

Variable space diversity, crossover and mutation in MOEA solving many-objective knapsack problems

Hiroyuki Sato · Hernán Aguirre · Kiyoshi Tanaka

Published online: 15 April 2012
© Springer Science+Business Media B.V. 2012

Abstract In this work, we analyze variable space diversity of Pareto optimal solutions (POS) and study the effectiveness of crossover and mutation operators in evolutionary many-objective optimization. First we examine the diversity of variables in the true POS on many-objective 0/1 knapsack problems with up to 20 items (bits), showing that variables in POS become noticeably diverse as we increase the number of objectives. We also verify the effectiveness of conventional two-point and uniform crossovers, Local Recombination that selects mating parents based on proximity in objective space, and two-point and uniform crossover operators which Controls the maximum number of Crossed Genes (CCG). We use NSGA-II, SPEA2, IBEA_{ε+} and MSOPS, which adopt different selection methods, and many-objective 0/1 knapsack problems with $n = \{100, 250, 500, 750, 1,000\}$ items (bits) and $m = \{2, 4, 6, 8, 10\}$ objectives to verify the search performance of each crossover operator. Simulation results reveal that Local Recombination and CCG operators significantly improve search performance especially for NSGA-II and MSOPS, which have high diversity of genes in the population. Also, results show that CCG operators achieve higher search performance than Local Recombination for $m \geq 4$ objectives and that their effectiveness becomes larger as the number of objectives m increases. In addition,

H. Sato (✉)

Faculty of Informatics and Engineering, The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585, Japan
e-mail: sato@hc.uec.ac.jp

H. Aguirre · K. Tanaka

Faculty of Engineering, Shinshu University, 4-17-1 Wakasato, Nagano, 380-8553, Japan

H. Aguirre

e-mail: ahernan@shinshu-u.ac.jp

K. Tanaka

e-mail: ktanaka@shinshu-u.ac.jp

the contribution of CCG and mutation operators for the solutions search is analyzed and discussed.

Keywords Multiobjective evolutionary algorithms • Many-objective optimization • Many-objective 0/1 knapsack problem • Local recombination • Controlling the number of crossed genes

Mathematics Subject Classification (2010) 90C27

1 Introduction

Multi-objective optimization tries to find Pareto optimal solutions (POS) that capture the trade-off among conflicting objective functions. To find POS in Multi-objective optimization problems (MOPs), multi-objective evolutionary algorithms (MOEAs) are particularly suitable because they evolve simultaneously a population of potential solutions to the problem in hand, which allows us to search a set of POS in a single run of the algorithm [1]. So far, MOEAs have been successfully applied mostly in two and three objective problems [1].

Recently, the research interest of the MOEA community has rapidly shifted to develop effective algorithms for many-objective optimization problems (MaOPs), which optimize four or more objective functions simultaneously. However, in general, MOEAs noticeably deteriorate their search performance as we increase the number of objectives [2, 3], especially Pareto dominance-based MOEAs such as NSGA-II [4] and SPEA2 [5]. One reason for this is that these MOEAs face difficulty to rank solutions in the population, i.e., most of the solutions become non-dominated and the same rank is assigned to them, which seriously spoils proper selection pressure required in the evolution process.

To overcome this problem in selection, several methods have been proposed [6]. One approach is to improve selection pressure and assign more fine-grained rank to solutions by using extensions of Pareto dominance relation [7–12] or aggregating objective functions [2, 13, 14]. A comparative study of several selection schemes on MaOPs has been made in [15]. Other approaches aim to reduce the dimensionality of the problem, so that conventional algorithms known to perform well in low dimensional spaces can be used [16–18]. User preferences are also being investigated, aiming to focus the search on a small region of the Pareto front [19–21]. Most of the methods proposed so far have focused on improving convergence of solutions in objective space. However, little attention has been given to the distribution of solutions in objective space, the diversity of solutions in variable space, and the effectiveness of recombination operators.

Contrary to these studies, in this work we focus on the diversity of POS in variable space. The motivation of the present work is to show the importance of taking into account a fundamental feature of many-objective landscapes when we design a recombination operator. The contributions of this work are as follows. We show that the dispersion extent of optimal solutions in variable space gets larger with the number of objectives. To do that, first we analyze variable space diversity in the true

POS obtained by exhaustive search on many-objective 0/1 knapsack problem [22] with $n = \{10, 15, 20\}$ bits (items), showing that variables in POS become noticeably diverse in the same way as the ratio of POS in feasible solution space increases with the number of objectives. In MOEAs, if variables of solutions in the population become noticeably diverse, conventional recombination might become too disruptive and decrease its effectiveness.

To verify this hypothesis, next we examine the effectiveness of conventional two-point and uniform crossover, Local Recombination that selects mating parents based on proximity in objective space and two-point and uniform crossover operators which Controls the maximum number of Crossed Genes (CCG). To verify the effectiveness of each crossover operator, we use NSGA-II [4], SPEA2 [5], IBEA_{ε+} [9] and MSOPS [2], well known MOEAs that adopt different selection methods, and many-objective 0/1 knapsack problems with $n = \{100, 250, 500, 750, 1,000\}$ items (bits) and $m = \{2, 4, 6, 8, 10\}$ objectives. In addition, we analyze the difference of roles between CCG and mutation operators, and compare the number of offspring that survive selection by CCG operators and conventional crossovers.

2 Evolutionary multi and many-objective optimization

2.1 Multi and many-objective optimization concepts and definitions

A multiobjective optimization problem including m kinds of objective functions is defined as follows:

$$\begin{cases} \text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{subject to } \mathbf{x} \in \mathcal{F} \end{cases} \quad (1)$$

where, \mathcal{F} is the feasible solutions and $\mathbf{x} \in \mathcal{F}$ is a feasible solution vector in the solution space \mathcal{S} ($\mathcal{F} \subseteq \mathcal{S}$), and $f_i (i = 1, 2, \dots, m)$ are the m objectives to be maximized. That is, we try to find a feasible solution vector $\mathbf{x} \in \mathcal{F}$ in the solution space maximizing each objective function $f_i (i = 1, 2, \dots, m)$ in a vector fitness function \mathbf{f} . Important concepts used in determining a set of solutions for multiobjective optimization problems are dominance, Pareto optimality, Pareto set and Pareto front. Next we define *dominance* between solutions $\mathbf{x}, \mathbf{y} \in \mathcal{F}$ as follows: If

$$\begin{aligned} \forall i \in \{1, 2, \dots, m\} : f_i(\mathbf{x}) &\geq f_i(\mathbf{y}) \quad \wedge \\ \exists i \in \{1, 2, \dots, m\} : f_i(\mathbf{x}) &> f_i(\mathbf{y}). \end{aligned} \quad (2)$$

are satisfied, \mathbf{x} dominates \mathbf{y} . In the following, \mathbf{x} dominates \mathbf{y} is denoted by $\mathbf{f}(\mathbf{x}) \succ \mathbf{f}(\mathbf{y})$. A solution vector \mathbf{x} is said to be *Pareto optimal* with respect to \mathcal{F} if it is not dominated by other solution vectors in \mathcal{F} . The presence of multiple objective functions, usually conflicting among them, gives rise to a set of optimal solutions. The set of Pareto optimal solutions (POS) is defined as

$$\mathcal{POS} = \{\mathbf{x} \in \mathcal{F} \mid \neg \exists \mathbf{y} \in \mathcal{F} : \mathbf{f}(\mathbf{y}) \succ \mathbf{f}(\mathbf{x})\}, \quad (3)$$

and the Pareto front is defined as

$$\mathcal{Front} = \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{POS}\}. \quad (4)$$

If the problem is multi-objective then the presence of trade-off solutions is unavoidable and Pareto dominance is useful to identify such trade-off solutions. In the case of many-objective problems, the number of trade-off solutions increases exponentially with the number of objectives [1, 3, 11, 23]. This is an underlying characteristic of many-objective landscapes, independently of the method we use to search for them. In practice one or few preferred solutions are implemented. However, searching for a large number of Pareto optimal solutions (that give a good representation of the true Pareto front) makes perfect sense, because they are essential to extract relevant knowledge about the optimal region of the problem, both in objective and variable space. This knowledge provides useful guidelines to designers (or decision makers) during the implementation of the preferred solutions.

The goal of multi and many-objective optimization is to obtain POS which can approximate the entire true Pareto optimal front. Thus, it is important to simultaneously achieve for the obtained POS convergence and diversity towards the true Pareto front. Evolutionary algorithms (EAs) are particularly suitable for this task because they evolve simultaneously a population of potential solutions to the problem at hand, which allows us to search a set of Pareto non-dominated optimal solutions in a single run of the algorithm [1].

2.2 MOEAs for many-objective optimization problems

In representative MOEAs, such as NSGA-II [4] and SPEA2 [5], selection incorporates elitism and it is biased by Pareto dominance ranking and a diversity preserving strategy in objective space. Pareto dominance based selection has been successfully applied, especially in two and three objective problems. Recently, there is a growing interest in applying MOEAs to solve many-objective optimization problems (MaOPs) which optimize four or more objective functions simultaneously. However, current research reveals that ranking by Pareto dominance loses its effectiveness as we increase the number of objectives of the problem, severely deteriorating the performance of MOEAs for many objectives optimization [2, 3]. To overcome this problem, several studies have been made on methods to determine the superiority of non-dominated solutions in a more effective manner in order to strengthen parent selection pressure [6, 15].

Contrary to these studies, in this work we study on effective genetic operator in evolutionary many-objective optimization based on analysis of variable space diversity in the Pareto optimal solutions. It is well known that in MaOPs the number of non-dominated solutions increases considerably with the number of objectives. However, not much is known about the distribution of those solutions in decision space, how selection shapes that distribution in the instantaneous sample population, and how both the distribution in variable space and selection in objective space influence the effectiveness of genetic operators of MOEAs in MaOPs.

3 Analysis of many-objective 0/1 Knapsack problem

3.1 Definition of multi-objective 0/1 Knapsack problem

In this paper we focus on many-objective 0/1 knapsack problems [22]. The problem is formulated to maximize the function

$$f_j(\mathbf{x}) = \sum_{i=1}^n x_i \cdot p_{i,j} \quad (j = 1, 2, \dots, m) \quad (5)$$

subject to

$$g_j(\mathbf{x}) = \sum_{i=1}^n x_i \cdot w_{i,j} \leq W_j \quad (j = 1, 2, \dots, m) \quad (6)$$

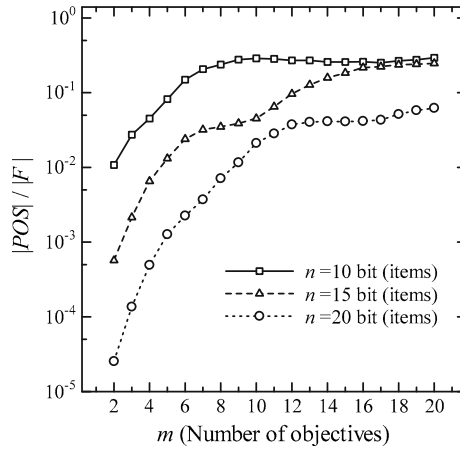
where $x_i \in \{0, 1\}$ ($i = 1, 2, \dots, n$) are elements of solution vector $\mathbf{x} = (x_1, x_2, \dots, x_n)$, which gives a combination of items. Thus, we use binary representation in this work. Note that here we are interested in finding a set of non-dominated Pareto optimal solutions (POS). Also, $p_{i,j}$ and $w_{i,j}$ ($j = 1, 2, \dots, m$) denote profit and weight of item i according to knapsack (objective) j . W_j is the capacity of knapsack j , and solutions not satisfying this condition are considered as infeasible solutions $\overline{\mathcal{F}} = (\mathcal{S} - \mathcal{F})$. The capacity of knapsack is given by $W_j = \phi \cdot \sum_{i=1}^n w_{i,j}$ based on feasibility ratio ϕ which is ratio of solutions which do not exceed the capacity of each knapsack in all solutions of variable space. In this problem, we can control the number of objectives by varying m , the size of search space by varying n , and the feasibility ratio of the search space by ϕ . Similar to [22], in this work $p_{i,j}$ and $w_{i,j}$ are generated as random integers in the interval [10, 100].

3.2 Analysis of Pareto optimal solutions in many-objective 0/1 Knapsack problem

Here, we analyze many-objective 0/1 knapsack problems [22] by observing the number of Pareto optimal solutions |POS| and their variable space diversity in discrete solution space. Here, we generate problems with $n = \{10, 15, 20\}$ items (bits) and $m = 2 \sim 20$ objectives, setting the feasibility ratio $\phi = 0.5$. We generate 90 problems for each combination of parameters m and n , find all true POS by exhaustive search in solution space \mathcal{S} , and analyze average results.

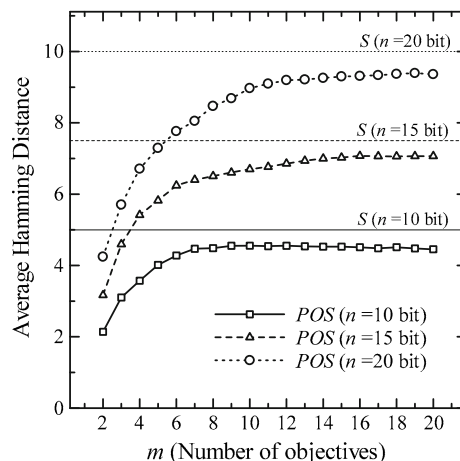
Figure 1 shows the ratio $|\text{POS}|/|\mathcal{F}|$ of true POS in feasible solution space $\mathcal{F} (\subseteq \mathcal{S})$. From these results, we can see a tendency that the ratio of POS in \mathcal{F} increases significantly with the number of objectives m . Also, note that it appears like the fraction for $n = \{15, 20\}$ tends to approach the fraction obtained on $n = 10$ bits for large m . It will be interesting to analyze a larger number of objectives to elucidate this convergence tendency with more detail. Next, to observe the variable space diversity of POS, Fig. 2 shows the average hamming distance of POS. Here, we also plot the average hamming distance of all solutions in the solution space \mathcal{S} as a horizontal line. From these results, note that the average hamming distance of POS noticeably increases with the number of objectives m . In case of a small number of objectives m , the ratio of POS in \mathcal{F} is relatively low and the average hamming distance of POS is

Fig. 1 Ratio of true Pareto optimal solutions POS in feasible solution space \mathcal{F}



short compared to the average hamming distance of all solutions \mathcal{S} , suggesting that POS are distributed in a relatively narrow region in the solution space \mathcal{S} . On the other hand, the ratio of POS in \mathcal{F} increases with the number of objectives m and the average hamming distance also increases, approaching to the average hamming distance of all solution in \mathcal{S} . For example, on $n = 20$ bits and $m = 20$ objectives, around 6% of feasible solutions become POS and the average hamming distance of POS is 9.36 bits, which is very close to the 10 bits average hamming distance of all solutions in \mathcal{S} . This tendency is also observed in problems with $n = \{10, 15\}$ bits, where the average hamming distances of POS $\{4.45, 7.06\}$ are close to the average hamming distance of \mathcal{S} $\{5.0, 7.5\}$, respectively. These results suggest that POS becomes distributed almost uniformly in solution space by increasing m . That is, we can expect that genes become noticeably diverse in the population during evolutionary many-objective optimization. Also, the effectiveness of exploitation by the conventional recombination might decrease if difference of genes between two parents becomes too large.

Fig. 2 Average hamming distance of true POS



4 Mating based on proximity in objective space

4.1 Related works

To realize effective recombination of solutions in MOEAs, several studies that apply crossover for two parents located near each other in the objective function space have been made. NCGA (Neighborhood Cultivation GA) introduces neighborhood crossover in the objective function space [24]. In NCGA, after sorting solutions in the population according to one objective function value, two neighboring solutions become a pair for crossover. The improvement of convergence and diversity of obtained POS by NCGA on continuous and combinatorial 0/1 knapsack problems with two objectives functions has been reported in [24]. In another study, Local Recombination [25] selects pairs of parents by considering proximity of the search direction of solutions, using a locality parameter n_{LR} . When we use a small n_{LR} , parents are selected with high locality in the objective space. Increasing n_{LR} , the neighborhood expands and in the extreme it comes to be the entire parent population. The extreme is equivalent to conventional recombination because mates are selected without restriction from the entire parent population. The effectiveness of Local Recombination has been verified on $m = 2 \sim 4$ objective problems [25]. In [26], a mating scheme was proposed to select one pair of solutions for crossover by first selecting various candidates performing multiple binary tournaments, and then picking two of them based on their distance in the objective space. This method controls the balance between convergence and diversity of obtained solutions by the mating scheme. Additionally, MOEA/D [27] utilizes multiple scalarization functions to find POS, selecting pair of solutions for recombination from solutions that maximize neighbor scalarization functions.

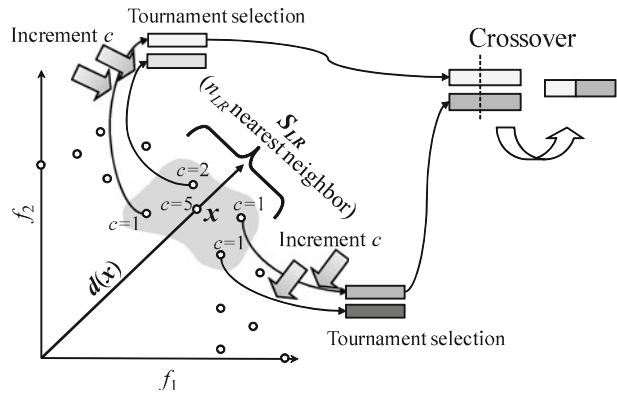
In terms of avoiding the inefficient recombination of solutions having very different objective function values, it is thought that these methods bring similar effects for the search performance of MOEAs. In problems where there is some correlation between objective and variable space, it is expected that these methods can effectively apply crossover to solutions that have relatively similar gene structure even in MaOP. In this paper we pick up Local Recombination [25] which controls locality in objective space for recombination with parameter n_{LR} and verify its effectiveness on MaOPs for performance comparison. Note that in epistatic problems, where solutions close by in objective space might be far away in variable space, mating based on a neighborhood calculated in objective space may not lead to improve the effectiveness of recombination.

4.2 Local Recombination (LR)

To create n_{LR} neighbors which have similar search direction, the original Local Recombination method utilizes the angle information of the objective function values transformed to polar coordinate vectors [25]. In this work, we use search direction $\mathbf{d}(\mathbf{x}) = (d_1(\mathbf{x}), d_2(\mathbf{x}), \dots, d_m(\mathbf{x}))$ calculated by the fraction of each objective function value $f_i(\mathbf{x})$ on the sum of all m objective function values, namely

$$d_i(\mathbf{x}) = f_i(\mathbf{x}) / \sum_{j=1}^m f_j(\mathbf{x}) \quad (i = 1, 2, \dots, m). \quad (7)$$

Fig. 3 Neighborhood creation and mating for Local Recombination [25]



Then, we calculate the Euclidean distance between $d(x)$ and the search direction of other solutions, and create a sub-population S_{LR} of n_{LR} neighboring solutions, as shown in Fig. 3. Here, LR stands for Local Recombination [25]. Note that n_{LR} is the locality parameter for recombination. Similar to [25], mating is performed within the neighborhood S_{LR} and then recombination followed by mutation are carried out. We enforce equal participation in the tournaments. To accomplish that we keep for each individual in the parent population P_t a counter c showing the number of times it has participated in a tournament and select the individuals that will undergo a binary tournament randomly from among those with smallest value in its counter c . Note that the individual's counters are not re-initialized until the offspring population Q_t has been completely created. Varying the number of elements in the neighborhood $n_{LR} \leq |P_t|$ we can control the degree of locality for recombination. In the extreme, $n_{LR} = |P_t|$, we have conventional recombination as mentioned in the previous subsection.

5 Controlling crossed genes for crossover

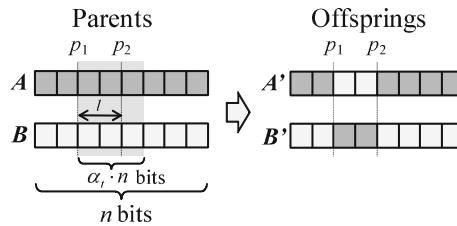
5.1 Problem of Local Recombination in MaOPs

Since Local Recombination selects mates having similar search direction, the probability that some selected pairs of solutions have similar genes structure increases. However, as mentioned in Section 3.2, the diversity of genes in solutions noticeably increase in MaOPs. In this case, even if we select neighborhood solutions in objective space for recombination, it is expected that they have a large difference in genes and recombination might be inefficient. To solve this problem, in this work we consider two methods to restrict the length of crossed genes when we apply crossover in MaOPs.

5.2 CCG for two-point crossover (CCG_{TX})

When we apply the conventional one- or two-point crossover for individuals with n genes, the length of crossed genes vary in the range $[0, n]$ by randomly choosing the crossover point(s). To restrict the variation of genes in crossover for parents having large difference in gene structure, in this work we propose to control the

Fig. 4 Controlling crossed genes for two-point crossover (CCG_{TX})

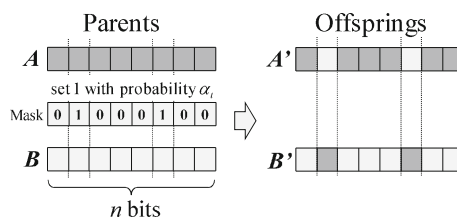


maximum number of crossed genes in crossover for MaOP. In this section we explain a method for controlling crossed genes (CCG) in two-point crossover (CCG_{TX}). CCG_{TX} controls the length of crossed genes by using a user-defined parameter α_t . Figure 4 shows the conceptual diagram of CCG_{TX}. First we select parents **A** and **B** from the parent population \mathcal{P}_t , and randomly choose the 1st crossover point p_1 . Then, we randomly determine the length of the crossed genes ℓ in the range $[0, \alpha_t \cdot n]$. The second crossover point is set to $p_2 = (p_1 + \ell) \bmod n$. Here, the possible range of the parameter α_t is $[0.0, 1.0]$. In this method, when we utilize a small α_t , the maximum length of crossed segment becomes short. On the other hand, when we utilize a large α_t , the maximum length of crossed segment becomes long. In the case of $\alpha_t = 0.0$, since the length of the crossed segment becomes $\alpha_t \cdot n = 0$, the solutions search is equivalent to mutation-only without crossover. Also, in the case of $\alpha_t = 1.0$, the maximum length of the crossed segment become $\alpha_t \cdot n = n$. This case is equivalent to the conventional two-point crossover. In this work we verify the effects of CCG_{TX} in MOEA as we vary α_t in the range $\alpha_t \in [0.0, 1.0]$.

5.3 CCG for uniform crossover (CCG_{UX})

Next, we explain a method for CCG in uniform crossover (CCG_{UX}). As shown in Fig. 5, for uniform crossover we randomly select two parents from the parent population and generate a n bit mask [28, 29]. For offspring **A'**, if mask bit is 0, the gene is copied from parent **A**. If mask bit is 1, the gene is copied from parent **B**. Similarly, for offspring **B'**, if mask bit is 0, the gene is copied from parent **B**. If mask bit is 1, the gene is copied from parent **A**. To control the number of crossed genes, in this work we control the probability of 1 in the mask by using the parameter α_u . The possible range of α_u is $[0, 1]$, and $\alpha_u = 0.5$ indicates typical uniform crossover [28]. In this method, when we utilize a small α_u , the number of crossed genes becomes small. On the other hand, when we utilize a large α_u , the number of crossed genes becomes large. $\alpha_u = 0.0$ is equivalent to mutation-only without crossover. Also, $\alpha_u = 1.0$ is equivalent to $\alpha_u = 0.0$ because all gene are exchanged in this crossover. In this work we verify the effects of CCG_{UX} in MOEA as we vary α_u in the range $\alpha_u \in [0.0, 0.5]$.

Fig. 5 Controlling crossed genes for uniform crossover (CCG_{UX})



6 Preparation

6.1 Algorithms and selection methods

To verify the effectiveness of Local Recombination [25], CCG_{TX} and CCG_{UX}, in this work we implement them in NSGA-II [4], SPEA2 [5], IBEA_{ε+} [9] and MSOPS [2], which use different selection methods. NSGA-II and SPEA2 are dominance based MOEAs that use Pareto dominance to determine the superiority of solutions in parent selection. IBEA_{ε+} (Indicator-based Evolutionary Algorithm) introduces fine grained ranking of solutions by calculating fitness value based on the indicator $I_{ε+}$ which measures the degree of superiority for each solution in the population [9]. MSOPS (Multiple single objective Pareto sampling) aggregates fitness vector with multiple weight vectors, and reflects the ranking of solutions calculated for each weight vector in parent selection [2].

According to a previous performance comparison [12], in NSGA-II the convergence of obtained POS gradually deteriorates as we increase the number of objectives m , but the diversity of POS significantly increases. On the other hand, POS obtained by IBEA_{ε+} achieves extremely high convergence but scarce diversity. In contrast, MSOPS realizes a well-balanced search between convergence and diversity in MaOPs.

6.2 Problems, parameters and metrics

For performance verification, we use many-objective 0/1 knapsack problems with $m = \{2, 4, 6, 8, 10\}$ objectives, $n = \{100, 250, 500, 750, 1,000\}$ items, and feasibility ratio $\phi = 0.5$. To deal with infeasible solutions we use the greedy repair mechanism proposed in [22] for all algorithms to be compared in this paper. Also, we adopt crossover with a crossover rate $P_c = 1.0$, and apply bit-flipping mutation with a mutation rate $P_m = 1/n$. In the following experiments, we show the average performance with 30 runs, each of which spent $T = 2,000$ generations. Population size is set to $N = 200$ (parent population size $|P_t| = 100$ and offspring population size $|Q_t| = 100$). In IBEA_{ε+}, scaling parameter κ is set to 0.05 similar to [9]. Also, in MSOPS, we use $W = 100$ uniformly distributed weight vectors [15], which maximizes *Hypervolume* (HV) [30] in the experiments. NSGA-II is a well known algorithm whose performance scales up very badly with the number of objectives. The decision to test our hypothesis on recombination using selection and mating as it is applied in conventional NSGA-II is precisely to show that a bad performing algorithm improves significantly just by taking into account the characteristics of many-objective landscapes in the design of the recombination operator. For similar reasons, in our experiments we use relatively small population sizes, although there is evidence that larger populations improve the evolutionary search on many-objective problems [31].

In this work, to evaluate the search performance of MOEAs we use HV , which measures the m -dimensional volume of the region enclosed by the obtained non-dominated solutions and a dominated reference point in objective space. Here we use $\mathbf{r} = (0, 0, \dots, 0)$ as the reference point. Obtained POS showing a higher value of HV can be considered as a set of solutions with either better convergence, better

diversity or both better convergence and diversity. To calculate the hypervolume, we use the improved dimension-sweep algorithm proposed by Fonseca et al. [32], which significantly reduces computational time especially for large m . To provide additional information separately on convergence and diversity of the obtained POS, in this work we also use *Norm* [33] and *Maximum Spread (MS)* [30], respectively. Higher value of *Norm* generally means higher convergence to true POS. Although *Norm* cannot precisely reflect local features of the distribution of the obtained POS, we can observe the general tendency of convergence for POS from their values. On the other hand, higher *MS* indicates better diversity in POS, i.e., a widely spread Pareto front.

7 Experimental results and discussion

7.1 Diversity of genes in the population obtained by conventional crossover

First, we observe the diversity of genes in the population at the final generation when conventional two-point crossover is used. Figure 6 shows the average hamming distance of solutions in the population obtained by NSGA-II, SPEA2, IBEA $_{\epsilon+}$ and MSOPS on many-objective 0/1 knapsack problems with $m = \{2, 4, 6, 8, 10\}$ objectives and $n = \{100, 250, 500, 750, 1,000\}$ items (bits).

For NSGA-II, SPEA2 and IBEA $_{\epsilon+}$, we can see that the average hamming distance increases as we increase the number of objectives m . This tendency is similar to the aforementioned results obtained by exhaustive search on $n = \{10, 15, 20\}$ bits problems, as shown in Fig. 2. On the other hand, although MSOPS shows higher average hamming distance than other MOEAs in $m = \{2, 4\}$ objectives, the average hamming distance tends to become short for $m \geq 6$. Also, we can see that the average hamming distance obtained by IBEA $_{\epsilon+}$ is the shortest among all MOEA compared in Fig. 6. That is, the population obtained by IBEA $_{\epsilon+}$ is distributed in a narrow region of solution space. On the other hand, the population obtained by NSGA-II shows the highest average hamming distance for $m \geq 6$. That is, the population obtained by NSGA-II is widely distributed in solution space. In the case of $m = 10$ objectives and $n = 1,000$ bits, note that the average hamming distance obtained by NSGA-II becomes around 250 bits at the final generation. In this case, if we randomly select two solutions from the population as parents, they will be different in 250 bits out of 1,000 bits. Thus, since diversity of genes in the population obtained by NSGA-II is significantly high in MaOPs, the likelihood that the conventional recombination becomes too disruptive is also high, making it an inefficient genetic operator for solutions search.

Next, Fig. 7 shows the transition of the number of offspring that survive selection and become a part of the population P_t by NSGA-II with conventional two-point and uniform crossover. From the result, we can see that the number of offspring that survived selection substantially decreases by increasing the number of objectives m . This result suggests that the contribution of conventional genetic operator for solutions search substantially deteriorates in MaOPs due to the increase of genetic diversity in the population as shown in Fig. 6.

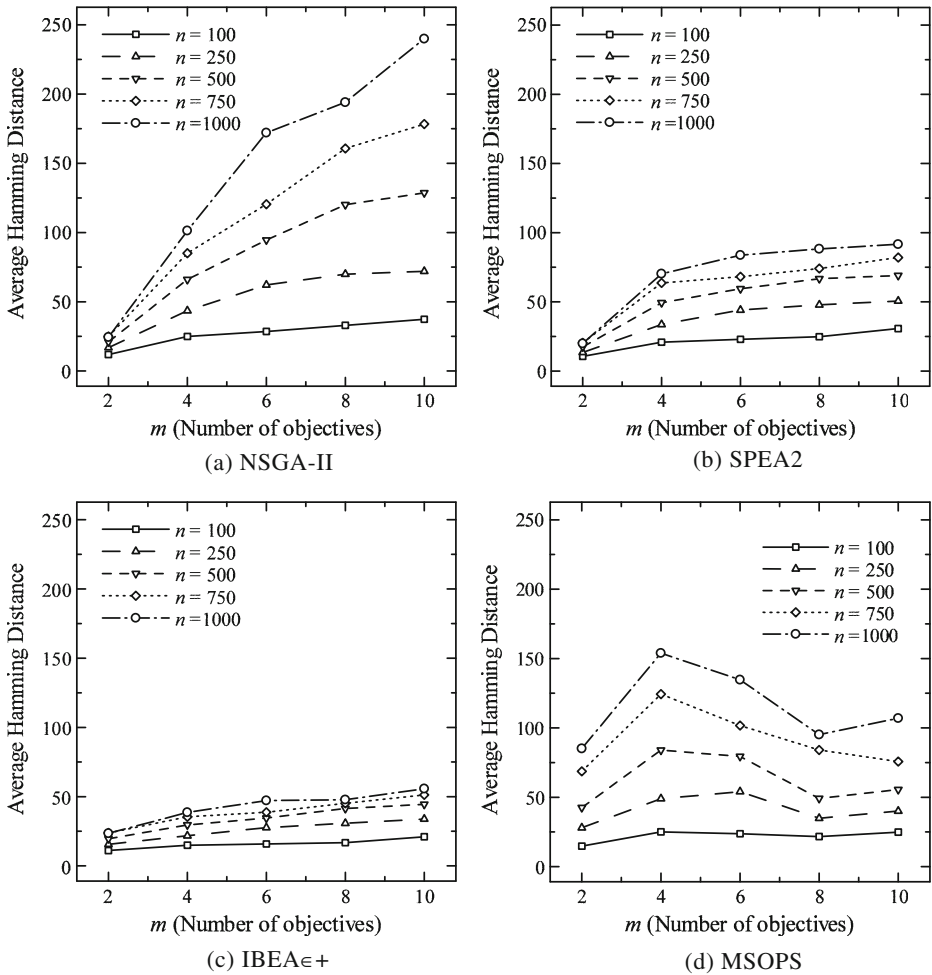


Fig. 6 Average hamming distance of solutions in the population at the final generation

7.2 Effects of Local Recombination in MaOPs

Next, we observe the effects of Local Recombination [25] in NSGA-II, SPEA2, IBEA ϵ +, and MSOPS on problems with $n = 1,000$ items (bits) and $m = \{2, 4, 6, 8, 10\}$ objectives. Figs. 8, 9, 10 and 11 show results on HV as a combined metric of convergence and diversity, $Norm$ as a measure of convergence, and MS as a measure of diversity, varying the locality of recombination n_{LR} . In the case of $n_{LR} = 4$, tournament selection for recombination is performed in highest locality. Increasing n_{LR} the locality of recombination decrease, and the conventional recombination is applied when we utilize $n_{LR} = 100$. After we select a pair of parents, the conventional two-point crossover is applied. In these figures, all the plots are normalized by the results of NSGA-II using the conventional two-point crossover.

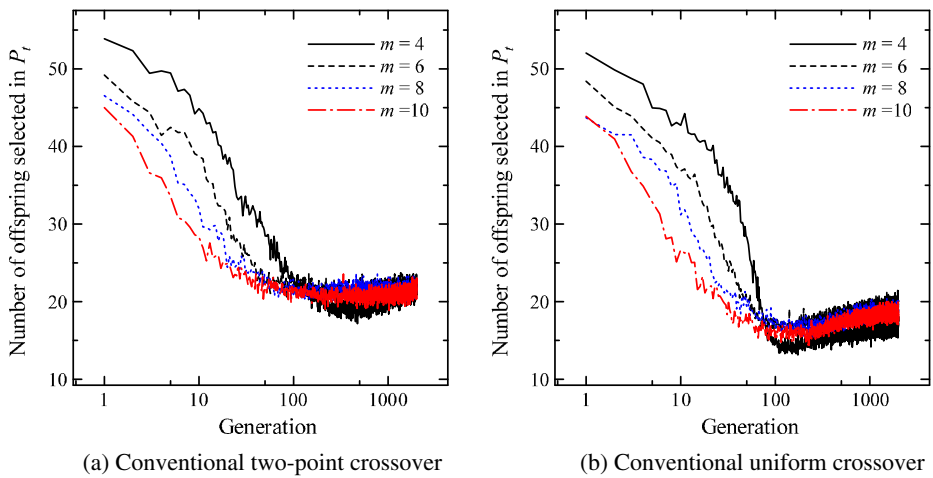


Fig. 7 Number of offsprings selected in P_t by NSGA-II using the conventional two-point and uniform crossover ($n = 1,000$)

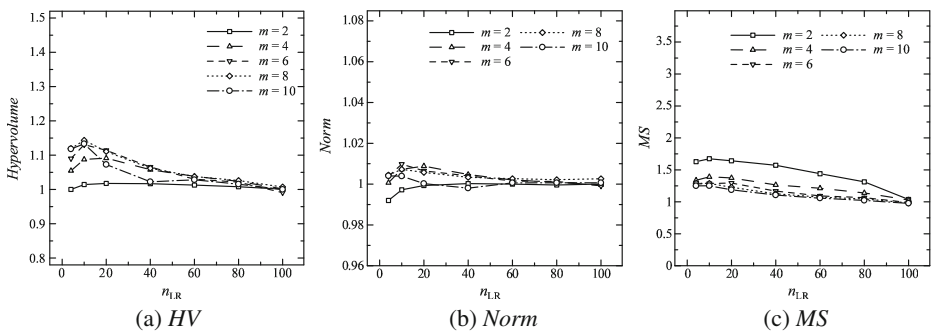


Fig. 8 Performance obtained by NSGA-II [4] with Local Recombination ($n = 1,000$)

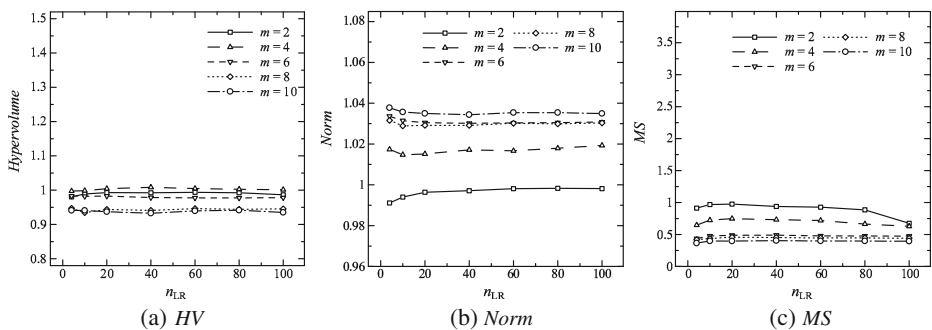


Fig. 9 Performance obtained by SPEA2 [5] with Local Recombination ($n = 1,000$)

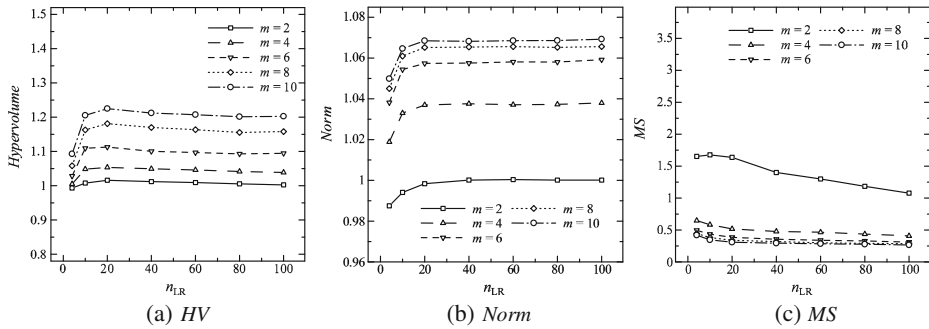


Fig. 10 Performance obtained by IBEA $_{\epsilon+}$ [9] with Local Recombination ($n = 1,000$)

First, from results of HV in Figs. 8–11a, we can see that IBEA $_{\epsilon+}$ and MSOPS with conventional recombination ($n_{LR} = 100$) achieve higher HV than dominance based NSGA-II and SPEA2 with conventional recombination as we increase the number of objectives m . When we decrease n_{LR} and enhance the locality of selected pair of parents in objective space, we see improvements on HV by NSGA-II and MSOPS, but not by SPEA2 and IBEA $_{\epsilon+}$. Improvement of HV by NSGA-II becomes significant as we increase the number of objectives m . In the case of $m = 8$ objectives, although HV obtained by NSGA-II with conventional recombination ($n_{LR} = 100$) is lower than IBEA $_{\epsilon+}$ and MSOPS, NSGA-II using Local Recombination with $n_{LR} = 10$ achieves higher HV than MSOPS and comparative with IBEA $_{\epsilon+}$.

Next, from results of $Norm$ in Figs. 8–11b, SPEA2, IBEA $_{\epsilon+}$ and MSOPS achieve higher $Norm$ than NSGA-II especially for large number of objectives m . We can see that small improvement in $Norm$ is obtained by NSGA-II as we decrease n_{LR} , but SPEA2, IBEA $_{\epsilon+}$ and MSOPS do not improve $Norm$ by varying n_{LR} .

Next, from results of MS in Figs. 8–11c, all MOEAs improve MS by decreasing n_{LR} . MS obtained by SPEA2 and IBEA $_{\epsilon+}$ are relatively lower than NSGA-II and MSOPS. Consequently, average hamming distance of the population by SPEA2 and IBEA $_{\epsilon+}$ becomes short in Fig. 6, and these populations are distributed in a relatively narrow region in objective/solution space.

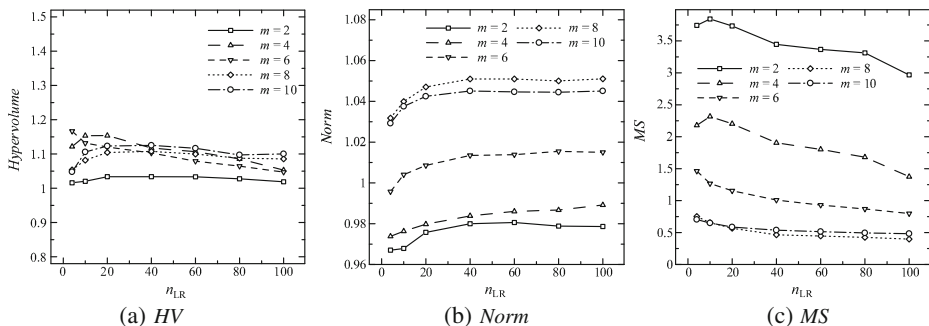


Fig. 11 Performance obtained by MSOPS [2] with Local Recombination ($n = 1,000$)

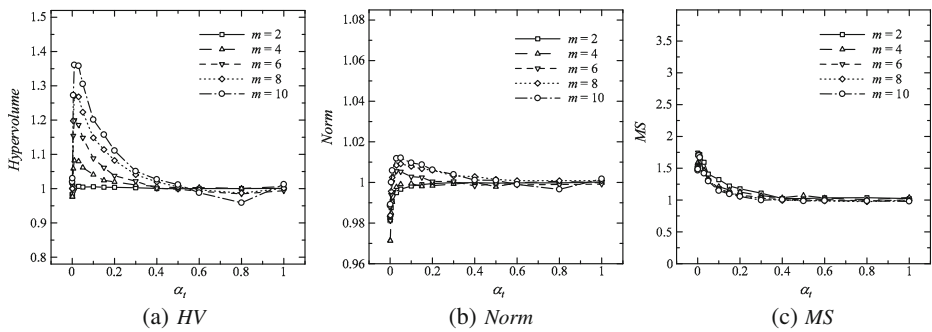


Fig. 12 Performance obtained by NSGA-II [4] with CCG_{TX} ($n = 1,000$)

Summarizing, it is difficult to induce the effectiveness of Local Recombination in SPEA2 and $IBEA_{\epsilon+}$ which evolve less diverse populations. On the other hand, although NSGA-II obtains well-spread solutions, convergence towards Pareto optimal front is not sufficient. In contrast, MSOPS achieves well-balanced search between convergence and diversity of obtained solutions. Since NSGA-II and MSOPS achieve relatively high diversity of solutions in objective/solution space, average hamming distance becomes large as shown in Fig. 6. Thus, the effectiveness of Local Recombination becomes noticeable in NSGA-II and MSOPS which evolve well-spread populations. Also, the effectiveness becomes significant increasing the number of objectives m .

7.3 Effects of CCG_{TX} in MaOPs

Next, we observe the effects of CCG_{TX} in NSGA-II, SPEA2, $IBEA_{\epsilon+}$ and MSOPS. Figures 12, 13, 14 and 15 show results on HV , $Norm$ and MS varying the parameter α_t . Similar to the previous sub-section, all plots are normalized by the results of NSGA-II using conventional two-point crossover.

First, from results of HV in Figs. 12–15a, we can see that there is no improvement when we use CCG_{TX} varying α_t in SPEA2 and $IBEA_{\epsilon+}$, which evolve less diverse

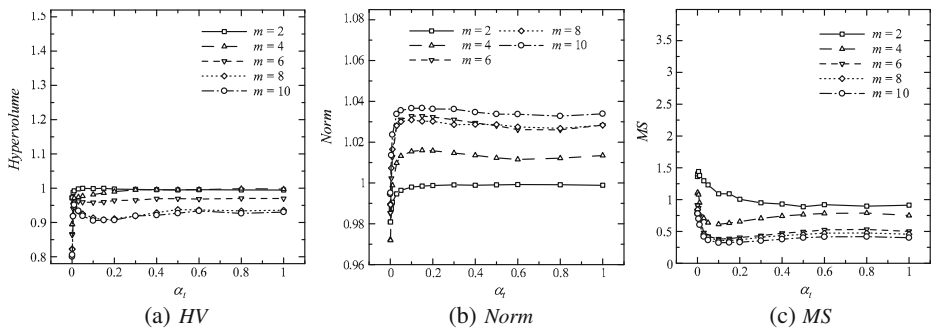


Fig. 13 Performance obtained by SPEA2 [5] with CCG_{TX} ($n = 1,000$)

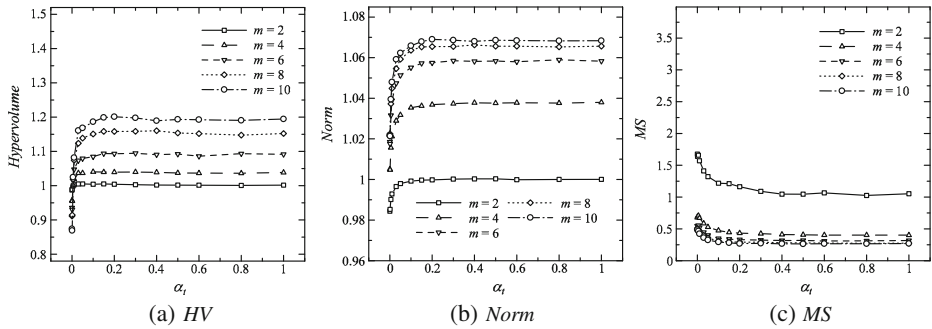


Fig. 14 Performance obtained by IBEA $_{\epsilon+}$ [9] with CCG $_{TX}$ ($n = 1,000$)

solutions in the population. On the other hand, NSGA-II and MSOPS having well-spread population significantly improve *HV* when we set small α_l . Compared with *HV* achieved by Local Recombination shown in Figs. 8–11a, the maximum *HV* obtained by NSGA-II with CCG $_{TX}$ is higher than the maximum *HV* obtained by NSGA-II with Local Recombination. The same is true for MSOPS with CCG $_{TX}$ compared to MSOPS with Local Recombination. Next, from results of *Norm* in Figs. 12–15b, it can be seen that NSGA-II, SPEA2 and MSOPS improve the convergence of obtained POS by using smaller α_l . Also, from results of *MS* in Figs. 12–15c, as general tendency, we can see that *MS* improves by decreasing α_l . It is interesting to note that, although MSOPS achieves the highest *HV* in $m = 6$ objectives, NSGA-II achieves highest *HV* in $m = \{8, 10\}$ objectives problems. This is because deterioration of *MS* in MSOPS becomes significant for a large number of objectives. From these results, we can conclude that the effectiveness of CCG $_{TX}$ becomes significant especially for NSGA-II and MSOPS because these MOEAs evolve well-spread solutions in the population. Also, we can conclude that the improvement of *HV* by CCG $_{TX}$ is due to the improvement of *MS* and *Norm*. *HV* obtained by CCG $_{TX}$ is higher than *HV* obtained by Local Recombination especially for large number of objectives. This is because crossover under Local Recombination still might be too disruptive, especially for large m , whereas CCG $_{TX}$ can control better the number of genes being crossed.

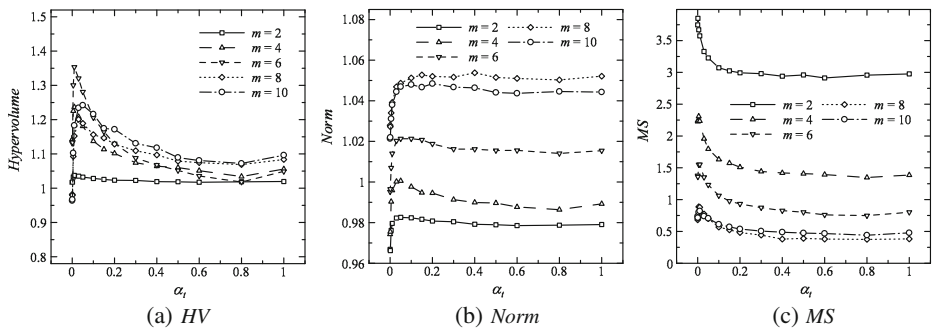


Fig. 15 Performance obtained by MSOPS [2] with CCG $_{TX}$ ($n = 1,000$)

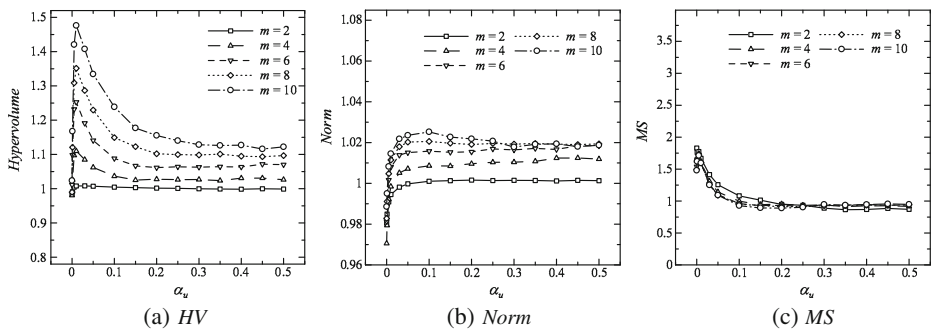


Fig. 16 Performance obtained by NSGA-II [4] with CCG_{UX} ($n = 1,000$)

7.4 Effects of CCG_{UX} in MaOPs

Next, we observe the effects of CCG_{UX} in NSGA-II, SPEA2, IBEA $_{\epsilon+}$ and MSOPS. Figures 16, 17, 18 and 19 show results on HV , $Norm$, and MS varying the parameter α_u . Similar to previous sections, all plots are normalized by the results of NSGA-II using conventional two-point crossover. Results obtained by CCG_{UX} have similar tendency to results obtained by CCG_{TX} shown in Figs. 12–15.

However, values of $Norm$ obtained by CCG_{UX} become higher than CCG_{TX} . Consequently, CCG_{UX} achieves higher HV than CCG_{TX} due to the improvement of $Norm$. It should be noted that in the case of NSGA-II, $Norm$ obtained by CCG_{UX} using α_u^* is slightly lower than the one obtained by typical uniform crossover ($\alpha_u = 0.5$). On the other hand, values of MS improved by decreasing α_u , and this tendency is similar to MS obtained by CCG_{TX} . Therefore, in the case of CCG_{UX} , we can conclude that the improvement of HV is due to the improvement of MS in NSGA-II. We can see similar tendency in MSOPS.

7.5 Discussion

These results reveal that Local Recombination, CCG_{TX} , and CCG_{UX} improve the search performance significantly. Also, the effectiveness is emphasized when

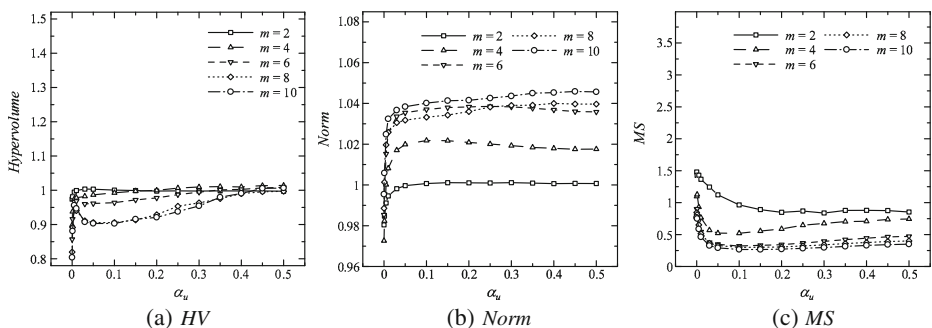


Fig. 17 Performance obtained by SPEA2 [5] with CCG_{UX} ($n = 1,000$)

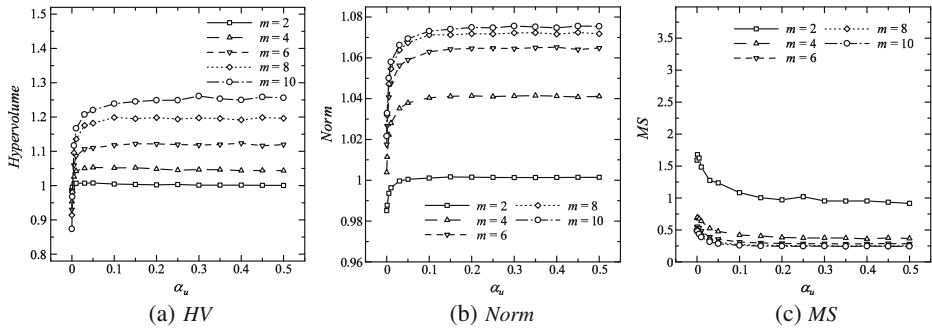


Fig. 18 Performance obtained by IBEA $_{\epsilon+}$ [9] with CCG $_{UX}$ ($n = 1,000$)

we apply these crossover operators to NSGA-II and MSOPS, which evolve well-spread solutions in objective/solution space. Furthermore, the effectiveness of CCG operators is higher than Local Recombination especially for large number of objectives. Additionally, CCG $_{UX}$ achieves higher HV than CCG $_{TX}$ by enhancing the convergence of obtained POS toward Pareto optimal front. This is because CCG $_{UX}$ can control more precisely the number of genes being crossed than CCG $_{TX}$. An important observation is that CCG mainly improves diversity of obtained POS in objective space, and it contributes to the significant improvement of HV in MaOPs. In almost all studies on evolutionary many-objective optimization, the deterioration in the convergence property of MOEA towards the Pareto front has been discussed [6]. However, these results obtained in this work clearly show that the diversity of solutions by the conventional MOEAs is not enough in MaOPs and search performance can be improved by using CCG.

It is difficult to assess the quality of the obtained sets for high dimensional spaces. For this reason, we use several indicators (HV , $Norm$, MS) to have a better idea of the kind of solutions obtained by the algorithms. In addition, we also included parallel coordinate plots as shown in Fig. 20. Looking at the plots, we can see that when recombination that controls the number of crossed genes is included, the solutions found are better spread than when conventional two-point or uniform recombination is used. This is in accordance with values observed for HV and MS . In

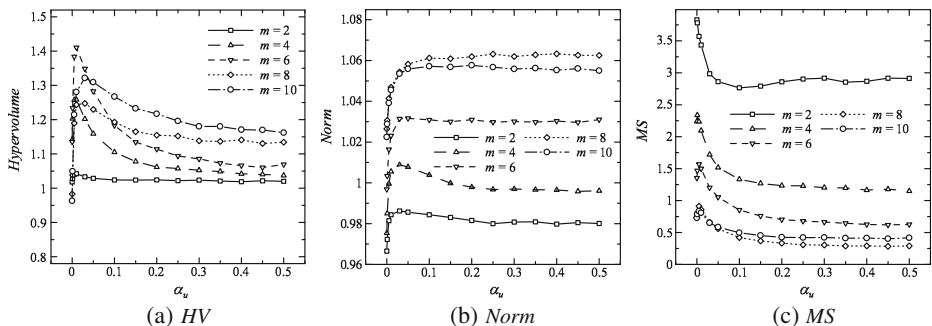


Fig. 19 Performance obtained by MSOPS [2] with CCG $_{UX}$ ($n = 1,000$)

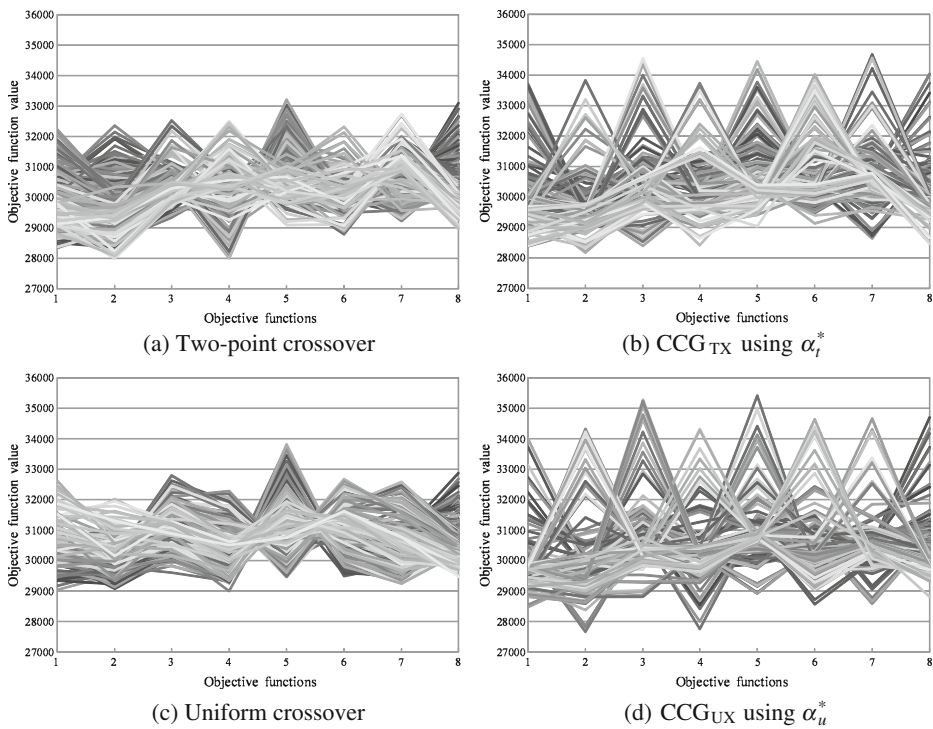


Fig. 20 Parallel coordinate plots of objective function values ($n = 1,000$ items, $m = 8$ objectives)

the parallel coordinate plots we cannot visually determine what set of solutions show better convergence. However, that can be determined by the *Norm* metric. Thus, we can say that solutions obtained by introducing a recombination operator that controls the number of crossed genes are better than those obtained by conventional recombination.

8 Difference of roles between CCG and mutation in solutions search

CCG has two features as genetic operator. First one is that CCG can control the amount of genetic variation by using the parameter α . Another one is that genetic variations are brought by copying genes from another individual in the population. On the other hand, although mutation operator can also control the amount of genetic variation by using the mutation probability P_m , genetic variations are brought randomly without referring genes in the population. To verify the effectiveness of these two different genetic operators on the search performance, in this section we compare the search performance among mutation-only, crossover (CCG)-only and both crossover (CCG) and mutation based search on NSGA-II.

Figure 21a shows results of *HV* obtained by NSGA-II with mutation-only based search as we vary the mutation probability P_m . Similar to the previous sections, values of *HV* are normalized by the value of *HV* obtained by NSGA-II with the

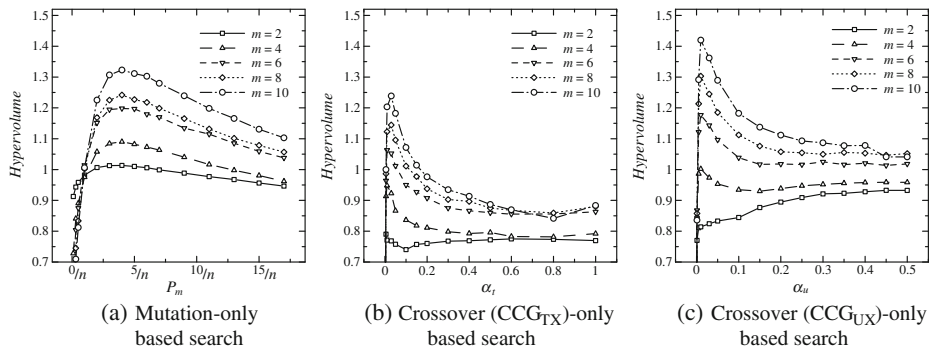


Fig. 21 HV obtained by mutation-only, CCG_{TX}- and CCG_{UX}-only based search ($n = 1,000$)

conventional two-point crossover and mutation ($P_m = 1/n$). From this result, we can see that the maximum HV is obtained when we utilize $P_m^* = 4/n$, and these values become gradually higher than 1.0 by increasing the number of objectives m . That is, an efficient reproduction is performed by restricting the variation of genes using mutation operator compared to solutions search using too disruptive two-point crossover in many-objective problems. Next, Fig. 21b and c show the results of HV obtained by CCG-only based search as we vary α_t and α_u , respectively. From the result of Fig. 21b, we can see that values of HV are lower than 1.0 for all objectives m when we utilize the conventional two-point crossover with $\alpha_t = 1.0$. It reveals that enough search performance cannot be obtained by only the conventional two-point crossover. However, values of HV can be improved by decreasing α_t for $m \geq 4$ objectives, and the maximum HV using the optimal α_t^* becomes higher than 1.0 for $m \geq 6$ objectives. That is, CCG-only based search is able to achieve higher search performance than the solution search using both the conventional two-point crossover and mutation in many-objective problems. However, values of HV obtained by CCG-only based search are lower than mutation based search shown in Fig. 21a. Next, Fig. 21c shows results obtained by CCG_{UX}-only based search. Although we can see the similar tendency to the results obtained by CCG_{TX}, CCG_{UX}-only based search achieves higher HV than mutation-only based search for $m = \{8, 10\}$ objectives problems.

Next, Figs. 22 and 23 show the search performance of mutation-only, crossover (CCG)-only and both based search when we vary the number of objectives m . For mutation-only based search we plot results using P_m^* which maximizing HV shown in Fig. 21a. Also for CCG-only based search we plot results with α_t^* and α_u^* maximizing HV shown in Fig. 21b and c, respectively. For combined search using CCG and mutation, we use α_t^* and α_u^* maximizing HV shown in Figs. 12 and 16 with mutation using $P_m = \{1/n, 4/n\}$.

From results of Fig. 22a, we can see that mutation-only based search achieves higher HV than crossover (CCG_{TX})-only based search for all objectives m . On the other hand, combined search using CCG_{TX} and mutation with $P_m = 1/n$ shows higher HV than mutation-only based search for $m \geq 8$ objectives. Also, combined search using $P_m = 4/n$ achieves higher HV than mutation-only based search for all objectives. Next, we analyze convergence and diversity of obtained POS by using *Norm* and *MS*, respectively. From results of *Norm* shown in Fig. 22b, we can

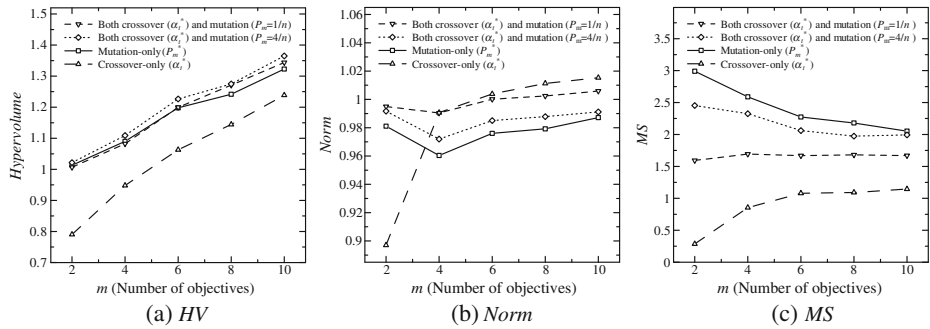


Fig. 22 Performance comparison among mutation-only, crossover (CCG_{TX})-only and both based search ($n = 1,000$)

see crossover (CCG_{TX})-only based search shows the highest convergence toward the true Pareto front except the case of $m = 2$. On the other hand, convergence of mutation-only based search becomes the lowest. Values of *Norm* obtained by combined search using CCG and mutation approaches to values of *Norm* obtained by crossover-only based search by decreasing P_m . Next, from results of *MS* shown in Fig. 22c, the highest diversity is achieved by mutation-only based search for all objectives. On the other hand, crossover (CCG)-only based search shows the lowest diversity. For results obtained by CCG_{UX} shown in Fig. 23, we can see the similar tendency to results shown in Fig. 22. For results of *HV*, combined search using CCG_{UX} and mutation with $P_m = 1/n$ achieves the highest *HV* for all objectives m .

From these results, we can see that mutation operator can improve diversity of obtained POS but convergence becomes scarce. Although mutation-only based search enhances the search performance by using appropriate P_m in many-objectives, the highest *HV* is not obtained by mutation-only. Also, CCG operator can improve convergence of obtained POS but diversity becomes scarce. Although enough search performance is not achieved by only the conventional two-point and uniform crossover, CCG-only based search with a small α improves the search performance especially for many-objectives. Finally, we can see that combined search using CCG

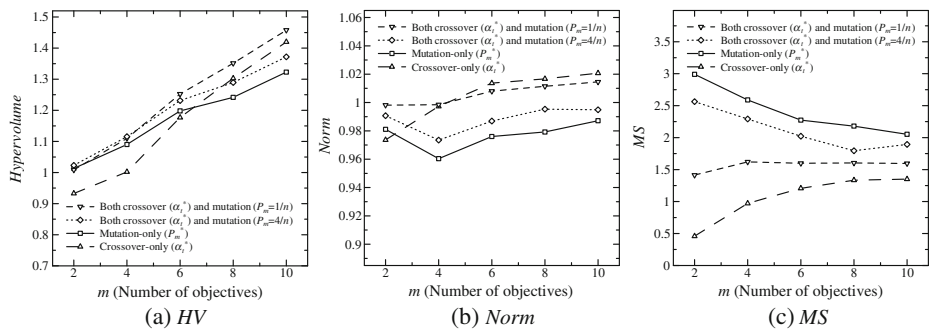


Fig. 23 Performance comparison among mutation-only, crossover (CCG_{UX})-only and both based search ($n = 1,000$)

and mutation achieves the highest HV for all objectives m . These results suggest that it is important to combine CCG and mutation operator to realize well-balanced search between convergence and diversity. CCG and mutation have the common feature which can control the amount of variation, and it improves the search performance especially for many-objectives problems. However, roles in the solution search are quite different. Mutation operator brings diversity by random variation for parent's gene structure. On the other hand, CCG operator brings convergence by copying genes from another individual. To achieve better HV realizing well-balanced search, it is necessary to use both CCG and mutation simultaneously.

9 Analysis of solutions search by using CCG_{TX} and CCG_{UX}

Next, we analyze solutions search on NSGA-II using CCG_{TX} and CCG_{UX} operators. To observe the variation of genes by conventional two-point, uniform crossover and CCG, Fig. 24 shows transition of average hamming distance between offspring and parents on the problem with $m = \{4, 6, 8, 10\}$ objectives and $n = 1,000$ items (bits). To observe hamming distance between an offspring C and a pair of parents P_1 and P_2 , first we calculate hamming distance between C and P_1 and hamming distance between C and P_2 . Then, we adopt the smaller one as hamming distance of the pair of offspring and parents in our analysis. And, Fig. 24 shows average hamming distance of all pair of offspring and parents. For CCG_{TX} and CCG_{UX}, we use the optimal parameter $\alpha_t^* = 0.03$ and $\alpha_u^* = 0.01$. Also we use $P_m = 1/n$ for bit-flipping mutation. When we compare conventional crossovers, we can see that average hamming distance between parent and offspring by uniform crossover is longer than two-point crossover. This is because, although typical uniform crossover exchanges around $n/2$ bits for each crossover, the length of crossed genes vary in the range $[0, n]$ in the conventional two-point crossover. Also, we can see that average hamming distance increases as we increase the number of objectives m when we use two-point and uniform crossovers. On the other hand, by applying CCG operators which controls the number of crossed genes, we can see that average hamming distance between parent and offspring become significantly short and the variation of genes in crossover become small. Average hamming distance by applying CCG_{TX} using $\alpha_t^* = 0.03$ at the final generation becomes $\{2.24, 3.28, 3.57, 4.03\}$ bits for $m = \{4, 6, 8, 10\}$ objectives, respectively. Also, average hamming distance by applying CCG_{UX} using $\alpha_u^* = 0.01$ at the final generation becomes $\{1.63, 2.29, 2.47, 2.79\}$ bits for $m = \{4, 6, 8, 10\}$ objectives, respectively. From these results, we can see that average hamming distance obtained by CCG slightly increases as we increase the number of objectives m . Although CCG can control the length of crossed genes by using parameter α , variation does not occur if two parents have the same gene on a locus of crossed genes. As shown in Fig. 6a, since NSGA-II explores a broad region of variable space and genetic diversity of the population increases as we increase the number of objectives m , actual genetic variations by CCG slightly increase by increasing m even if we use same α .

Next, Fig. 25 shows transition of average hamming distance of solutions in the population when we apply conventional two-point, uniform crossover and CCG operators in NSGA-II. From this result, we can see that the average hamming distance of the population obtained by conventional two-point and uniform crossover

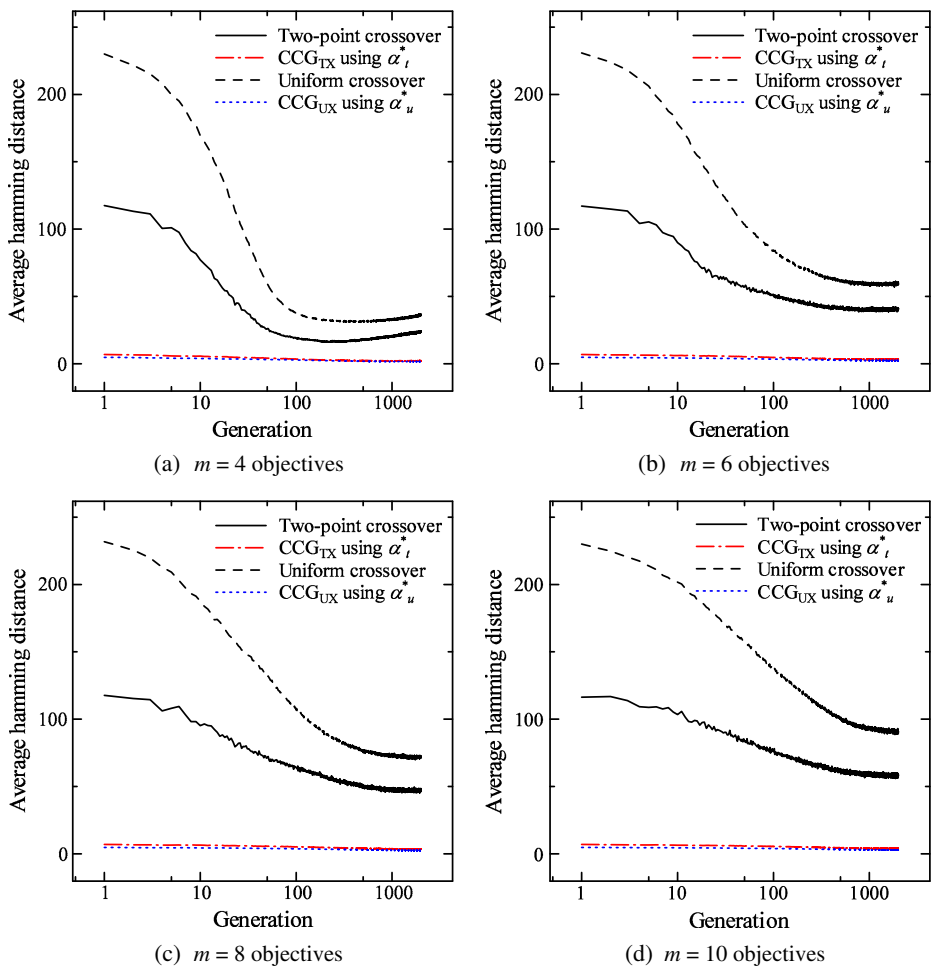
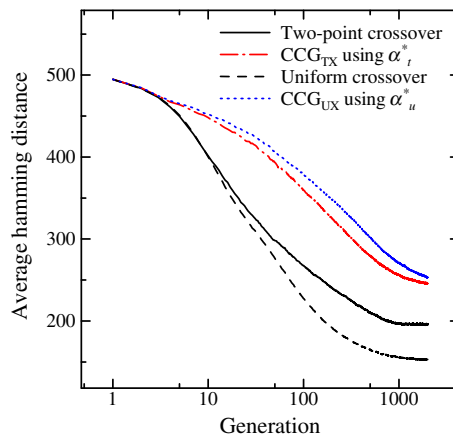


Fig. 24 Transition of average hamming distance between parent and offspring ($n = 1,000$ items)

decreases and deteriorate the genetic diversity in early stage of evolution. On the other hand, the average hamming distance of solutions in the population by CCG operators is larger. From this result, we can see that CCG operators, by decreasing the number of crossed genes, allows NSGA-II to widely search in the solutions space enhancing its search performance.

Next, to analyze the contribution of CCG_{TX} and CCG_{UX} for solution search, we observe the transition of the number of offspring that survived selection and become a part of the parent population P_t . Figure 26 shows the result obtained by NSGA-II using the conventional two-point crossover and CCG_{TX} with optimal α_t^* . Also, Fig. 27 shows the result when we perform typical uniform crossover and CCG_{UX} with optimal α_u^* . From these results, we can see that the number of offspring created by CCG operators that survived selection is larger than conventional two-point and uniform crossover. These results reveal that CCG_{TX} and CCG_{UX} can create successful offspring more often than conventional two-point and uniform crossovers.

Fig. 25 Transition of average hamming distance of solutions in the population ($n = 1,000$ items, $m = 8$ objectives)



In Figs. 26 and 27, we count the number of survived offspring on each population evolved by the conventional crossovers and CCG, respectively. From these results, we can see that the number of survived offspring generated by CCG is higher than the one by the conventional crossovers through the entire solutions search.

Furthermore, we observe the number of survived offspring generated from the same population by the conventional crossovers and CCG. First, we evolve the population by using only conventional two-point crossover. The offspring by conventional two-point crossover are used in actual selection and their survivors are counted. Simultaneously, we generate virtual offspring by using CCG_{TX} from the same parent population. The virtual offspring is used to emulate selection and the number of survivors are counted. Note that since they are not added to the population, they do not affect the evolution by the conventional two-point crossover. Figure 28a shows the number of survived offspring generated by the conventional two-point crossover and virtually generated by CCG_{TX} . From the results, after initial 10 generations, we can see that when the population is evolved by the conventional two-point

Fig. 26 Number of offspring selected in P_t for two-point crossover operators ($n = 1,000$ items, $m = 8$ objectives)

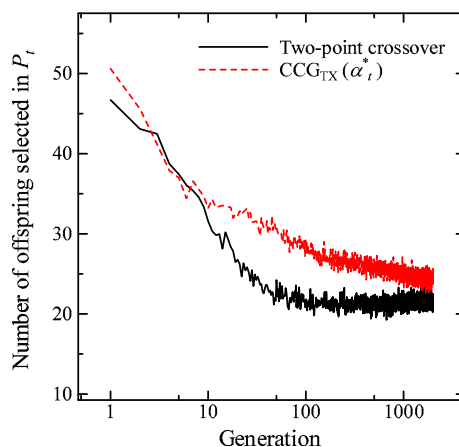
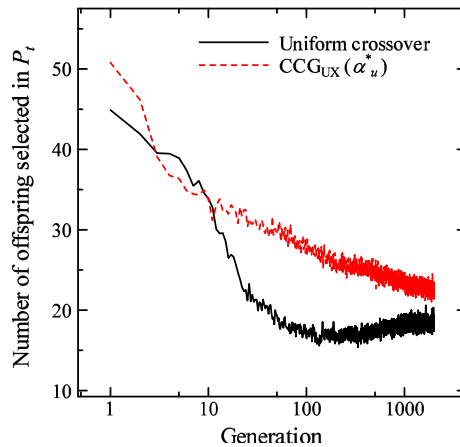
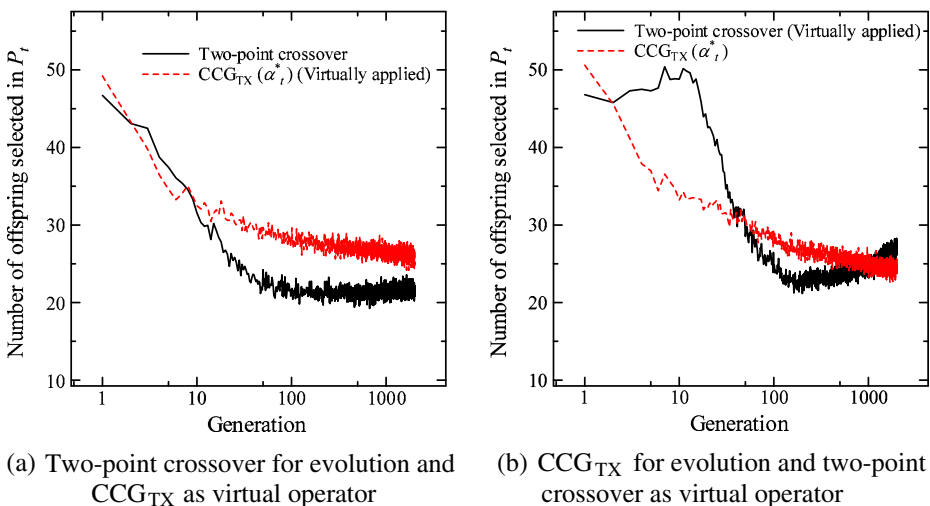


Fig. 27 Number of offspring selected in P_t for uniform crossover operators ($n = 1,000$ items, $m = 8$ objectives)



crossover, offspring generated by CCG_{TX} are more often selected than offspring by conventional two-point crossover.

Next, inversely, we evolve the population by only using CCG_{TX} and simultaneously we generate virtual offspring by conventional two-point crossover from the same population. Figure 28b shows the number of survived offsprings generated by CCG_{TX} and virtually generated by conventional two-point crossover. From the results, we can see that when the population is evolved by CCG_{TX} , offspring generated by the two-point crossover are more often selected than offspring by CCG_{TX} in early stage of evolution. Note that this tendency is more pronounced than in Fig. 28a. These results suggest that there is an optimal length of crossed genes



(a) Two-point crossover for evolution and CCG_{TX} as virtual operator

(b) CCG_{TX} for evolution and two-point crossover as virtual operator

Fig. 28 Number of actual offspring selected in P_t and number of virtual offspring after emulated selection for two-point crossover operators ($n = 1,000$ items, $m = 8$ objectives)

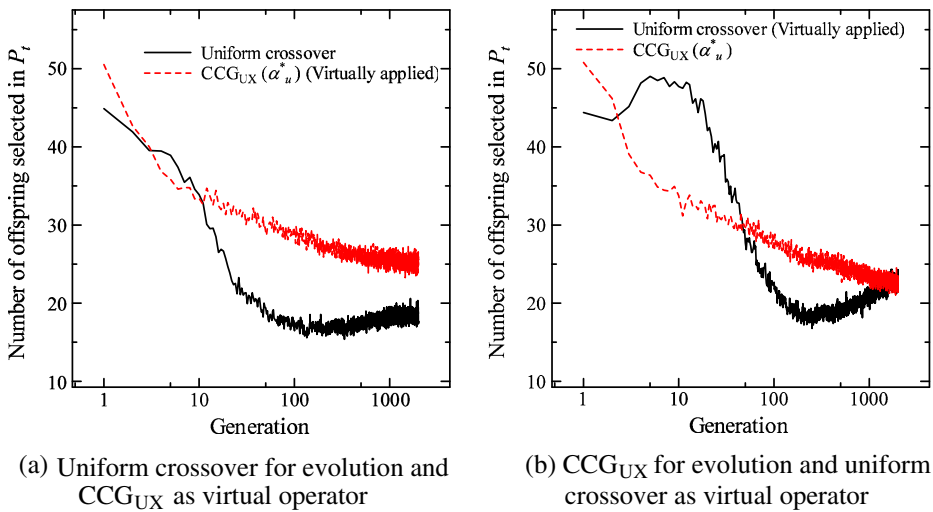


Fig. 29 Number of actual offspring selected in P_t and number of virtual offspring after emulated selection for uniform crossover operators ($n = 1,000$ items, $m = 8$ objectives)

depending on the distribution of the population, and that it dynamically changes during the solution search. This point should be investigated with detail in the future.

Similarly, Fig. 29 shows results obtained by typical uniform crossover and CCG_{UX}. We can see a similar tendency to Fig. 28.

10 Conclusions

In this work, we have analyzed variable space diversity of Pareto optimal solutions in many-objective optimization problems and studied the effectiveness of crossover for many-objective optimization. First, we examined the true Pareto optimal solutions obtained by exhaustive search on many-objective 0/1 knapsack problem with $n = \{10, 15, 20\}$ bits, verifying that the ratio of Pareto optimal solutions in feasible solution space increases with the number of objectives. Also, we observed that genes of Pareto optimal solutions become noticeably diverse, and Pareto optimal solutions come to be distributed almost uniformly in solution space by increasing the number of objectives m . Then, we used NSGA-II, SPEA2, IBEA_{ε+}, and MSPOS, well known multi-objective evolutionary algorithms that adopt different selection methods, to analyze the search performance of conventional recombination, Local Recombination that selects mating parents based on proximity in objective space, and crossover operators which Controls the maximum number of Crossed Genes (CCG). Simulation results on many-objective 0/1 knapsack problems with $m = \{2, 4, 6, 8, 10\}$ objectives revealed that Local Recombination and CCG operators significantly improve search performance, especially for NSGA-II which have high diversity of genes in the population. CCG operators achieve higher search performance than Local Recombination for $m \geq 4$ objectives problems. Also, the effectiveness becomes more significant as the number of objectives m increase. Additionally, we showed the difference of role in the solutions search between CCG

and mutation operators. Furthermore, we showed that offspring created by CCG survive selection more often than conventional crossover.

Our main intention in this work was to show the importance of taking into account a fundamental feature of many-objective landscapes when we design a recombination operator. That is, the dispersion extent of optimal solutions in variable space gets larger with the number of objectives. Recombining very dissimilar individuals, without putting a limit to the information being crossed, causes the operator to become too disruptive and decreases its effectiveness to find better solutions. However, note that our conclusion is obtained from the results on many-objective combinatorial knapsack problems. To show the general tendency on many-objective problems, further research on other many-objective problems should be done.

As future works, analyzing different recombination operators used in other algorithms under the perspective we put forward in this work would be a very interesting and useful line of research. In addition, we should further analyze Pareto optimal solutions on various many-optimization problems to better understand the relationship between variable and objective space and how to feedback that into genetic operators used in the algorithm. Also, we want to study the effect of the proposed CCG in other MOEAs.

References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons (2001)
2. Hughes, E.J.: Evolutionary many-objective optimisation: many once or one many? In: Proc. of 2005 IEEE Congress on Evolutionary Computation (CEC2005), pp. 222–227 (2005)
3. Aguirre, H., Tanaka, K.: Working principles, behavior, and performance of MOEAs on MNK-landscapes. *Eur. J. Oper. Res.* **181**(3), 1670–1690 (2007)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**, 182–197 (2002)
5. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength Pareto evolutionary algorithm. TIK-Report, No.103 (2001)
6. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Evolutionary many-objective optimization: a short review. In: Proc. of 2008 IEEE Congress on Evolutionary Computation (CEC2008), pp. 2424–2431 (2008)
7. Sato, H., Aguirre, H., Tanaka, K.: Self-controlling dominance area of solutions in evolutionary many-objective optimization. In: Proc. of 8th Intl. Conf. on Simulated Evolution and Learning (SEAL2010). LNCS, vol. 6457, pp. 455–465 (2010)
8. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Iterative approach to indicator-based multiobjective optimization. In: Proc. of 2007 IEEE Congress on Evolutionary Computation (CEC2007), pp. 3697–3704 (2007)
9. Zitzler, E., Kunzili, S.: Indicator-based selection in multiobjective search. In: Proc. of 8th Intl. Conf. on Parallel Problem Solving from Nature (PPSN-VIII). LNCS, vol. 3242, pp. 832–842 (2004)
10. Kukkonen, S., Lampinen, J.: Ranking-dominance and many-objective optimization. In: Proc. of 2007 IEEE Congress on Evolutionary Computation (CEC2007), pp. 3983–3990 (2007)
11. Fabre, M., Pulido, G., Coello, C.: Alternative fitness assignment methods for many-objective optimization problems. In: Proc. of 9th International Conference Evolution Artificielle (EA2009). LNCS, vol. 5975, pp. 146–157 (2010)
12. Sato, H., Aguirre, H., Tanaka, K.: Pareto partial dominance MOEA and hybrid archiving strategy included CDAS in many-objective optimization. In: Proc. of 2010 IEEE Congress on Evolutionary Computation (CEC2010), pp. 3720–3727 (2010)

13. Ishibuchi, H., Nojima, Y.: Optimization of scalarizing functions through evolutionary multiobjective optimization. In: Proc. of 4th Intl. Conf. on Evolutionary Multi-Criterion Optimization (EMO2007), pp. 51–65 (2007)
14. Hughes, E.J.: Many-objective directed evolutionary line search. In: Proc. of 2011 Genetic and Evolutionary Computation Conference (GECCO2011), pp. 761–768. ACM Press (2011)
15. Wagner, T., Beume, N., Naujoks, B.: Pareto-, aggregation- and indicator-based methods in many-objective optimization. In: Proc. of 4th Intl. Conf. on Evolutionary Multi-Criterion Optimization (EMO2007), pp. 742–756 (2007)
16. Deb, K., Saxena, K.: Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: Proc. of 2006 IEEE Congress on Evolutionary Computation (CEC2006), pp. 3353–3360 (2006)
17. Lopez Jaimes, A., Coello, C., Chakraborty, D.: Objective reduction using a feature selection technique. In: Proc. of 2008 Genetic and Evolutionary Computation Conference (GECCO2008), pp. 674–680. ACM Press (2008)
18. Brockhoff, D., Zitzler, E.: Objective reduction in evolutionary multiobjective optimization: theory and applications. *Evol. Comput.* **17**(2), 135–166 (2009)
19. Deb, K., Sundar, J.: Preference point based multi-objective optimization using evolutionary algorithms. In: Proc. of 2006 Genetic and Evolutionary Computation Conference (GECCO2006), pp. 635–642 (2006)
20. Lopez Jaimes, A., Montano, A., Coello, C.: Preference incorporation to solve many-objective airfoil design problems. In: Proc. of 2011 IEEE Congress on Evolutionary Computation (CEC2011), pp. 1605–1612 (2011)
21. Auger, A., Bader, J., Brockhoff, D., Zitzler, E.: Articulating user preferences in many-objective problems by sampling the weighted hypervolume. In: Proc. of 2009 Genetic and Evolutionary Computation Conference (GECCO2009), pp. 555–562. ACM Press (2009)
22. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **3**(4), 257–271 (1999)
23. Bentley, J.L., Kung, H.T., Schkolnick, M., Thompson, C.D.: On the average number of maxima in a set of vectors and applications. *J. ACM* **25**(4), 536–543 (1978)
24. Watanabe, S., Hiroyasu, T., Miki, M.: Neighborhood cultivation genetic algorithm for multi-objective optimization problems. In: Proc. of Genetic and Evolutionary Computation Conference (GECCO2002), pp. 458–465 (2002)
25. Sato, H., Aguirre, H., Tanaka, K.: Local dominance and local recombination in MOEAs on 0/1 multiobjective knapsack problems. *Eur. J. Oper. Res.* **181**(3), 1670–1690 (2007)
26. Ishibuchi, H., Shibata, Y.: Mating scheme for controlling the diversity-convergence balance for multiobjective optimization. In: Proc. of 2004 Genetic and Evolutionary Computation Conference (GECCO2004), pp. 1259–1271 (2004)
27. Zhang, Q., Li, H.: MOEA/D: A multi-objective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)
28. Syswerda, G.: Uniform crossover in genetic algorithms. In: Proc. of the 3rd International Conference on Genetic Algorithms (ICGA89), pp. 2–9 (1989)
29. Spears, W., De Jong, K.A.: An Analysis of Multi-Point Crossover. *Foundations of Genetic Algorithms* (1990)
30. Zitzler, E.: Evolutionary algorithms for multiobjective optimization: methods and applications. Ph.D. thesis, Swiss Federal Institute of Technology, Zurich (1999)
31. Kowatari, N., Oyama, A., Aguirre, H., Tanaka, K.: A study on large population MOEA using adaptive ϵ -box dominance and neighborhood recombination for many-objective optimization. In: Proc. of 6th Learning and Intelligent Optimization Conference (LION6) (2012, in USB-memory)
32. Fonseca, C., Paquete, L., López-Ibáñez M.: An improved dimension-sweep algorithm for the hypervolume indicator. In: Proc. of 2006 IEEE Congress on Evolutionary Computation (CEC2006), pp. 1157–1163 (2006)
33. Sato, M., Aguirre, H., Tanaka, K.: Effects of δ -similar elimination and controlled elitism in the NSGA-II multiobjective evolutionary algorithm. In: Proc. of 2006 IEEE Congress on Evolutionary Computation (CEC2006), pp. 3980–398 (2006)