

Enhancing Multiobjective Evolutionary Algorithms by Local Dominance and Local Recombination: Performance Verification in Multiobjective 0/1 Knapsack Problems

HIROYUKI SATO,[†] HERNÁN AGUIRRE[†] and KIYOSHI TANAKA[†]

This paper proposes a method to enhance single population multiobjective evolutionary algorithms (MOEAs) by searching based on local dominance and local recombination. In this method, first, all fitness vectors of individuals are transformed to polar coordinate vectors in objective function space. Then, the population is iteratively divided into several subpopulations by using declination angles. As a result, each sub-population covers a sub-region in the multiobjective space with its individuals located around the same search direction. Next, local dominance is calculated separately for each sub-population after alignment of its principle search direction by rotation. Selection, recombination, and mutation are applied to individuals within each sub-population. The proposed method can improve the performance of MOEAs that use dominance based selection, and can reduce the entire computational cost to calculate dominance among solutions as well. In this paper we verify the effectiveness of the proposed method obtaining Pareto optimal solutions in two representative MOEAs, i.e. NSGA-II and SPEA2, with Multiobjective 0/1 Knapsack Problems.

1. Introduction

Recently, multiobjective evolutionary algorithms (MOEAs)^{1),2)} have been increasingly investigated. MOEAs solve multiobjective optimization problems (MOPs) by using evolutionary algorithms^{3),4)} inspired from biological evolution. MOEAs are particularly suitable to solve MOPs because they evolve simultaneously a population of potential solutions to the problem in hand, which allows us to search a set of Pareto optimal solutions (POS) in a single run of the algorithm.

Two important goals of a MOEA are to achieve POS converging to the true Pareto front and keep a good distribution in objective space of the solutions found. Among the various methods proposed so far^{1),2)}, approaches that use elitism based on dominance are becoming the state of the art. In general, these algorithms are quite effective obtaining POS when the search space is relatively small. However, when the search space becomes large and/or the number of objectives increases, it becomes gradually difficult for them to obtain POS with sufficient diversity in objective space. This is mainly caused by elitism on dominance that gives strong priority to individuals having higher rank of dominance in the selection process. In this way, conventional schemes try to

satisfy the convergence condition of POS but have difficulties satisfying simultaneously diversity requirements. That is, solutions in the population tend to be distributed in a relatively narrow region of the Pareto optimal front.

In order to solve this problem and obtain POS satisfying diversity conditions, we propose a method to enhance single population MOEAs by performing local dominance and local recombination. In this method, we first transform all fitness vectors of individuals to polar coordinate vectors in the objective function space. Then we divide the population into several sub-populations by using declination angles. We calculate local dominance for individuals belonging to each sub-population after alignment of its principle search direction by rotation and apply selection, recombination, and mutation to individuals within each sub-population. The proposed method can be easily applied to MOEAs that use dominance based selection. An additional and important advantage of the proposed method is that it can reduce the entire computational cost to calculate dominance among solutions.

In this paper we pick up NSGA-II⁵⁾ and SPEA2⁶⁾ as two representatives of the latest generation of elitist MOEAs and enhance them with our method. We verify the effectiveness of the proposed method obtaining Pareto optimal solutions satisfying diversity conditions by comparing the search performance between

[†] Faculty of Engineering, Shinshu University

the conventional algorithms and their enhanced versions. Also, we give an estimate of the entire computational cost to calculate dominance.

2. Related Works

Related works to the way we perform local recombination include methods that bias mating to recombine similar parents and methods that group individuals by similar search direction in the objective space.

Recombination of similar parents in MOEAs was initially implemented following the mating restriction suggested by Goldberg⁴⁾ for single objective genetic algorithms, see for example Refs. 7), 8). In these approaches, individuals whose distance is farther apart than a value σ_{mating} are banned from recombination. In general, the parameter σ_{mating} is difficult to set and depends on the problem at hand. The importance of mating restrictions was also emphasized in Refs. 10), 11), where recombination is performed only between individuals next to each other in one of the objectives. Recently, a simple yet effective similarity based-mating scheme to recombine similar parents has been proposed in Ref. 13). In this approach the first parent is selected using binary tournament selection. Then, other β individuals are selected using binary tournament selection and the individual closer to the first parent is selected as its mate. This approach has been extended successfully to recombine extreme and similar parents^{14),15)}. All these methods that restrict mating have been incorporated in algorithms that bias selection by calculating dominance in the whole population. The approach we propose, in addition to local recombination, is used together with local dominance.

Other way to induce local recombination is achieved by temporarily grouping the population around a search direction^{9),16)~19)}. Besides local recombination, an additional and important objective of these methods is to assign uniformly the search effort towards all search directions trying to avoid that the algorithm focuses only in a narrow region of the objective space. However, the methods proposed so far have been based on aggregation approaches, which use the weighted sum of m objectives as a fitness function and specify weighting coefficients randomly whenever a pair of parent individuals are selected. These methods have shown good performance in several applications but are expected to face difficulties on problems

with non-convex Pareto fronts. In contrast, the method we propose is based on dominance and would not suffer from the limited applicability of weighting based methods.

Regarding dominance, most methods that use Pareto based selection calculate dominance globally using a population that could be distributed along the whole search space. There are methods that restrict dominance to individuals located in a neighborhood, see for example Ref. 20), but there are no previous references to methods that calculate local dominance after alignment of search direction by rotation.

Other related work can be found in approaches that use a distributed GA to bias the search of each sub-population to a given region of objective space, aiming to obtain fully distributed POS. See for example Refs. 20), 21). The method we propose is an alternative to single population MOEAs and at this time no special considerations are being taken for a parallel distributed implementation. Nonetheless, we include as a reference results by Distributed Computation of Pareto Optimal Solutions²¹⁾ (DCPOS) to show the effectiveness of the proposed approach.

3. Solving MOP Using MOEAs

A MOP 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, $\mathbf{x} \in \mathcal{F}$ 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 MOP are dominance, Pareto optimality, Pareto set and Pareto front. We define *dominance* between solutions $\mathbf{x}, \mathbf{y} \in \mathcal{F}$ as follows: If

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

are satisfied, \mathbf{x} dominates \mathbf{y} . In the following, \mathbf{x} dominates \mathbf{y} is denoted by $\mathbf{f}(\mathbf{x}) \succeq \mathbf{f}(\mathbf{y})$. 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

$$\mathcal{P} = \{ \mathbf{x} \in \mathcal{F} \mid \neg \exists \mathbf{y} \in \mathcal{F} : \mathbf{f}(\mathbf{y}) \succeq \mathbf{f}(\mathbf{x}) \}, \tag{3}$$

and the Pareto front is defined as

$$\mathcal{PF} = \{ (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \mid \mathbf{x} \in \mathcal{P} \}. \tag{4}$$

MOEAs evolve a population to obtain \mathcal{P} for a given MOP in a single run.

4. Proposed Method

4.1 Concept

Dominance offers important advantages to multiobjective algorithms. It helps to push the search towards higher fronts and is effective for problems with convex and non-convex fronts. However, some global non-dominated solutions may have a too strong influence and may undermine the contribution of other solutions that, although globally dominated, have the potential to make the entire population diverse in objective space. In other words, a solution may dominate a broad region but may not be the best point from which to reach other not yet found non-dominated solutions. Rather, we consider that some individuals that appear globally dominated may be a good source to populate underrepresented regions. As illustrated in **Fig. 1**, solutions that appear globally dominated may be ignored by selection (especially if elitism is used) although it could be worth trying to populate insufficiently represented regions from them. Another concern with dominance is the computational cost required for its calculation in the entire population.

From this point of view, in order to introduce necessary diversity and accomplish efficient search for POS, we divide the entire pop-

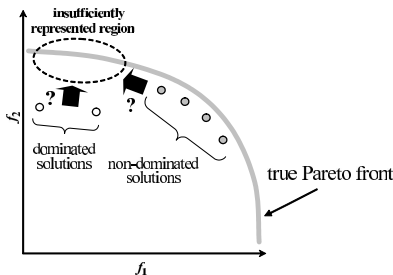


Fig. 1 Dominated solutions could help populate unrepresented optimal regions.

ulation into several sub-populations generation by generation. Each sub-population consists of individuals having similar search directions. Then, we calculate local dominance among individuals in each sub-population. Next, we apply parent selection and genetic operations to individuals within each sub-population by reflecting local dominance. Population division allows us to reduce the entire computational cost while obtaining dispersed POS. The significant points of this work are the population division in objective space using polar coordinates and the way of calculation of local dominance with alignment of the principle search direction by rotation. After calculation of local dominance, we can simply apply conventional MOEAs^{5),6)} in each sub-population. The main loop of the enhanced MOEAs with the proposed method is illustrated in **Fig. 2**. In the following we detail the procedures for population division and local dominance.

4.2 Population Division in Objective Function Space

The objective of the population division is to group individuals with similar search direction in the m -dimensional objective space. To achieve this efficiently, the m -dimensional fitness vector $\mathbf{f}(\mathbf{x})$ for each individual is expressed in polar coordinates by a norm r and $m-1$ declination angles $\theta_j (j = 1, 2, \dots, m-1)$ as shown in **Fig. 3**. First, we create a minimum fitness vector $(f_1^{\min}, f_2^{\min}, \dots, f_m^{\min})$ consisting of the minimum fitness values in each objective function $f_i (i = 1, 2, \dots, m)$ in the population, and calculate temporal fitness vectors for all individuals as $\mathbf{f}'(\mathbf{x}) = (f'_1(\mathbf{x}), f'_2(\mathbf{x}), \dots, f'_m(\mathbf{x})) = (f_1(\mathbf{x}) - f_1^{\min}, f_2(\mathbf{x}) - f_2^{\min}, \dots, f_m(\mathbf{x}) - f_m^{\min})$. Note that in this paper the prime character (') denotes the result after a transformation, and

Procedure enhanced MOEA

```

for  $t \leftarrow 1 \dots T$ 
     $\{P_1, \dots, P_k, \dots, P_{d^{m-1}}\} \leftarrow$  population division ( $P(t), d, m$ )
    for  $k \leftarrow 1 \dots d^{m-1}$ 
        ranking by local dominance ( $P_k$ )
         $Q_k \leftarrow$  truncation ( $P_k$ )
         $R_k \leftarrow$  mating and reproduction ( $Q_k$ )
        fitness evaluation ( $R_k$ )
    end
    /*  $Q(t) = \cup_{k=1}^{d^{m-1}} Q_k, R(t) = \cup_{k=1}^{d^{m-1}} R_k$  */
     $P(t+1) \leftarrow Q(t) \cup R(t)$ 
end
    
```

Fig. 2 Main loop of enhanced MOEA.

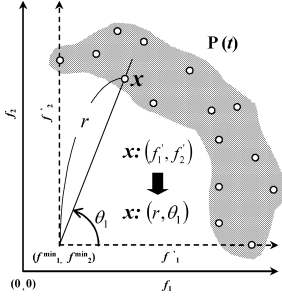


Fig. 3 Solution expressed as polar coordinate vector, $m = 2$.

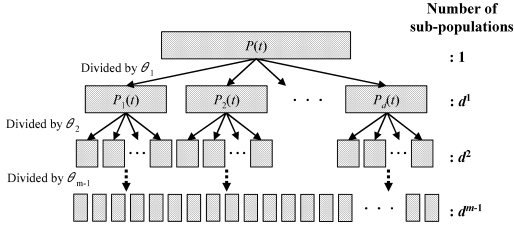


Fig. 4 Population division using declination angles $(\theta_1, \theta_2, \dots, \theta_{m-1})$.

not derivative. Then we transform all temporal fitness vectors $\mathbf{f}'(\mathbf{x})$ to polar coordinate vectors $\mathbf{p}(\mathbf{x})$, i.e., $\mathbf{f}'(\mathbf{x}) = (f'_1(\mathbf{x}), f'_2(\mathbf{x}), \dots, f'_m(\mathbf{x})) \rightarrow \mathbf{p}(\mathbf{x}) = (r(\mathbf{x}), \theta_1(\mathbf{x}), \theta_2(\mathbf{x}), \dots, \theta_{m-1}(\mathbf{x}))$.

Next, the joined population of parents and offspring $P(t)$ is split iteratively according to declination angles into d^{m-1} sub-populations, where d is a parameter indicating the division factor at each one of the $m-1$ iterations. In other words, $P(t)$ is split into d sub-populations according to θ_1 , then each one of these d sub-populations is split again into other d sub-populations according to θ_2 , and so on. The iterative population division method to divide the entire population $P(t)$ at t -th generation into d^{m-1} sub-populations $P_k(t)$ ($k = 1, 2, \dots, d^{m-1}$) is illustrated in **Fig. 4** and the main steps of the algorithm are as follows.

- step1*: Set $l = 1$ and $P_1 = P(t)$.
- step2*: Sort all individuals in P_k ($k = 1, 2, \dots, d^{l-1}$) with l -th angle information θ_l .
- step3*: Divide each P_k ($k = 1, 2, \dots, d^{l-1}$) into d sub-populations in order of angle information θ_l .
- step4*: Increment l . If $l \leq m-1$, repeat *step2* ~ *step3* for sub-populations already obtained.

In *step3* of the above procedure, we use a method that slightly varies sub-population size to avoid gaps in objective space among sub-

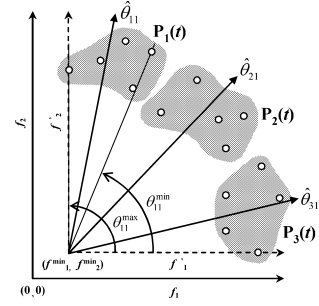


Fig. 5 Determination of principle search direction for each sub-population ($m = 2, d = 3$).

populations. See Section 4.5 for further details.

4.3 Calculation of Local Dominance in Sub-population

Local dominance among individuals in each sub-population P_k ($k = 1, 2, \dots, d^{m-1}$) is calculated after rotating the principle search direction of P_k . The algorithm of this procedure is as follows.

- step1*: Find maximum and minimum declination angles, θ_{kj}^{\max} and θ_{kj}^{\min} ($j = 1, 2, \dots, m-1$), in $P_k(t)$, and determine the principle search direction as shown in **Fig. 5** by

$$\hat{\theta}_{kj} = \frac{\theta_{kj}^{\max} - \theta_{kj}^{\min}}{2} + \theta_{kj}^{\min} \quad (j = 1, 2, \dots, m-1). \quad (5)$$

- step2*: Calculate $m-1$ rotation angles

$$\psi_{kj} = \hat{\theta}_{kj} - \frac{\pi}{4} \quad (j = 1, 2, \dots, m-1). \quad (6)$$

- step3*: Rotate all declination angles of the polar coordinate vectors of individuals in $P_k(t)$ as shown in **Fig. 6** by

$$\begin{aligned} \mathbf{p}'(\mathbf{x}) &= (r_k(\mathbf{x}), \theta'_{k1}(\mathbf{x}), \theta'_{k2}(\mathbf{x}), \dots, \theta'_{km-1}(\mathbf{x})) \\ &= (r_k(\mathbf{x}), \theta_{k1}(\mathbf{x}) - \psi_{k1}, \theta_{k2}(\mathbf{x}) - \psi_{k2}, \dots, \theta_{km-1}(\mathbf{x}) - \psi_{km-1}). \end{aligned} \quad (7)$$

- step4*: Transform all polar coordinates vectors changed in $P_k(t)$ to modified temporal fitness vectors as

$$\begin{aligned} \mathbf{p}'(\mathbf{x}) &= (r(\mathbf{x}), \theta'_{k1}(\mathbf{x}), \theta'_{k2}(\mathbf{x}), \dots, \\ &\quad \theta'_{km-1}(\mathbf{x})) \rightarrow \\ \mathbf{f}''(\mathbf{x}) &= (f''_{k1}(\mathbf{x}), f''_{k2}(\mathbf{x}), \dots, f''_{km}(\mathbf{x})). \end{aligned} \quad (8)$$

- step5*: Calculate local dominance using the modified fitness vectors $\mathbf{f}''(\mathbf{x})$ in $P_k(t)$.

4.4 Local Dominance and Local Recombination

Recalculation of fitness vectors for indi-

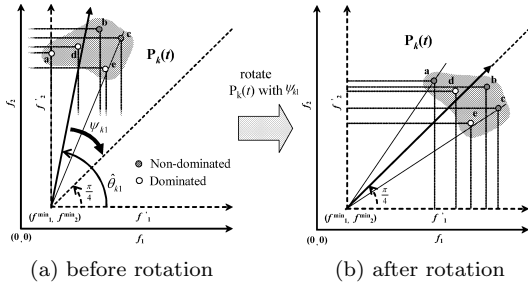


Fig. 6 Rotation of sub-population $P_k(t)$ and its affection to dominance among solutions.

viduals in each sub-population after rotation changes dominance among solutions in objective function space. This brings more selective advantages to become parent individuals to solutions having the potential to spread the search rather than conventional schemes. As shown in Fig. 6 (a), if we calculate dominance among solutions with a conventional scheme, say NSGA-II⁵⁾, individuals *a*, *d* and *e* would be dismissed with high probability in the parent selection process since they are dominated by *b* and *c*. On the other hand, if we take into account the principle search direction of P_k and properly rotate declination angles, as shown in Fig. 6 (b), the individual *a* becomes a non-dominated solution, which is expected to make the entire population spread. In this example, *a* has the potential to disperse the distribution of P_k to the direction of objective function f_2 .

Local dominance is reflected in parent selection within the current sub-population. We apply crossover and mutation operators to parent individuals selected within each sub-population based on local dominance. Because the individuals included in each sub-population have similar search direction in the objective function space, the enhanced algorithm locally achieves recombination between individuals having similar fitness vector. This effectively works to avoid inefficient recombination in MOEAs.

4.5 Gap Suppression by Varying Sub-population Size

If we evolve the entire population with a fixed number of individuals in each sub-population, the search tends to produce gaps among sub-populations in the objective space as illustrated in Fig. 7 (a). In order to suppress this phenomenon, we incorporate a particular population dividing scheme that slightly varies the size of sub-populations, which fluctuates the regions covered by the sub-populations in the objective

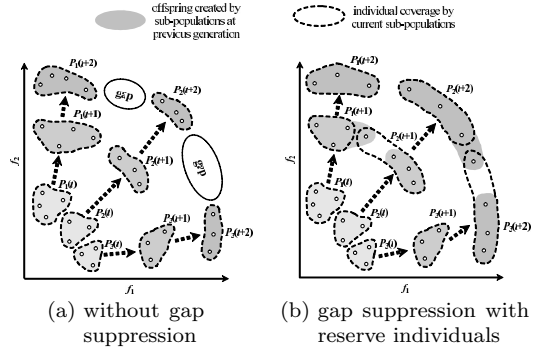


Fig. 7 Illustration of gap suppression among sub-populations by reserve individuals ($|P| = 10$, $\mu = 3$, $\pi = (1, 0, 0)$, $d = 3$).

space. Suppose that a population P is divided into K sub-populations P_k ($k = 1, 2, \dots, K$). In this process, we describe the total population size in P as

$$|P| = \sum_{k=1}^K |P_k| = \sum_{k=1}^K (\mu + \pi_k) \quad (9)$$

where $|\cdot|$ means number of individuals in the population. The size of sub-population $|P_k|$ ($k = 1, 2, \dots, K$) is determined by an equivalent number of individuals μ plus a number π_k of reserve individuals, which are randomly assigned to sub-populations. We prepare in advance a vector $\pi = (\pi_1, \pi_2, \dots, \pi_K)$ containing the number of reserve individuals, and randomly permute its elements at every generation. We show an example of this population division in Fig. 7 (b) in case of $|P| = 10$, $\mu = 3$, $\pi = (1, 0, 0)$ and $d = 3$. In this example, we allocate one reserve individual ($\pi_1 = 1$) to P_1 at t -th generation, thus $|P_1| = 4$, $|P_2| = 3$ and $|P_3| = 3$, respectively. But we allocate the reserve individual to P_2 at $(t + 1)$ -th generation, to P_3 at $(t + 2)$ -th generation, and so on. From this illustration we can see that a part of the offspring created by a sub-population can be covered (included) by a different neighbor sub-population. This slight fluctuation of the solution coverage with sub-populations works to suppress gaps in the objective space effectively. Although this idea is somehow similar to immigration among sub-populations in Distributed EAs, it differs in two points. It fluctuates the size of sub-populations in the process of sorting with declination angles in polar coordinate vectors. Eventually, it switches belongingness of just neighbor individuals between sub-populations.

5. Computational Cost Reduction

Recent MOEAs using dominance generally require the computation of order $O(mN^2)$, where m and N denote number of objectives and population size, respectively.

The proposed method can reduce substantially the computational cost to calculate dominance since we divide the entire population into d^{m-1} sub-populations. The computation order for calculating local dominance in all sub-populations by the proposed scheme is $O\left(\frac{mN^2}{d^{m-1}}\right)$ and the computation order for population division is $O(mN \log_2 N)$, where sorting by angle information is used. Thus, the overall computation order of the proposed method is given by $O\left(\frac{mN^2}{d^{m-1}}\right) + O(mN \log_2 N) < O(mN^2)$ for $m \geq 2$ and $d \geq 2$. This could be a big advantage of the proposed method as population size and/or number of objectives increase. Details of the computational cost are given in Appendix A.

6. Problems, Parameters, and Metrics

In this paper we use multiobjective 0/1 knapsack problems¹²⁾ as benchmark problems to verify the search performance of the proposed method. A KP n - m problem of n objects and m knapsacks is formulated to maximize m functions

$$f_j(\mathbf{x}) = \sum_{i=1}^n x_i \cdot p_{i,j} \quad j = (1, 2, \dots, m) \quad (10)$$

subject to

$$g_j(\mathbf{x}) = \sum_{i=1}^n x_i \cdot w_{i,j} \leq W_j \\ j = (1, 2, \dots, m) \quad (11)$$

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. 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 no satisfying this condition are considered as infeasible solutions $\mathcal{F} = (\mathcal{S} - \mathcal{F})$. In this paper, we use benchmark problems with $m = \{2, 3\}$ objectives downloaded from²²⁾, for which we know the true POS only in case of two objectives $m = 2$.

We adopt two-point crossover with a crossover probability $p_c = 1.0$ for recombination, and apply bit-flipping mutation with a

mutation probability $p_m = 1/n$. In the following experiments, we show the average performance with 30 runs, each of which spent 2,000 generations. Population sizes are set to $|P| = \{200, 600\}$ for $m = \{2, 3\}$ objectives, respectively.

In this paper, we use several metrics to evaluate MOEAs performance. The first metric is hyper-volume (HV), which measures a m -dimensional volume covered by POS in objective function space²³⁾. Here we use $(f_1, f_2, \dots, f_m) = (0, 0, \dots, 0)$ as the reference point to calculate HV . POS showing higher HV can be considered as better POS from both convergence and diversity viewpoints. The second metric is generational distance GD proposed by Veldhuizen²⁴⁾, which measures a degree of convergence to the true POS. In order to more precisely compare the obtained POS with the true POS, we also use inverse generational distance IGD originally proposed by Czyzak and Jaskiewicz²⁵⁾ as the $D1_R$ measure. GD measures the average distance for all members in the obtained POS to their nearest solutions in the true POS, while IGD measures the average distance for all members in the true POS to their nearest solutions in the obtained POS. POS showing smaller GD and IGD can be considered as better POS satisfying convergence condition. However, note that IGD gives a small value only if all members of the obtained POS dispersively converges to all members of the true POS, while GD becomes small even if they converge to some of the members in the true POS. The third metric is spread (SP) proposed by Deb et al.¹⁾, which measures the degree of dispersion on the distribution of POS. POS showing smaller SP can be considered as better POS satisfying diversity condition.

7. Experimental Results and Discussion

7.1 Performance Enhancement from Conventional Single Population MOEAs

In this section, we compare the search performance of conventional NSGA-II and SPEA2 and their enhanced versions incorporating the proposed method.

Here we adopt these sizes to observe as much as possible the behavior of the proposed method in a similar range of sub-population sizes while using different number of sub-populations

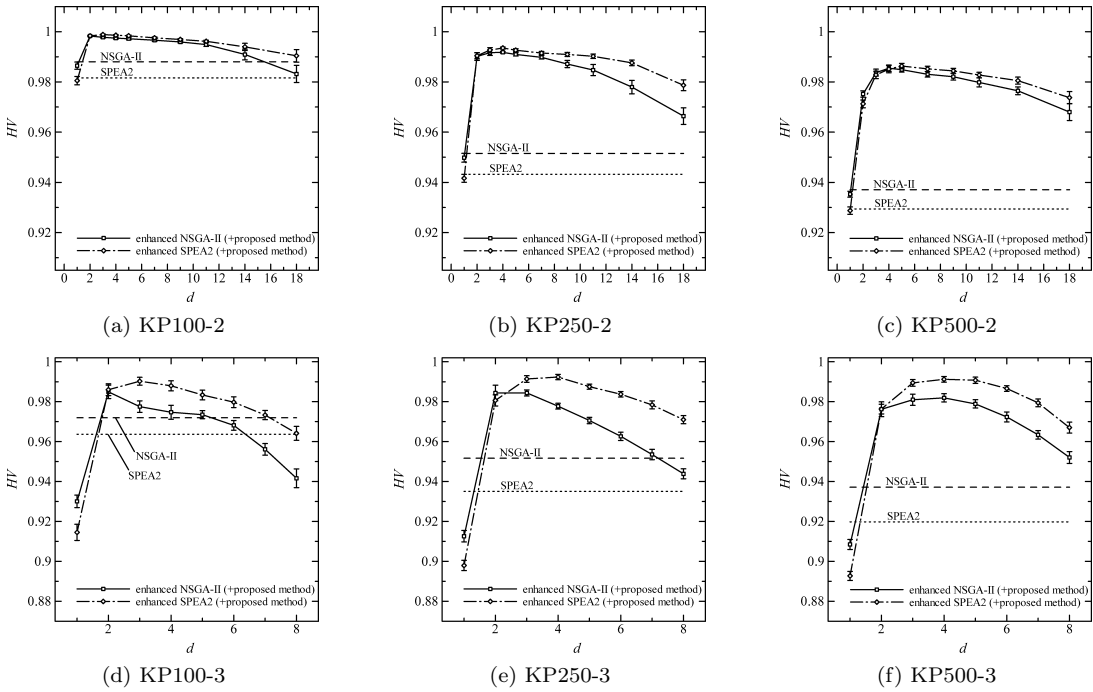


Fig. 8 Performance comparison on hyper-volume (HV).

7.1.1 Hypervolume

First, we show in Fig. 8 the obtained HV by the enhanced MOEAs over the parameter d used for population division. Note that in the case of m objectives, the size of sub-population is given by $|P_k| \sim N/d^{m-1}$. The two parallel dashed lines are the results by conventional NSGA-II and SPEA2. Here we depicted the normalized HV for obtained POS with regard to HV for the true POS in case of two objectives ($m = 2$), while the normalized one for the maximum HV experimentally obtained in case of three objectives ($m = 3$) since we do not know the true POS. The vertical bars, overlaying the mean curves, represent 95% confidence intervals. The main conclusions from Fig. 8 are as follows: (i) As a general tendency, the superiority of the proposed method becomes remarkably large as we increase the solution space by increasing the number of objectives (knapsacks) m and/or the number of items n in the problem. In other words, the proposed method increases its effectiveness as the dynamic range of POS in the objective space becomes large (dispersive). This means that it becomes difficult for a single population MOEA to cover POS widely spread in the objective space. (ii) There is an optimum parameter d^* to maximize HV depending on benchmark problem and algorithm to be

used. For example, $d^* = 3$ ($|P_k| \sim 66$) for enhanced SPEA2 while $d^* = 2$ ($|P_k| \sim 150$) for enhanced NSGA-II for test problem KP100-3, respectively. If we increase d excessively, the performance is gradually deteriorated because the algorithm searches with many but very small sub-populations, which leads to unstable performance with larger variance. Also, for $m = 3$ objectives, we can see performance deterioration in case of $d = 1$, because in this special case the principle search direction of the proposed scheme is not always coincident with $\pi/4$ as the conventional scheme. (iii) Larger improvement by our method can be observed in case of SPEA2 rather than NSGA-II, while conventional NSGA-II always outperforms SPEA2 in these problems.

7.1.2 Convergence and Diversity

Second, we observe the performance separately on convergence and diversity by using optimum parameter d^* for test problems KP100-2, KP250-2, and KP500-2 for which we know the true POS. Figure 9 shows the Pareto front of the obtained POS. In this figure we can see that as the solution space increases from 2^{100} to 2^{500} , both enhanced NSGA-II and SPEA2 implementing our method achieve robust performance obtaining fully dispersed POS close to the true POS. On the other hand, the range

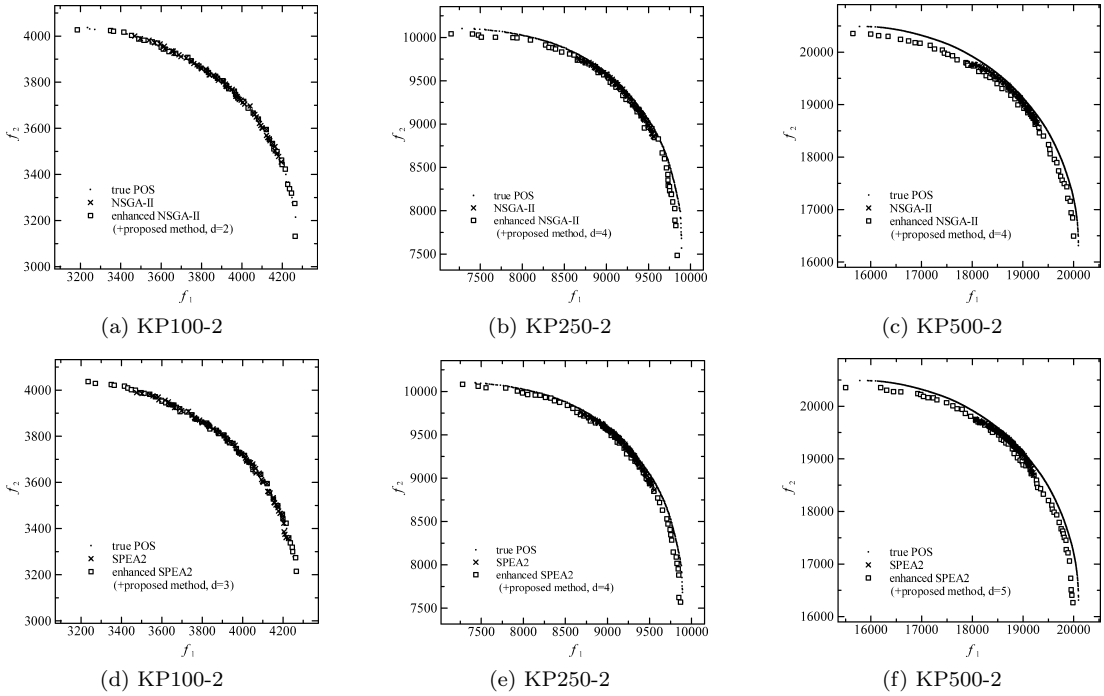


Fig. 9 Obtained POS for KPn-2 ($m = 2$).

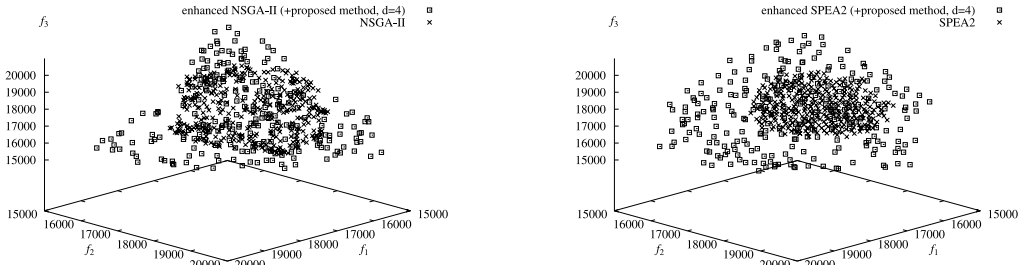


Fig. 10 Obtained POS for KP500-3 ($m = 3$).

of the obtained POS by conventional NSGA-II and SPEA2 becomes narrow for the entire distribution of the true POS. This trend is preserved in case of three objectives ($m = 3$) as shown in Fig. 10, where we present the Pareto front of obtained POS as a bird's-eye view only for KP500-3.

Next, we show in Fig. 11 and Fig. 12 the transition of GD and IGD over the generations as indicators of convergence of POS, respectively. Conventional NSGA-II and SPEA2 achieve smaller GD than their enhanced versions because conventional methods tend to incline the search to the direction of a part of the true POS, which advantageously works to reduce GD . On the other hand, enhanced NSGA-II and SPEA2 achieve clearly smaller IGD than conventional ones. This is because the en-

hanced methods evolve the search dispersively inducing a necessary diversity in the entire population. Population division and calculation of local dominance within sub-populations are quite effective to keep the search dispersive in the objective space.

Furthermore, we show the transition of SP over the generations as an indicator of diversity of POS in Fig. 13. From this figure, we can see that initially SP increases substantially in conventional NSGA-II and SPEA2 for all problems indicating that these algorithms remarkably lose diversity in the early stage of evolution. On the other hand, the enhanced methods continuously induce diversity into the entire population from the beginning of evolution. Precisely, enhanced SPEA2 always achieved smaller SP than enhanced NSGA-II, which supports the

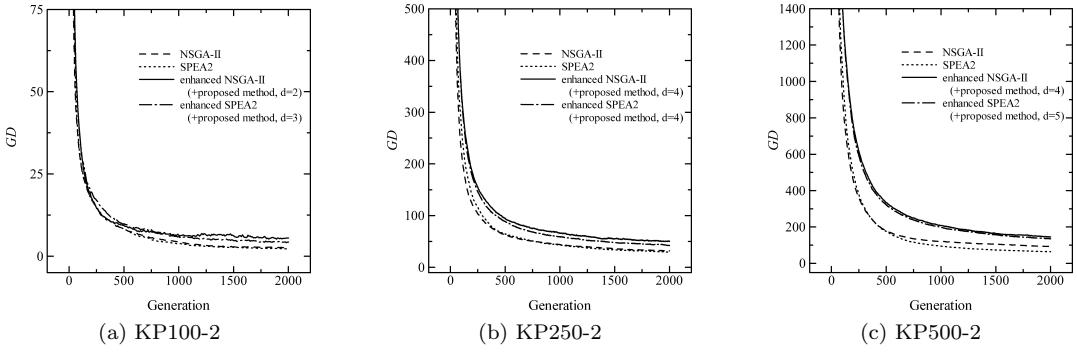


Fig. 11 Transition of *GD* over generations.

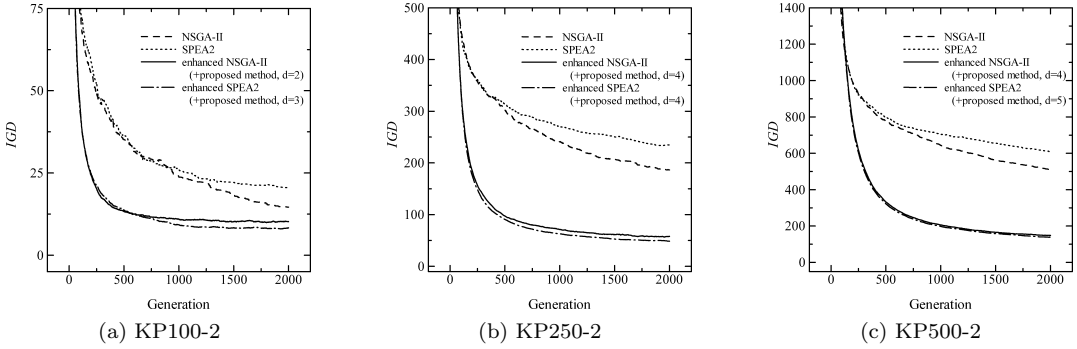


Fig. 12 Transition of *IGD* over generations.

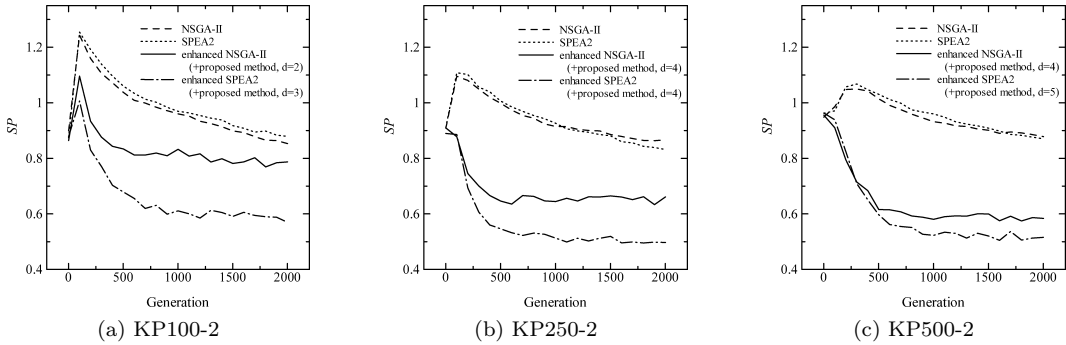


Fig. 13 Transition of *SP* over generations.

result that the former algorithm shows larger improvement on *HV*.

These results illustrate the difficulty a single population MOEA faces to cover widely spread POS in the objective space and also show the effectiveness of the proposed method based on local dominance and local recombination.

7.2 Detailed Observation of Proposed Method

In this section, we give detailed observation of the proposed method from several viewpoints.

7.2.1 Change of Dominance Status

First, we observe the change of dominance status by the proposed method. Our scheme rotates the principle search direction for each sub-population in order to calculate local dominance. As a consequence, globally dominated solutions might become locally non-dominated. Similarly, globally non-dominated solutions might become locally dominated solutions. **Figures 14** and **15** illustrate the ratios of the number of solutions that change their dominance status for representative problems,

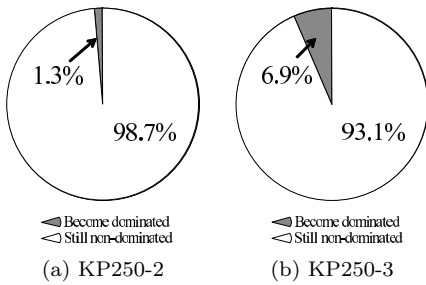


Fig. 14 Status change of globally non-dominated solutions.

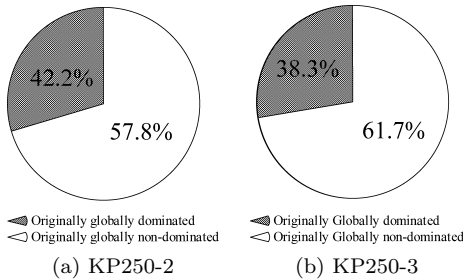


Fig. 15 Origin of locally non-dominated solutions.

KP250-2 and KP250-3. These ratios are the average fractions on 30 runs of the total number of solutions for each category counted from the first to the last generation.

From Fig. 14 (a) we can see that in $m = 2$ objectives only 1.3% of all globally non-dominated solutions become locally dominated after rotation. This percentage increases to 6.7% in case of $m = 3$ objectives, as shown Fig. 14 (b). On the other hand, from Fig. 15 (a) we can see that in $m = 2$ objectives 42.2% of all locally non-dominated solutions are originally globally dominated. This percentage slightly decreases to 38.3% in case of $m = 3$ objectives.

These results show that the negative effect of losing globally non-dominated solutions after local dominance is quite small. On the contrary, there are significant high percentages of solutions originally globally dominated that become locally non-dominated, which help spreading the search of MOEA.

7.2.2 Local Dominance without/with Rotation of Principle Search Direction

In the proposed method, we calculate dominance for each sub-population after rotating the principle search direction. Since the individuals in the sub-population cover a sub-region in objective space forming a neighborhood we call this kind of dominance as *local*. In addition, ro-

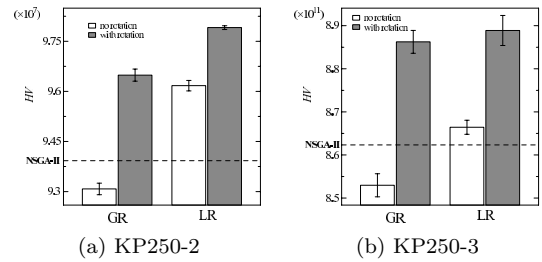


Fig. 16 LocalDominance without/with rotation.

tation is important to reassign non-dominance ranking in order to increase the chances of solutions that can spread the search around the given search direction. The simplest way to calculate local dominance is without performing rotation, in which case solutions outside the sub-population will not influence domination ranking but also no special measure is taken to give more chances to solutions that can spread the search. In this section we especially focus on performance of local dominance without and with rotation of the principle search direction.

Figure 16 (a) and (b) show bar diagrams of the HV obtained by a NSGA-II based algorithm on problems with $m = 2$ and $m = 3$ objectives and $n = 250$ objects. In all cases, the algorithms perform local dominance either without or with rotation. In addition, in order to clearly distinguish the effect of rotation in local dominance, we present results when the algorithm does local recombination (LR) within each sub-population or global recombination (GR) considering the whole population. As a reference for comparison, results by conventional NSGA-II are shown as a horizontal dashed line as well.

From these figures we can see that local dominance after rotation of the principle search direction achieves a much better HV than without rotation in both $m = 2$ and $m = 3$ objectives, whether recombination is applied globally or locally. A second observation is that the HV reduces significantly, becoming lower than conventional NSGA-II, if local dominance without rotation is combined with global recombination. This is because local dominance without rotation blindly weakens selection pressure. A third important observation is that the inclusion of local recombination increases the HV but the effect of local dominance with rotation is higher than local recombination.

From these observations, we conclude that

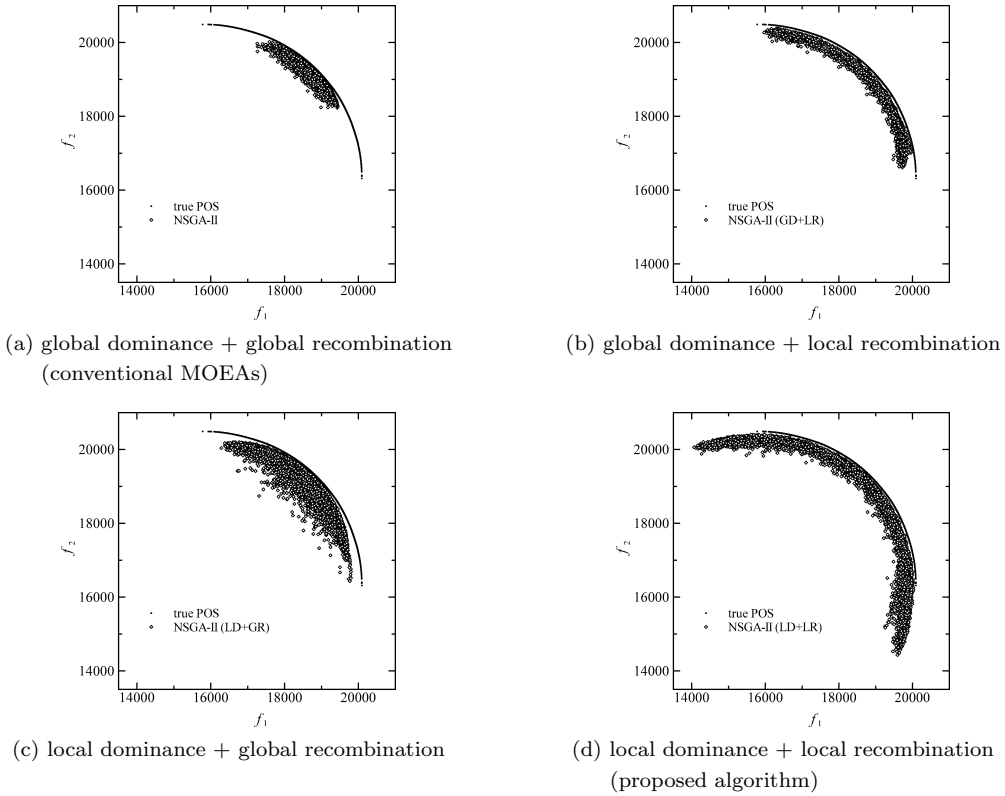


Fig. 17 All solutions of the entire population at the final generation in the objective function space (KP500-2).

the alignment of the principle search direction by rotation is quite important within local dominance. Note that in the following *local dominance* in the proposed method always includes rotation.

7.2.3 Effect of Local Dominance and Local Recombination

Here we observe the effect of local dominance (with rotation) and local recombination separately and verify whether their combined effect aggregate in a positive manner.

Figure 17 plot all solutions of the population at the final generation in the objective function space. We show results by four algorithms, a conventional NSGA-II that applies dominance and recombination globally using the whole population, a NSGA-II (GD+LR) that applies global dominance but uses population division to apply recombination locally, a NSGA-II (LD+GR) that uses population division to calculate dominance locally within each sub-population but applies recombination globally, and the proposed NSGA-II (LD+LR) that apply both dominance and recombination locally. Results are for KP500-2 setting the opti-

mum d^* that maximizes HV for each algorithm.

From this figure we can see that in algorithms that use *local recombination*, the entire population tends to form an arc sticking to the entire true Pareto front. Whereas in algorithms that adopt global recombination, the entire population tends to form a semicircle sticking to a part of true Pareto front. See Fig. 17 (b), (d) and compare with Fig. 17 (a) and (c), respectively. Also, note that the population of *local dominance* algorithms covers a broader region of objective space than the population of global dominance algorithms. See Fig. 17 (c), (d) and compare with Fig. 17 (a), (b), respectively. In other words, both *local recombination* and *local dominance* make the population more diverse in objective space and the effect of both aggregates positively. These good results obtained by local recombination are in accordance with previous reports on mating restrictions to recombine similar parents in elitist MOEAs^{(10),(11),(13)~(15)}. However, note that the proposed algorithm NSGA-II (LD+LR) in Fig. 17 (d) produces the most diverse population, fully covering the true Pareto front.

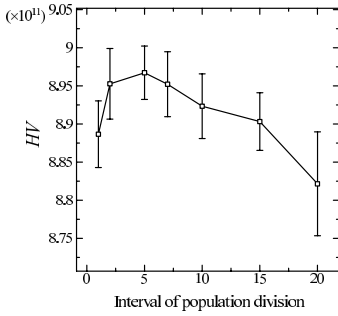


Fig. 18 Interval of population division (KP250-3).

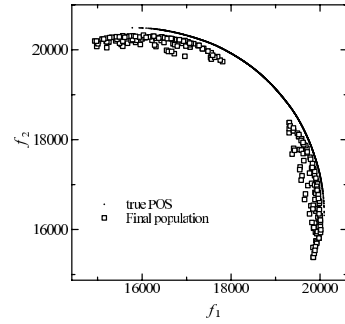
Summarizing, in order to obtain fully spread POS sticking to the true POS it is quite important to rank and select potential solutions as parent individuals using local dominance and additionally perform local recombination between them rather than apply global dominance and global recombination as conventional algorithms do.

7.2.4 Interval of Population Division

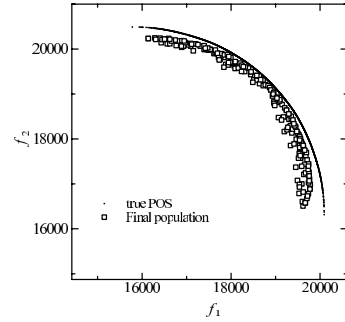
Here we observe the behavior of the algorithm when we vary the interval to divide the population in the proposed method. Note that Gap Suppression is performed only when we divide the population. **Figure 18** shows the absolute HV (not normalized) for test problem KP250-3 as we vary the interval in the range 1 to 20. From this figure we can see that there is an optimum interval around 5 generations to divide the population with Gap Suppression. In this case we can further improve the HV shown in Fig. 8.

7.2.5 Effect of Gap Suppression

This section shows the effect of Gap Suppression in the proposed method. **Figure 19** (a) and (b) show the obtained final population without and with Gap Suppression for test problem KP500-2 setting the parameter for population division $d = 2$. Note from Fig. 19 (a) that without Gap Suppression the final population does not cover a broad section of the true Pareto front, i.e., we can see a big gap between the two sub-populations. On the other hand, as shown in Fig. 19 (b), Gap Suppression effectively merges the regions covered by the two sub-populations. The effect of Gap Suppression is especially needed in problems with large search space and when a small number of sub-populations is used, as the example illustrated here.



(a) Without gap suppression



(b) With gap suppression

Fig. 19 Effect of gap suppression (KP500-2, $d = 2$).

7.3 Performance Comparison with DCPOS

Finally, in this section we compare results obtained by the proposed method with the ones obtained by DCPOS (Distributed Computation of Pareto Optimal Solutions)²¹. Both approaches aim to obtain fully distributed POS. However the differences between them make a fair comparison difficult. In the following, we discuss the main difference between the two approaches and then we present representative results by both schemes.

First, DCPOS divides the objective space and then assigns a population to each sub-space. On the contrary, the proposed method divides the actual population into sub-populations grouping the individuals by similar search direction in the objective space. In other words, the proposed method creates the sub-populations based on a sample of points of the actual objective space, i.e. population. As a result, each sub-population covers a different non-overlapping region of the objective space.

Second, DCPOS is static because the objective space is divided once at the beginning of the simulation and the sub-spaces remain fixed. The proposed method is dynamic in the sense that the regions of objective space covered by

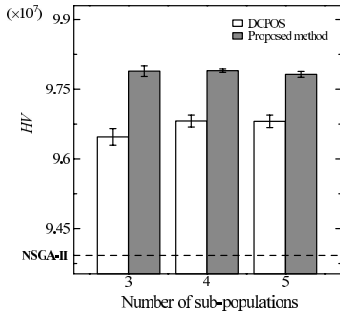
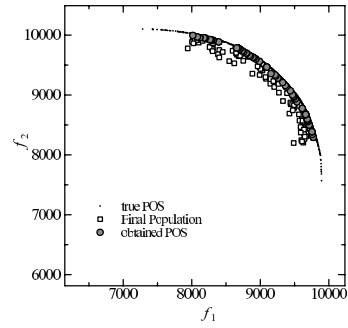


Fig. 20 HV obtained by DCPOS and proposed method (KP250-2).

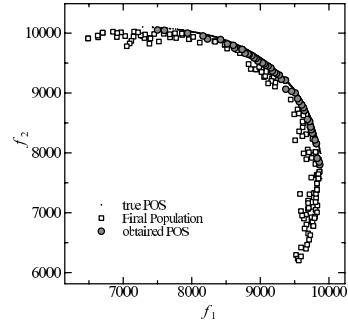
the sub-populations might vary at each generation. This is emphasized by slightly changing the sub-population size (Gap Suppression mechanism in Section 4.5). In the proposed method, the sub-populations would cover different regions even if the whole population remains the same in two consecutive generations (for example, due to elitism).

Third, DCPOS is a distributed EA, and each population could be assigned to one processor. The proposed method is designed as an alternative to single population MOEAs and no special consideration has been given for parallel distributed EAs. An extension of the proposed local dominance and local recombination for parallel implementation in the fashion of distributed EAs would be an important subject of future work.

Figure 20 illustrates the average *HV* achieved by DCPOS and the proposed method in test problem KP250-2 using 3, 4, and 5 sub-populations (population division parameter $d = \{3, 4, 5\}$ in the proposed method). Results by conventional NSGA-II are also included for comparison with a dashed line. DCPOS is set with a migration rate of 40 individuals and a migration frequency of 60 generations, which are the best parameters obtained experimentally for this problem. From this figure we can see that both methods achieve a significantly better *HV* than the conventional NSGA-II. However, note that the proposed method achieves higher *HV* than DCPOS in all settings tried here. **Figure 21** shows the final populations and the Pareto optimal solutions found by both methods in the case of 4 sub-populations, where the *HV* is maximum in both cases. Note from this figure that the propose method produces a better distributed final population and POS than DCPOS, which is consistent with the values of the *HV* reported in Fig. 20.



(a) DCPOS



(b) Proposed method

Fig. 21 Final population by DCPOS and proposed method (KP200-2, 4 sub-populations).

These results suggest that the proposed method can be superior to DCPOS to obtain better *HV* and better distribution of POS. But, again, it is difficult to fairly compare both methods because of the differences discussed above. In the future we would like to conduct a comprehensive comparison with DCPOS by parallelizing our scheme.

8. Conclusions

In this paper, we have proposed a method to enhance single population MOEAs by searching based on local dominance and local recombination. We verified that enhanced NSGA-II and SPEA2 implemented with our method show better search performance to obtain fully spread POS than the conventional versions of the same algorithms. The difference in performance increased with the number of objective and/or size of the search space. Another important advantage of the proposed method is a reduction in the entire computational cost to calculate dominance. As a reference, we could reduce about 20% of CPU time by NSGA-II and 80% by SPEA2 with the proposed method without any special consideration on optimization of programming and overhead removal.

In addition, we presented a detailed observation of the main components of the proposed method. Among them we showed that alignment of the principle search direction by rotation within local dominance is a very important feature to achieve high performance. We also showed that both local dominance and local recombination contribute to make the population more disperse in objective space and that their individual effects aggregate positively. However, there still remains the problem that convergence is slightly sacrificed in return for the noticeable improvement of diversity in the enhanced MOEAs. Also, note that the results given in this paper is on a limited set of examples of 0/1 multiobjective knapsack problems.

As future works, we would like to improve this method to achieve higher convergence of POS while keeping diversity as it is. In addition, we should verify the performance of this method for other kinds of problems, and more than three objectives. We would also like to introduce a more flexible and adaptive population division method. This would allow to achieve a better balance between convergence and diversity conditions, and would facilitate the scalability from multiple to many objectives optimization. Furthermore, the extension to parallel MOEAs of the principle of local dominance with alignment of the search direction by rotation, combined with local recombination, is another interesting line of research.

References

- 1) Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons (2001).
- 2) Coello, C.A.C., Veldhuizen, D.A.V. and Lamont, G.B.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, Boston, Kluwer Academic Publishers (2002).
- 3) Holland, J.H.: *Adaptation in Natural and Artificial Systems*, University of Michigan Press (1975).
- 4) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, Reading (1989).
- 5) Deb, K., Agrawal, S., Pratap, A. and Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, *KanGAL report 200001* (2000).
- 6) Zitzler, E., Laumanns, M. and Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm, *TIK-Report*, No.103 (2001).
- 7) Hajela, P. and Lin, C.Y.: Genetic search strategies in multicriterion optimal design, *Structural Optimization*, 4, pp.99–107 (1992).
- 8) Fonseca, C.M. and Fleming, P.J.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization, *Proc. 5th International Conference on Genetic Algorithms*, pp.416–423 (1993).
- 9) Jaszekiewicz, A.: Genetic local search for multi-objective combinatorial optimization, *European Journal of Operational Research*, Vol.137, No.1, pp.50–71 (2002).
- 10) Watanabe, S., Hiroyasu, T. and Miki, M.: Neighborhood cultivation genetic algorithm for multi-objective optimization problems, *Proc. 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002)*, pp.198–202 (2002).
- 11) Kim, M., Hiroyasu, T., Miki, M. and Watanabe, S.: ‘SPEA2+’: Improving the performance of the strength Pareto evolutionary algorithm 2, *Lecture Notes in Computer Science 3242 (PPSN VIII)*, pp.742–751 (2004).
- 12) Zitzler, E. and Thiele, L.: Multiobjective optimization using evolutionary algorithms — a comparative case study, *Proc. 5th International Conference on Parallel Problem Solving from Nature (PPSN-V)*, pp.292–301 (1998).
- 13) Ishibuchi, H. and Shibata, Y.: An empirical study on the effect of mating restriction on the search ability of EMO algorithms, *Proc. 2nd International Conference on Evolutionary Multicriterion Optimization (EMO 2003)*, *Lecture Notes in Computer Science 2632*, pp.433–447 (2003).
- 14) Ishibuchi, H. and Shibata, Y.: A similarity-based mating scheme for evolutionary multiobjective optimization, *Proc. 2003 Genetic and Evolutionary Computation Conference (GECCO 2003)*, *Lecture Notes in Computer Science 2723*, pp.1065–1076 (2003).
- 15) Ishibuchi, H. and Shibata, Y.: Mating scheme for controlling the diversity-convergence balance for multiobjective optimization, *Proc. 2003 Genetic and Evolutionary Computation Conference (GECCO 2004)*, *Lecture Notes in Computer Science 3102*, pp.1259–1271 (2004).
- 16) Ishibuchi, H. and Murata, T.: A multi-objective genetic local search algorithm and its application to flowshop scheduling, *IEEE Trans. System, Man, and Cybernetics*, Part C, Vol.28, pp.392–403 (1998).
- 17) Murata, T., Ishibuchi, H. and Gen, M.: Cellular genetic local search for multi-objective optimization, *Proc. Genetic and Evolutionary Computation Conference 2000*, pp.307–314 (2000).
- 18) Murata, T., Ishibuchi, H. and Gen, M.: Speci-

fication of Genetic Search Direction in Cellular Multi-objective Genetic Algorithms, *Proc. 1st Int'l. Conf. Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, Springer, Vol.1993, pp.82–95 (2001).

- 19) Jin, Y., Okabe, T. and Sendhoff, B.: Adaptive weighted Aggregation for Multiobjective Evolution Strategies, *Proc. 1st Int'l. Conf. Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, Vol.1993, pp.96–110 (2001).
- 20) Hiroyasu, T., Miki, M. and Watanabe, S.: The New Model of Parallel Genetic Algorithm in Multi-Objective Optimization Problems—Divided Range Multi-Objective Genetic Algorithm, *Proc. 2000 Congress on Evolutionary Computation*, Vol.1, pp. 333–340, Piscataway, New Jersey, IEEE Service Center (2000).
- 21) Deb, K., Zope, P. and Jain, A.: Distributed Computing of Pareto-Optimal Solutions with Evolutionary Algorithms, *Proc. 2nd Int'l. Conf. Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, Vol.2632, pp.534–549 (2003).
- 22) <http://www.tik.ee.ethz.ch/~zitzler/testdata.html>
- 23) Zitzler, E.: *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, Ph.D. thesis, Swiss Federal Institute of Technology, Zurich (1999).
- 24) Veldhuizen, D.A.V. and Lamont, G.B.: On Measuring Multiobjective Evolutionary Algorithm Performance, *2000 Congress on Evolutionary Computation*, Vol.1, pp.204–211 (2000).
- 25) Czyzak, P. and Jaszkiwicz, A.: Pareto-simulated annealing-A metaheuristic technique for multi-objective combinatorial optimization, *Journal of Multi-Criteria Decision Analysis*, Vol.7, No.1, pp.34–47 (1998).

Appendix

A.1 Computational Cost

In this section we give an estimate of the computational costs to perform population division and calculate dominance. The population division and local dominance procedures are illustrated in **Fig. 22** and **Fig. 23** for easy understanding of the computational cost calculation. In the following N denotes the size of the whole population and m the number of objectives.

A.1.1 Population Division

The three first steps of the population division procedure require mN computations each and the main loop that recursively divides the population requires

Procedure population division

```

fmin ← create minimum fitness vector ( $P(t)$ )
calculate temporal fitness vector ( $P(t)$ , fmin)
calculate polar coordinates ( $P(t)$ )
P( $t$ ) ←  $P(t)$ 
for  $i \leftarrow 1 \dots m - 1$ 
     $s \leftarrow \emptyset$ 
    for  $j \leftarrow 1 \dots d^{i-1}$ 
        sort (P $j$ ,  $\theta_i$ )
        { $s_1, \dots, s_d$ } ← divide (P $j$ ,  $d$ )
         $s \leftarrow s \cup \{s_1, \dots, s_d\}$ 
    end
P ←  $s$ 
end
    
```

Fig. 22 Population division method in objective space.

$\sum_{i=1}^{m-1} \left\{ d^{i-1} \left(\frac{N}{d^{i-1}} \log_2 \left(\frac{N}{d^{i-1}} \right) + \frac{N}{d^{i-1}} \right) \right\}$ using a sorting algorithm of $O(|P_j| \log_2 |P_j|)$, where $|P_j| \sim \frac{N}{d^{i-1}}$ is the size of current sub-population being split. Thus the time required by the population division procedure is given by

$$3mN + \sum_{i=1}^{m-1} \left\{ d^{i-1} \cdot \left(\frac{N}{d^{i-1}} \log_2 \left(\frac{N}{d^{i-1}} \right) + \frac{N}{d^{i-1}} \right) \right\}. \quad (12)$$

Factoring N and simplifying d^{i-1} we get

$$= 3mN + N \sum_{i=1}^{m-1} \{ \log_2 N - \log_2 d^{i-1} + 1 \}. \quad (13)$$

By eliminating $-\log_2 d^{i-1}$ we get

$$< 3mN + N(m-1)(\log_2 N + 1). \quad (14)$$

Again, by eliminating -1 we get

$$< mN(\log_2 N + 4). \quad (15)$$

which gives us the following order

$$O(mN \log_2 N). \quad (16)$$

A.1.2 Local Dominance

The two first two steps of the procedure to calculate local dominance require $m-1$ computations each and the next two steps $m|P_k|$ each, where $P_k \sim \frac{N}{d^{m-1}}$ is the sub-population size. The loop calculates dominance itself and requires $m|P_k|^2$. Since local dominance is calculated for each sub-population and there are d^{m-1} sub-populations, the overall computational cost of dominance is

Procedure local dominance

```

{ $\theta_{k1}, \dots, \theta_{km-1}$ }  $\leftarrow$  determine search direction
( $P_k(t)$ )
{ $\psi_{k1}, \dots, \psi_{km-1}$ }  $\leftarrow$  calculate rotation angle ( $\theta_k$ )
rotate ( $P_k(t), \psi_k$ )
transform polar coordinates to fitness ( $P_k(t)$ )
for  $p \leftarrow 1 \dots N/d^{m-1}$ 
  for  $q \leftarrow 1 \dots N/d^{m-1}$ 
    for  $j \leftarrow 1 \dots m$ 
       $P_k(t).individual[p].fitness[j] >? <$ 
         $P_k(t).individual[q].fitness[j]$ 
    end
  end
end
end
end

```

Fig. 23 Calculation of local dominance in sub-population.

$$d^{m-1} \left\{ 2(m-1) + 2m \left(\frac{N}{d^{m-1}} \right) + m \left(\frac{N}{d^{m-1}} \right)^2 \right\}. \quad (17)$$

By eliminating -1 and factoring m we get

$$< md^{m-1} \left\{ 2 + 2 \left(\frac{N}{d^{m-1}} \right) + \left(\frac{N}{d^{m-1}} \right)^2 \right\}. \quad (18)$$

By considering only the exponential term and simplifying d^{m-1} we get an order of

$$O \left(\frac{mN^2}{d^{m-1}} \right). \quad (19)$$

(Received November 22, 2005)

(Revised January 11, 2006)

(Accepted April 2, 2006)



Hiroyuki Sato received B.S. and M.S. degrees in Electrical and Electronic Engineering from Shinshu University, Nagano, Japan in 2003 and 2005, respectively. He is currently a doctoral student in Shinshu University. His research interests include multi-objective evolutionary algorithms and its applications. He is a member of IEICE.



Hernán Aguirre received his Engineer degree in Computer Systems from Escuela Politécnica Nacional, Quito, Ecuador in 1992. From 1997 to 2003 he was a research scholar sponsored by the Japanese Ministry of Education, Culture, Sports, Science and Technology. He received the M.S. and Ph.D. degrees from Shinshu University, Japan, in 2000 and 2003, respectively. Currently, he is a Research Associate at Shinshu University. His research interests include evolutionary computation, computational intelligence, information security, ad-hoc networks, and their applications. He is a member of IEEE, and IPSJ.



Kiyoshi Tanaka received B.S. and M.S. degrees from National Defense Academy, Yokosuka, Japan, in 1984 and 1989, respectively. In 1992, he received Dr. Eng. degree from Keio University, Tokyo, Japan. In 1995, he joined the Department of Electrical and Electronic Engineering, Faculty of Engineering, Shinshu University, Nagano, Japan, where he is currently a professor. His research interests include image and video processing, information hiding, evolutionary computation, chaos & fractals, and their applications. He is a member of IEEE, IEICE, IPSJ and IIEEJ.