to the population **Q**, and the individuals are classified to several ranked subsets, $\mathbf{F}_1$, $\mathbf{F}_2$, $\mathbf{F}_3$, $\cdots$. While not exceeding the size of the parent set **P**, the individuals of each subset are moved to the parent set in order. Individuals of the subset that exceeds the size of the parent set is sorted using crowding distance (CD sorting) and moved to the parent set. The individuals not selected are culled. The mating operators generates the child set **C** from the parent set **P** by using the crossover and mutation operators.

Although NSGA-II effectively solves MOP with less than four objective functions, as the objective number m increases, an appropriate $\mathcal{POS}$ could not be obtained even by those methods containing the conventional NSGA-II. When ND is performed based on the conventional Pareto dominance using all m objective functions, as the number of objective function increases, a subset of solutions satisfying Eq.(7) is difficult to obtain [37]. Then most solutions of the population become non-inferior solutions. As a result, the superiority/inferiority relationship between solutions is difficult to determined, and the selection pressure in the optimization is significantly reduced. This paper focuses to NSGA-II with Pareto partial dominance shown in Fig.6 for solving MaOP. Pareto partial dominance is based on a concept of partially applying Pareto domination to r objective functions extracted from all m objective functions. The Pareto partial dominance is defined by the following formula.

$$\mathbf{f}(\mathbf{x}) \sqsupset \mathbf{f}(\mathbf{y}) \triangleq \\ \forall i \in \mathbf{R} \subset \mathbf{M} : f_i(\mathbf{x}) \geqq f_i(\mathbf{y}) \wedge \exists i \in \mathbf{R} \subset \mathbf{M} : f_i(\mathbf{y}) > f_i(\mathbf{y}) \tag{10}$$

where **R** denotes a set of r indexes selected from **M**. Since conditions satisfying Pareto partial dominance are relaxed as compared with the conventional dominance using all m objective functions, the population is easier to rank finely in MaOP with large m.

In NSGA-II-PPD, first of all, given r, the number of objective functions to be considered in the partial ND sorting, a combination list of ${}_mC_r$ selections is prepared beforehand. For each I_g generations, the combination of the objective functions to be considered for Pareto partial dominance is changed, and $\mathbf{R}_{g+1}$ is selected with performing ND sorting on $\mathbf{P}_g + \mathbf{C}_g + \mathbf{A}$ using all m objective functions and copied to the archive set **A**, where + denotes the direct sum.

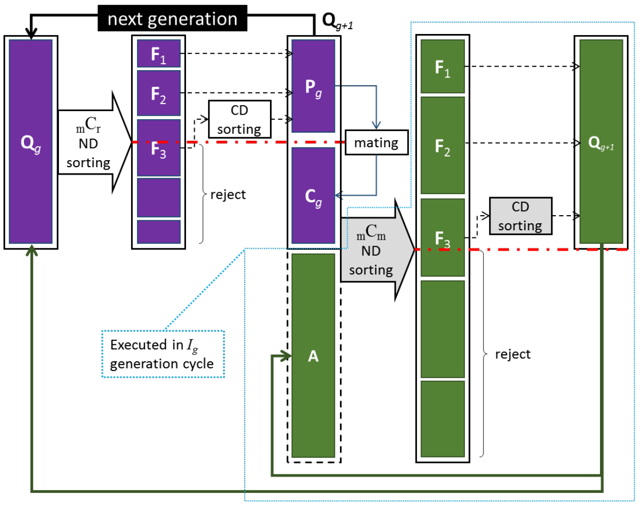

Fig. 6 NSGA-II with Pareto partial dominance.

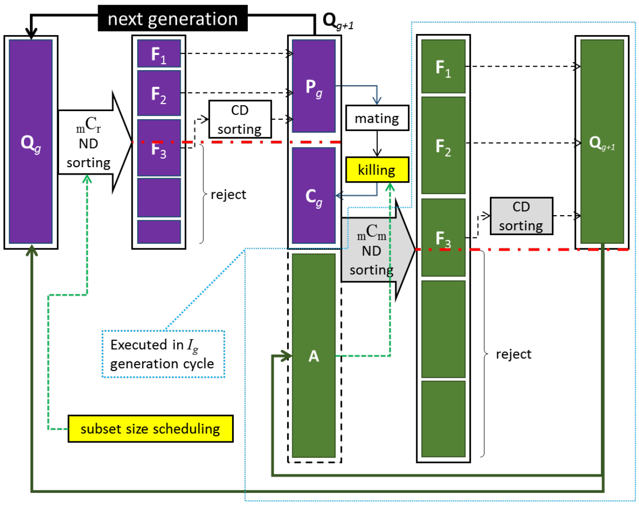

Fig. 7 Improved NSGA-II with Pareto partial dominance.

5 IMPROVEMENT OF NSGA-II BASED ON PARETO PARTIAL DOMINANCE

NSGA-II-PPD has the following three problems. The first problem is that the subset size of the objective functions to be used for Pareto partial dominance is required to beforehand specify before the optimization in a form of a list, or the combination list. The second one is that an appropriate value of the subset size according to the complexity of the problem is unknown. The contents of the combination list greatly influence the optimization result. On the other hand, the creation of the combination list is a very troublesome and difficult task for the user. NSGA-II-PPD performs ND sorting using all objective functions at a specific generation cycle, and preserves parents as an archive set for the next generation. This process generates child individuals having the same contents as the already existing individual in the archive set in some cases. As a result, the same individuals increases in the first front set, which disturbs effective ranking in the front selection. This is the third problem. In order to avoid these problems, this paper proposes two improvements. A block chart of the improved NSGA-II-PPD is shown in Fig.7.

As the first improvement, a subset size scheduling (SSS) is proposed for NSGA-II-PPD. NSGA-II-PPD treated in this paper does not use the combination list for each I_g generation cycle. The parameter r is given by the following equations.

$$q = \frac{g \cdot m}{G} + \mathrm{rand\_int}(2B+1) - B, \tag{11}$$

$$r = \begin{cases} B, \ q < B \\ q, \ \ B \leqq q < m \\ m, \ q \geqq m \end{cases} \tag{12}$$

where m denotes the number of the objective functions, rand_int$(\cdot)$ denotes a function returns a random integer less than the argument, B denotes an integer parameter larger than 1 and less than $m/2$, and G denotes the end generation. Fig.8 shows the possible value of the selection number, r.

In NSGA-II-PPD, several individuals having the same contents as an individual already existing in the children, $\mathbf{C}_t$, or the archive set, $\mathbf{A}$, are generated and stored by the mating. If the optimization proceeds while sustaining such the individuals having relatively good evaluation, duplicates of them increases within the population. If the problem to be optimized is relatively simple, individuals with the same content are frequently generated during the optimization. The second improvement is elimination of such the individuals having the same contents of an individual already existing in the children, $\mathbf{C}_g$, and the archive set, $\mathbf{A}$, after the

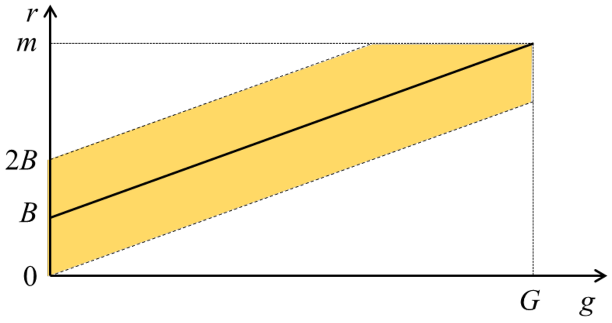

Fig. 8 The selection number, r, probablistically takes a value on the colored range according to the generation g, where rand_int$(\cdot)$ denotes a function returns a random integer less than the argument, B denotes an integer parameter larger than 1 and less than $m/2$, and G denotes the final generation.

mating. In other words, the duplicates created by the mating is eliminated, we call this elimination of duplicates (EoD). Since the optimization problem treated in this paper is the maximizing problem, by setting the value of all objective functions of such the individual to 0, the individual are eliminated. The same content individual become the worst individual. After EoD, the mating does not reproduce the individual.

6 Verification of the Proposed Techniques

6.1 Double Spiral Problem

A double spiral problem is applied to verify an effectiveness of the proposed techniques. The double spiral problem is a problem of classifying two data sets arranged in a spiral shape, and it is known as a problem that is difficult to solve even using neural networks[6,38]. These two data sets are arranged as shown in Fig.9 and are to be classified by the following function f.

$$\begin{cases} f(x,y) > 0 \Longleftrightarrow (x,y) \in \mathbf{D}_1, \\ f(x,y) < 0 \Longleftrightarrow (x,y) \in \mathbf{D}_2, \\ f(x,y) = 0 \Longleftrightarrow \text{FALSE}, \end{cases} \tag{13}$$

where (x,y) denotes the coordinates of each point on the two-dimensional plane, and $\mathbf{D}_1$ and $\mathbf{D}_2$ denote the data sets expressed with the red crosses and the blue circles shown in Fig.9 respectively. In this paper, the case when $f(x,y) = 0$ is treated as classification failure at the point (x,y).

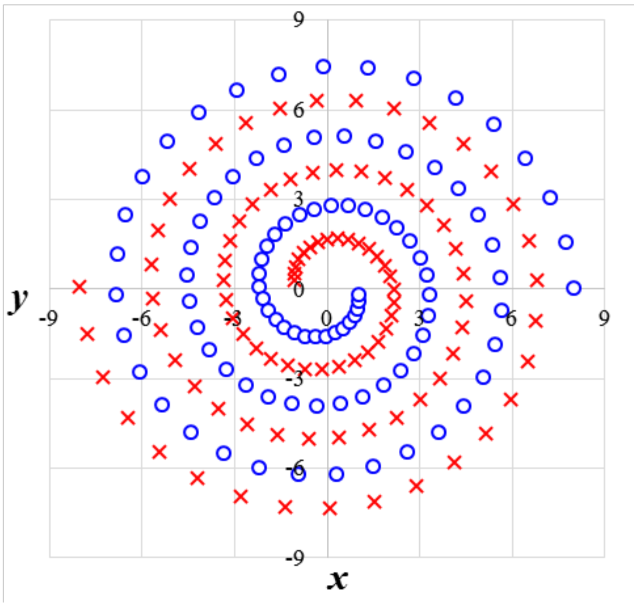

Fig. 9 Arrangement of two data sets for double spiral problem. The red cross denotes a point in the class $\mathbf{D}_1$ and the blue circle denotes a point in the class $\mathbf{D}_2$.

The following nodes are prepared as elements for constituting the classifying function f.

$$\mathcal{N}_N = \{+,-,*,\div,\sin,\cos,\tan,\text{ifltz}\} \tag{14}$$

$$\mathcal{N}_T = \{x,y,\text{constant}\} \tag{15}$$

where $\mathcal{N}_N$ denotes a set of non-terminal node, $\mathcal{N}_T$ denotes a set of terminal node, and "ifltz" denotes a function with three arguments representing a conditional branch as follows,

$$\text{ifltz}(a,b,c) \triangleq \text{"if } a < 0 \text{ then } b \text{ else } c\text{"} = \begin{cases} b \ (a < 0), \\ c \ (\text{otherwise}). \end{cases} \tag{16}$$

The objective function, h_1, according to the goodness of an individual is defined by the following equation.

$$\begin{aligned} h_1(\text{indiv}_i) &= \text{performance}(\text{root}_i), \\ &= \frac{1}{|\mathbf{D}_1 \cup \mathbf{D}_2|} \sum_{k=1}^{|\mathbf{D}_1 \cup \mathbf{D}_2|} g(x_k, y_k), \end{aligned} \tag{17}$$

$$g(x,y) = \begin{cases} 1 \ (f(x,y) > 0 \wedge (x,y) \in \mathbf{D}_1), \\ 0 \ (f(x,y) > 0 \wedge (x,y) \in \mathbf{D}_2), \\ 1 \ (f(x,y) < 0 \wedge (x,y) \in \mathbf{D}_2), \\ 0 \ (f(x,y) < 0 \wedge (x,y) \in \mathbf{D}_1), \\ 0 \ (f(x,y) = 0). \end{cases} \tag{18}$$

In this double spiral problem, the function, Size(root$_i$), applied for the objective function according to the size of tree structure is defined as the number of nodes of the tree structure.

In order to verify the effectiveness, the following four combinations are applied to the double spiral problem, combination of the conventional operators and CD (expressed as ”CO+MU & CD”), combination of the conventional operators and SD (expressed as ”CO+MU & SD”), combination of PS operator and CD (expressed as ”PS & CD”) and combination of PS operator and SD (expressed as “PS & SD”). The conventional operators denotes the conventional crossover and the conventional mutations[17,18,13,36]. The size of the population, N_{pop}, the running generations and the number of points in the double spiral problem, $|\mathbf{D}_1 \cup \mathbf{D}_2|$, are defined as 100, 1,000,000 and 190 respectively. The probability, p_{met}, for selecting the type of the metastasis is tried to 0.5, 0.25 and 0.75.

Fig.10 shows distributions on the $h_2 - h_1$ plane of the first-front set in the final generation. As shown by Fig.10, NSGA-II with combining PS operator and SD has given the best solution set, distributed in the upper right direction, in the widest range. The solutions obtained by NSGA-II with combining PS operator and CD has relatively high diversity but their evaluations are not so good. NSGA-II with combining the conventional operators and CD has given relatively good solutions but their diversity is low. NSGA-II with combining the conventional operators and SD has given the worst solution set with the lowest diversity.

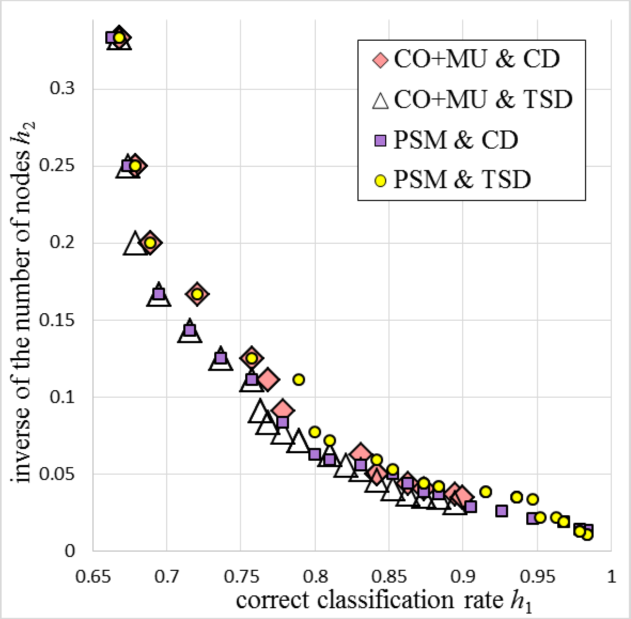

Fig. 10 Distribution on the h_2-h_1 plane of the first front set in the final generation when using each method.

Fig.11 shows a comparison of distribution on the h_2-h_1 plane of the first front set in the final generation when 3-objective and 2-objective optimizations are executed by using the PS operator with p_{met} = 0.50 and SD for the ranking. Compared to the distribution of solutions given by 2-objective optimization, the 3-objective optimization has acquired far better solutions in wider range. When PS operator with p_{met} = 0.50 and CD are combined, the same result has been obtained as shown in Fig.12. This shows an effectiveness of multi-objective optimization of the tree-structure data as proposed in this paper.

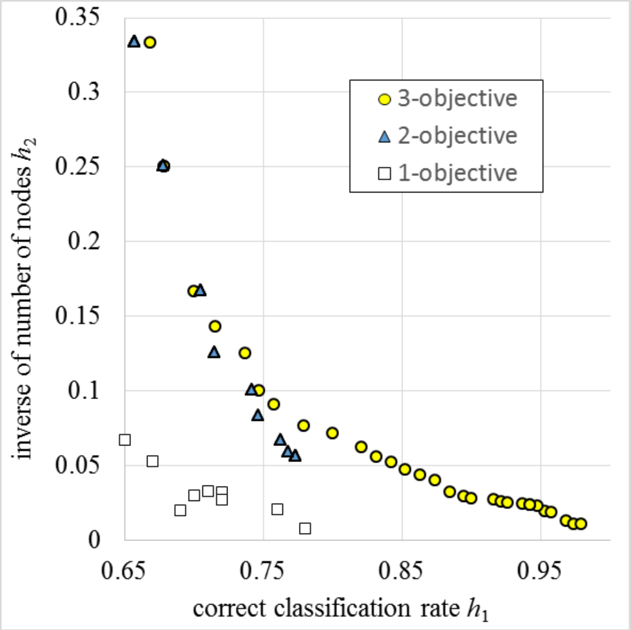

Fig. 11 Comparison of distribution on the h_2-h_1 plane of the first front set in the final generation when 3-objective and 2-objective optimizations are executed by using the PS operator with p_{met} = 0.50 and SD for the ranking.

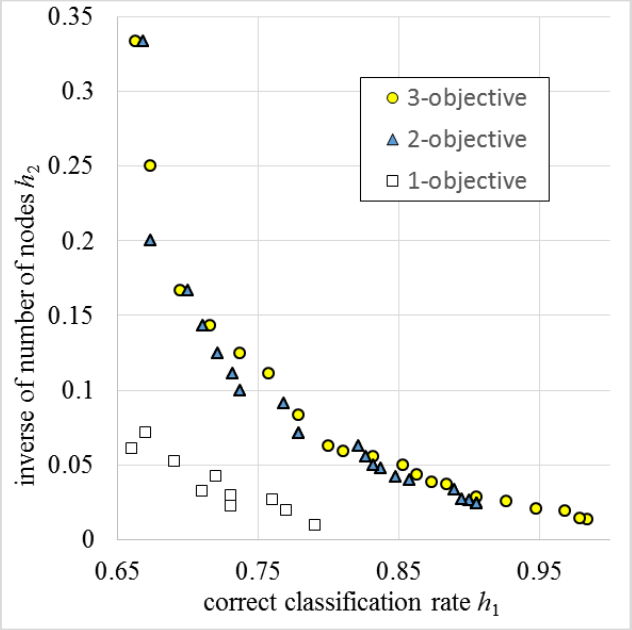

Fig. 12 Comparison of distribution on the h_2-h_1 plane of the first front set in the final generation when 3-objective and 2-objective optimizations are executed by using the PS operator with p_{met} = 0.50 and CD for the ranking.

Fig.13 shows values of Norm [35] and MS [39] given by each method. In this figure, PS$*.**$ denotes when PS operator with the metastasis selection probability, p_{met}, which is equal to $*.**$ is used for the mating. Concerning both Norm and MS values, the best result has been obtained by the method using PS operator with p_{met} = 0.50 and SD. The results using PS operator have gathered in the upper right of the figure, whereas the results using the conventional crossover and the conventional mutation have gathered in the lower left. This shows the superiority of PS operator. On the other hand, the advantage of SD has not been clearly shown by this experiment. SD have optimized relatively better only when combined with PS operator. NSGA-II even with SD has given the worst results when combined with the conventional operators. The reason for this result is considered as that SD has a low ability to preserve extreme solutions as CD does. In the case of the multi-objective optimization of the tree structure, the ability to retain the diversity of tree structures like the ranking with SD is necessary, then an improvement to add ability to preserve the extreme solutions like CD should be considered.

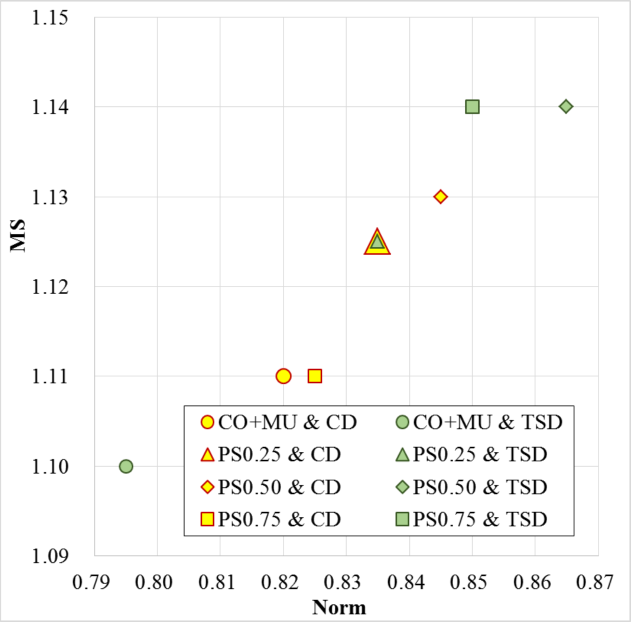

Fig. 13 Comparison of results by each method on MS-Norm plane. CO+MU denotes when the conventional crossover operator and the conventional mutation operators are used for the mating. PS$*.**$ denotes when PS operator with the metastasis selection probability, p_{met}, which is equal to $*.**$, is used for the mating.

6.2 MANY-OBJECTIVE 0/1 KNAPSACK PROBLEM

In order to verify the effectiveness of the improved technique, a many-objective 0/1 knapsack Problem (MaOKSP) [42] is performed. MaOKSP composed of m knapsacks and j items. The capacity of the i-th knapsack is c_i. The weight and the price of the j-th item are w_{ij} and p_{ij} respectively in the i-th knapsack. Let an individual $\mathbf{x} \in 0, 1^n$ be the n dimensional vector that selects the items. MaOKSP is defined by the following formula.

$$\begin{cases} \text{max.} \ \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})]^T \\ \text{s.t.} \quad \sum_{j=1}^{n} w_{ij} \cdot x_j \leqq c_i \end{cases}, \tag{19}$$

$$f_i(\mathbf{x}) = \sum_{j=1}^{n} p_{ij} \cdot x_j \quad \text{for } i = 1, 2, \cdots, m, \tag{20}$$

where the number of items n, the weight matrix w_{ij}, the profit matrix p_{ij} and the knapsack capacity vector c_i are defined as follows,

$$n = 1000, \tag{21}$$

$$w_{ij} \in (0, 1) \subset \mathbf{R}, \tag{22}$$

$$p_{ij} \in (0, 1) \subset \mathbf{R}, \tag{23}$$

$$c_i \in (0, 100) \subset \mathbf{R}. \tag{24}$$

$\mathcal{POS}$ obtained by the optimization is evaluated by using Maximum Spread (MS)[39] and Norm[35].

Norm [35] and Maximum Spread (MS)[39] are applied for evaluation of each method. Norm denotes a measure of the convergence of the population to the Pareto front $\mathcal{PF}$ and is defined by the following equation.

$$\mathrm{Norm}(\mathcal{PF}) = \frac{1}{|\mathcal{PF}|} \sum_{j=1}^{|\mathcal{PF}|} \sqrt{\sum_{i=1}^{m} f_i(\mathbf{x}_j)^2}, \tag{25}$$

where $\mathbf{x}_j$ denotes the j-th individual of the Pareto front, $\mathcal{PF}$. The larger the Norm value, the closer the approximate Pareto front, $\mathcal{PF}$. MS denotes a measure of the spread of the first front at the final generation[39] and is defined by the following equation.

$$\mathrm{MS}(\mathcal{PF}) = \sqrt{\sum_{i=1}^{m} \left(\max_{j=1}^{|\mathcal{PF}|} f_i(\mathbf{x}_j) - \min_{j=1}^{|\mathcal{PF}|} f_i(\mathbf{x}_j) \right)^2}. \tag{26}$$

The larger the MS value, the wider the spread of the population given by the optimization.

The conventional NSGA-II, NSGA-II-PPD when $r = 3$, $r = 6$ and $r = 8$, NSGA-II-PPD in the case of giving the combination list shown in Table1 and the improved technique are carried out for the verification. The optimization is performed by setting the objective number to $m = 4, 6, 8, 10$ and the iterative generations to $G = 1,000,000$.

Fig.14 shows transition of the number of individuals of the first-front according to the generation in the case that $m = 10$ and $I_g = 500$. In the figure, “NSGA-II” denotes the results by the conventional NSGA-II, “PPD(r=*)” denotes the results by NSGA-II-PPD with the constant value of $r = *$, “PPD(list)” denotes the results by NSGA-II based on Pareto partial dominace with the combination list shown in Table1, and “Improved” denotes the results by the algorithm proposed in this paper. The conventional NSGA-II and NSGA-II-PPD in $r = 8$ has given large number of the individuals of the first-front set throughout the optimization. NSGA-II-PPD with $r = 6$ has given the number next to them. At the end of the optimization, the improved technique has caught

Table 1 The combination list for NSGA-II-PPD.

	generation range			
	0 −500k	500k −900k	900k −1M	
m	r			
4	2	3	4	
6	3	5	6	

	generation rage			
	0 −300k	300k −600k	600k −900k	900k −1M
m	r			
8	3	5	7	8
10	3	6	8	10

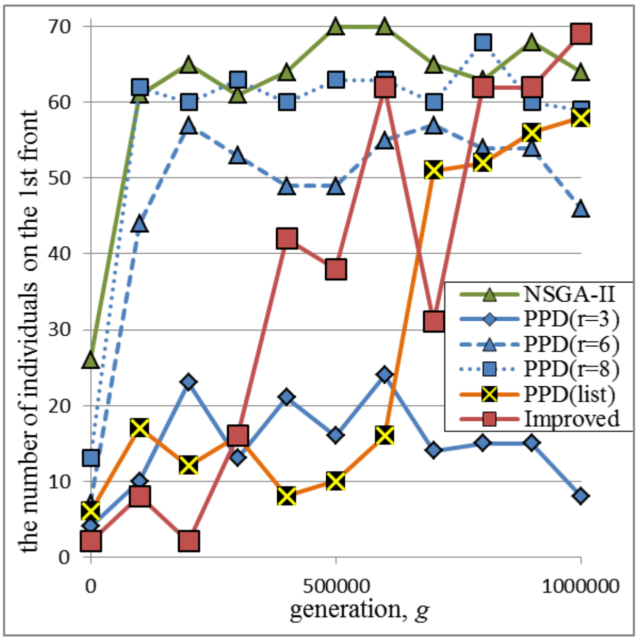

Fig. 14 Transition of the number of individuals of the first-front according to the generation.

up with these values. NSGA-II-PPD with the combination list is also similar.

Fig.15 shows Norm values after the optimization to the objective number m in the case that I_g = 500. In any technique, the convergence to $\mathcal{POS}$ increases as the number of objectives increases. Although, regarding to the convergence, NSGA-II-PPD in the case that $r = 3$, NSGA-II-PPD with the combination list and the improved technique have given almost equivalent results, the conventional NSGA-II has given relatively poor results.

Fig.16 shows MS values after the optimization to the objective number m in the case that I_g = 500. The MS value, or the diversity of $\mathcal{POS}$, given by NSGA-II-PPD in the case that $r = 3$ decreases as the objective number increases, whereas it increases with the other three techniques. In the improved technique, since r increases as the generation progresses, the superiority/inferiority relationship of solutions becomes difficult to decide by Pareto partial dominance at the end of the optimization, and many individuals belong to the first-front set. As a result, since most individuals of the parents are ranked by the CD sorting, and it is considered that diversity has increased. NSGA-II-PPD with the combination list has shown diversity equal to or less than that of the improved technique. The reason that sufficient diversity has not been obtained by NSGA-II-PPD in the case that $r = 3$ is considered as because partial dominance by using all objectives has not been performed only between 900,000-1 million generations. Regarding the diversity of solutions, the conventional NSGA-II has given the highest value.

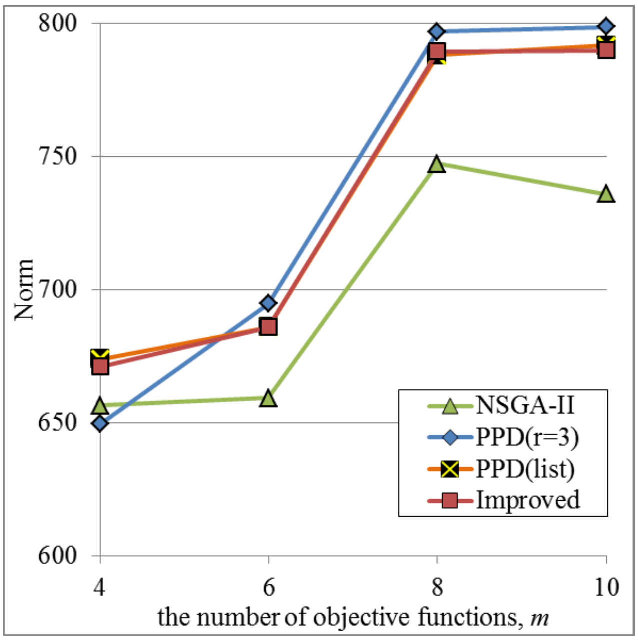

Fig. 15 Comparison of Norm values to the object number m.

Fig.17 shows Norm values to the generation g in the case that m = 10 and I_g = 500. In NSGA-II-PPD, the convergence to $\mathcal{POS}$ tends to decrease as the value of the parameter r increases. In this technique, when r approaches m, the solutions are hard to dominated by the partial dominance, so a large number of individuals are selected as the first-front set. As a result, sufficient ranking is not made in the non-dominated sorting, and the convergence has deteri-

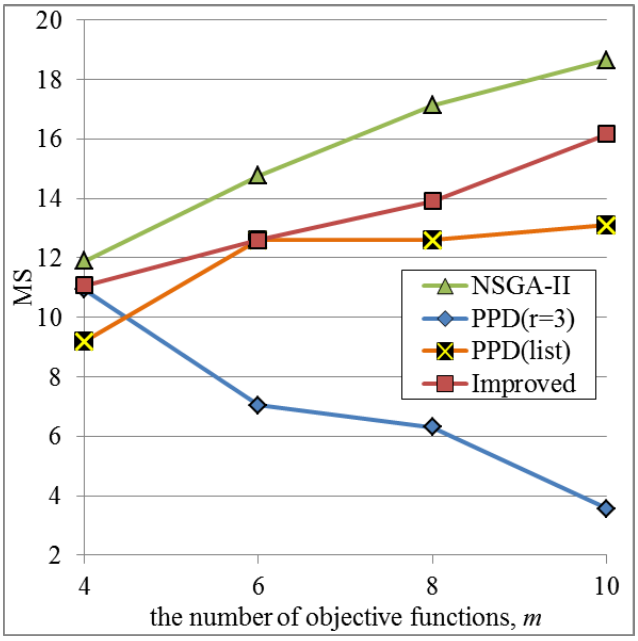

Fig. 16 Comparison of MS values to the object number m.

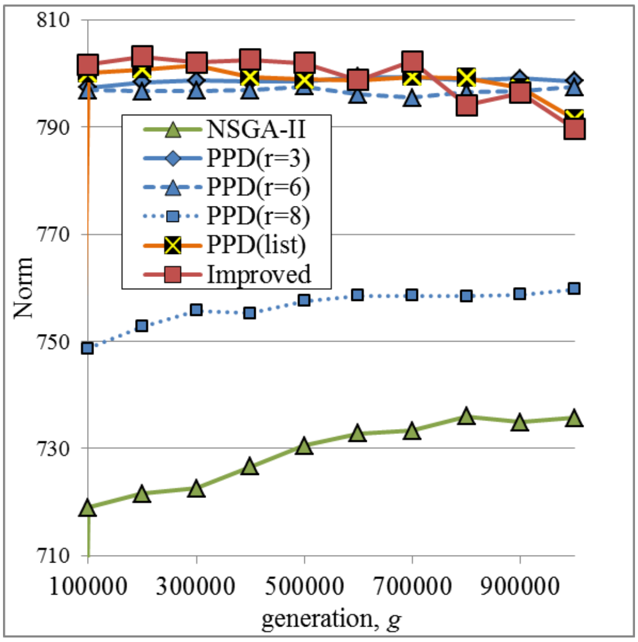

Fig. 17 Comparison of the Norm values to the generation g.

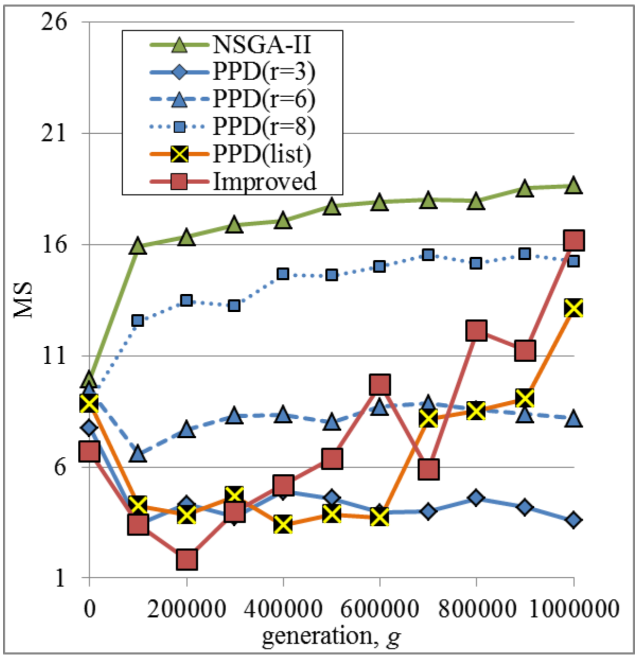

Fig. 18 Comparison of the MS values to the generation g.

orated. On the other hand, although the improved technique has shown the highest convergence at the beginning of the optimization, the convergence has declined at the final stage. In the improved technique, since the value of r increases as the generation progresses, the solutions become hard to dominated by the partial dominance. As a result, sufficient ranking is not made in the non-dominated sorting, and the convergence has deteriorated in the final stage.

Fig.18 shows MS values to the generation g in the case that $m = 10$ and $I_g = 500$. Although the diversity in the cases of the conventional NSGA-II and NSGA-II-PPD in $r = 8$, maintains a high value throughout, the convergence is low as shown in Fig.17, so it is not necessary to pay attention to them. On the other hand, the diversity is rising as the optimization progress in the case of the improved technique. Moreover, the improved technique brings relatively high convergence as shown in Fig.17, so that the superiority of the improved technique is shown overall.

7 Conclusion

In this paper, multi-objective optimization of tree-structure data, or MOGP, has been proposed, where the tree structure size and the structural distance (SD) are additionally introduced into the measure of the goodness of the tree structure as the objective functions. Furthermore, the partial sampling (PS) operator has been proposed to effectively search tree structure while avoiding the bloat. In order to verify the effectiveness of the proposed techniques, they have applied to the double spiral problem. yukari By means of the multi-objective optimization of tree-structure data, we have found that more diverse and better tree structures are acquired. The proposed method incorporating PS operator and SD in NSGA-II has given relatively good results. However, since PS opertor has low ability to numerically optimize constant nodes on the tree structure, it has not well worked effectively for the function optimization. In addition, since ranking with SD in NSGA-II has low ability to preserve extreme solutions in the objective function space, solutions not have been effectively selected.

When the index of the goodness of tree-structure data becomes two or more, the number of objective functions in MOGP becomes four or more, MaOGP. The improved NSGA-II-PPD applicable to such the MaOGP has been also proposed in this paper. In the improvement, we have proposed SSS and EoD.

The improved NSGA-II-PPD with SSS and EoD and other conventional techniques are applied to the many-objective 0/1 knapsack problem for verification of the effectiveness. The improved NSGA-II-PPD has given the higher diversity than other techniques as the number of the objective functions of the problem increases. On the other hand, the improved NSGA-II-PPD has given the convergence equal to or higher than the other techniques even when the number of the objective functions becomes large. By means of the proposed simple scheduling of the parameter r, sufficient convergence has been obtained in the early generations with the smaller r, and the diversity has been supplemented in the generations with the larger r at the end of the optimization.

In the future, a technique to incorporate numerical optimization ability such as a particle swarm optimization [16] and the mutation to PS operator and the ranking selection technique combining SD and CD should be considered in the future. The PS operator proposed in this paper has a mechanism to terminate the proliferation, but does not have no mechanism to forcibly exit from the PS operator. Such the mechanism to forcibly exit from the PS operator should be considered.

Since the improved NSGA-II-PPD still has given insufficient results in terms of the diversity, we need to improve this point while maintaining the current convergence. Although each technique has been applied to the relatively simple many-objective 0/1 knapsack problem in this paper, we need to apply to more complicated problems and verify the effectiveness. We also need to pursue the combination list and to compare the further improved NSGA-II-PPD and the conventional NSGA-II-PPD with the optimal combination list. And then, we need to propose an effective MaOGP by combining these improved techniques in the future.

In this paper, the quality indicator MS is applied to assess the diversity of the final solutions. However, MS is able to simply be affected by the convergence of the solutions, in favor of poorly-converged solutions. In this sense, MS is not necessarily effective for the assessment of the diversity. In the future research, it is necessary to consider techniques such as a diversity comparison indicator (DCI) [22] to assess the diversity of the solutions. On the other hand, the convergence of the solutions has been evaluated only using Norm. In this regard, the future research needs to visualize solutions in multi-objective optimization with parallel coordinates [24], which can partially reflect the convergence, spread and uniformity.

Acknowledgements This research work has been supported by JSPS KAKENHI Grant Number JP17K00339.

The author would like to thank to her families, the late Miss Blackin', Miss Blanc, Miss Caramel, Mr. Civita, Miss Marron, Miss Markin', Mr. Yukichi and Mr. Ojarumaru, for bringing her daily healing and good research environment.

Conflict of Interest

The author declares that she has no conflict of interest.

References

1. Aguirre, H.E., Tanaka, K.: Working principles, behavior, and performance of moeas on mnk-landscapes. European Journal of Operational Research **181**(3), 1670–1690 (2007)
2. Angeline, P.J.: Subtree crossover: Building block engine or macromutation. Genetic programming **97**, 9–17 (1997)
3. Angeline, P.J.: Subtree crossover causes bloat. In: Genetic Programming 1998: Proc. 3rd Annual Conference. Morgan Kaufmann (1998)
4. Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A., et al.: Evolutionary algorithms for solving multi-objective problems, vol. 5. Springer (2007)
5. David, C., Kroening, D.: Program synthesis: challenges and opportunities. Phil. Trans. R. Soc. A **375**(2104), 20150403 (2017)
6. De Bonet, J.S., Isbell Jr, C.L., Viola, P.A.: Mimic: Finding optima by estimating probability densities. In: Advances in neural information processing systems, pp. 424–430 (1997)
7. De Jong, E.D., Watson, R.A., Pollack, J.B.: Reducing bloat and promoting diversity using multi-objective methods. In: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, pp. 11–18. Morgan Kaufmann Publishers Inc. (2001)
8. Deb, K.: Multi-objective optimization using evolutionary algorithms, vol. 16. John Wiley & Sons (2001)
9. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In: International Conference on Parallel Problem Solving From Nature, pp. 849–858. Springer (2000)
10. Francone, F.D., Conrads, M., Banzhaf, W., Nordin, P.: Homologous crossover in genetic programming. In: Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2, pp. 1021–1026. Morgan Kaufmann Publishers Inc. (1999)
11. Goldberg, D.: Genetic algorithms in optimization, search and machine learning. Reading: Addison-Wesley (1989)
12. Hughes, E.J.: Evolutionary many-objective optimisation: many once or one many? In: Evolutionary Computation, 2005. The 2005 IEEE Congress on, vol. 1, pp. 222–227. IEEE (2005)
13. Ito, T., Iba, H., Sato, S.: Non-destructive depth-dependent crossover for genetic programming. In: European Conference on Genetic Programming, pp. 71–82. Springer (1998)
14. Jamali, A., Khaleghi, E., Gholaminezhad, I., Nariman-Zadeh, N., Gholaminia, B., Jamal-Omidi, A.: Multi-objective genetic programming approach for robust modeling of complex manufacturing processes having probabilistic uncertainty in experimental data. Journal of Intelligent Manufacturing **28**(1), 149–163 (2017)
15. Karger, D.: Random sampling in graph optimization problems. Ph.D. thesis, stanford university (1995)
16. Kenny, J.: Particle swarm optimization. In: Proc. 1995 IEEE Int. Conf. Neural Networks, pp. 1942–1948 (1995)
17. Koza, J.R.: Genetic Programming II, Automatic Discovery of Reusable Subprograms. MIT Press, Cambridge, MA (1992)
18. Koza, J.R.: Genetic programming as a means for programming computers by natural selection. Statistics and computing **4**(2), 87–112 (1994)
19. Langdon, W.B.: Size fair and homologous tree crossovers (1999)
20. Li, K., Wang, R., Zhang, T., Ishibuchi, H.: Evolutionary many-objective optimization: A comparative study of the state-of-the-art. IEEE Access **6**, 26194–26214 (2018)
21. Li, M., Grosan, C., Yang, S., Liu, X., Yao, X.: Multiline distance minimization: A visualized many-objective test problem suite. IEEE Transactions on Evolutionary Computation **22**(1), 61–78 (2018). DOI 10.1109/TEVC.2017.2655451
22. Li, M., Yang, S., Liu, X.: Diversity comparison of pareto front approximations in many-objective optimization. IEEE Transactions on Cybernetics **44**(12), 2568–2584 (2014). DOI 10.1109/TCYB.2014.2310651
23. Li, M., Yang, S., Liu, X.: Shift-based density estimation for pareto-based algorithms in many-objective optimization. IEEE Transactions on Evolutionary Computation **18**(3), 348–365 (2014)
24. Li, M., Zhen, L., Yao, X.: How to read many-objective solution sets in parallel coordinates [educational forum]. IEEE Computational Intelligence Magazine **12**(4), 88–100 (2017)
25. Mitchell, M., Crutchfield, J.P., Das, R., et al.: Evolving cellular automata with genetic algorithms: A review of recent work. In:

Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCAf96), vol. 8. Moscow (1996)

26. Mühlenbein, H., Paass, G.: From recombination of genes to the estimation of distributions i. binary parameters. In: International conference on parallel problem solving from nature, pp. 178–187. Springer (1996)
27. Nordin, P., Francone, F., Banzhaf, W.: Explicitly defined introns and destructive crossover in genetic programming. Advances in genetic programming **2**, 111–134 (1995)
28. Ohki, M.: Linear subset scheduling for many-objective optimization using nsga-ii based on pareto partial dominance. In: 15th International Conference on Informatics in Control, Automation and Robotics, vol. 1, pp. 277–283. INSTIC (2018)
29. Ohki, M.: Partial sampling operator and structural distance ranking for multi-objective gp. In: 2018 International Conference on Computer and Applications (ICCA), pp. 265–270 (2018)
30. Ohmoto, S., Takehana, Y., Ohki, M.: A consideration on relationship between optimizing interval and dealing day in optimization of stock day trading rules. IEICE technical report **113**(2103), 67–70 (2013)
31. Poli, R., Langdon, W.B.: On the search properties of different crossover operators in genetic programming. In: University of Wisconsin, pp. 293–301. Morgan Kaufmann (1998)
32. Poli, R., McPhee, N.F., Vanneschi, L.: The impact of population size on code growth in gp: analysis and empirical validation. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp. 1275–1282. ACM (2008)
33. Purshouse, R.C., Fleming, P.J.: Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In: International Conference on Evolutionary Multi-Criterion Optimization, pp. 16–30. Springer (2003)
34. Sato, H., Aguirre, H.E., Kiyoshi, T.: Effects of moea temporally switching pareto partial dominance on many-objective 0/1 knapsack problems. Transactions of the Japanese Society for Artificial Intelligence **25**, 320–331 (2010)
35. Sato, M., Aguirre, H.E., Tanaka, K.: Effects of δ-similar elimination and controlled elitism in the nsga-ii multiobjective evolutionary algorithm. In: Evolutionary Computation, 2006. CEC 2006. IEEE Congress on, pp. 1164–1171. IEEE (2006)
36. Sawada, K., Kano, H.: Structured evolution strategy for optimization problems using multi-objective methods. In: Proceedings of the 30th Symposium, Soc. of Instrument and Control Eng., Mar., 2003, pp. 1–6. Society of Instrument and Control Engineering (2003)
37. Tsuchida, K., Sato, H., Aguirre, H.E., Tanaka, K.: Analysis of nsga-ii and nsga-ii with cdas, and proposal of an enhanced cdas mechanism. JACIII **13**(4), 470–480 (2009)
38. Yang, J.M., Kao, C.Y.: An evolutionary algorithm to training neural networks for a two-spiral problem. In: Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, pp. 1025–1032. Morgan Kaufmann Publishers Inc. (2000)
39. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: Methods and applications, vol. 63. Citeseer (1999)
40. Zitzler, E., Laumanns, M., Thiele, L.: Spea2: Improving the strength pareto evolutionary algorithm. TIK-report **103** (2001)
41. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms comparative case study. In: international conference on parallel problem solving from nature, pp. 292–301. Springer (1998)
42. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE transactions on Evolutionary Computation **3**(4), 257–271 (1999)